Working Draft American National Standard

Project T10/2239-D

Revision 4
August 8, 2012

Information technology - SCSI over PCIe® Architecture (SOP) Revision 4

T10 Technical Editor:

Curtis E. Stevens
Western Digital Technologies, Inc.
3355 Michelson Ave. #100
Irvine, CA 92612
USA

Telephone: (949) 672-7933
Email: Curtis.Stevens@wdc.com

Reference number
ISO/IEC 14776-xxx:20xx
ANSI INCITS *-20xx**

Points of Contact

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

T10 Chair
John B. Lohmeyer
LSI Logic
4420 Arrows West Drive
Colorado Springs, CO 80907-3444
USA

Telephone: (719) 533-7560
Email: lohmeyer@t10.org

T10 Vice-Chair
Mark S. Evans
Western Digital Corporation
5863 Rue Ferrari
San Jose, CA 95138
USA

Telephone:(408) 363-5257
Email: mark.evans@wdc.com

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

T10 E-mail reflector:
Server: majordomo@t10.org
To subscribe send e-mail with “subscribe” in message body
To unsubscribe send e-mail with “unsubscribe” in message body

INCITS Secretariat
1101 K Street, NW
Suite 610
Washington, DC 20005
USA

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

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

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

Web site: http://www.techstreet.com/incitsgate.tmpl
Telephone: (734) 302-7801 or (800) 699-9277

Global Engineering Documents, an IHS Company
15 Inverness Way East
Englewood, CO 80112-5704
USA

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

American National Standard
for Information Technology

Draft

SCSI over PCIe Architecture

Secretariat
Information Technology Industry Council

Approved mm.dd.yy

American National Standards Institute, Inc.

ABSTRACT

This standard specifies the messages to be transported across PCIe to support the services required by SAM-5 of a Service Delivery Subsystem. The means of configuring the PCIe transport and passing messages across PCIe is included in the companion specification, PQI.

Draft

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made 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, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

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

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

Revision History

Revision History (part 1 of 6)			
Rev	**Date**	**Author**	**Description**
0	July 14, 2011	Curtis E. Stevens	1) Initial Draft, taken from 11-276r2
1	November 6, 2011	Curtis E. Stevens	1) Integrated 11-279r7 - SOP IU Definitions 2) Integrated 11-308r2 - bit and byte ordering conventions 3) Converted definitions format to match new conventions

Revision History (part 2 of 6)			
Rev	Date	Author	Description

1a	January 22, 2012	Curtis E. Stevens	1) Integrated 11-278r5 - Added definition of SOP domains.
			2) Integrated 11-515r1 - Renumbered and reorganized IUs.
			3) Removed all IUs that are being defined in other proposals.
			4) Integrated 12-003r1 - Add SCSI device names and port names.
			5) Integrated 12-002r2 - SCSI Port Identifiers. There was a lot of overlap with 12-003. The SAM-5 attribute mapping was incomplete. Used the 12-003 table that appeared to be complete. The names and identifiers tables was in the same condition. Used the complete table from 12-003.
			6) Integrated 12-025r2 - SOP SCSI Event Model
			7) Integrated 11-395r5 - Protocol Specific Field. This was written for the standard body, but there was also an Annex reserved for this definition. Reformatted the proposal into Annex C. This facilitates additions more readily. It could be prettier, I am sure this will get reworked a bit.
			8) Integrated 11-341r7 - Added descriptor list material to the COMMAND IUs and reworked the COMMAND IUs to include some new fields.
			9) Added “...” in table ranges.
			10) Integrating 11-212r6 - SOP Bridging. This assumes that 11-515 and 11-516 have been integrated; however, 11-516 has not been approved. The following items were discussed at the telecon 23-Jan-2012
			    1) The Bridge port identifier for iSCSI data structure was missing several length fields. Added these to the structure and renumbered table. Formulas need to be checked.
			    2) Removed duplicate field definitions from text. Kept the more developed ones.
			    3) Add an IU subclause for Bridge management requests. This added a level to this part of the proposal
			    4) Did not implement Table 7 (result field). Not sure where this goes, the material seems unreferenced.
			    5) Removed PAD field from REPORT BRIDGE NEXUS DETAILS parameter data (data-in). There was no description of z.
			11) Created a source for Figure 1 and inserted. This will probably need some rework.
			12) converted little endian fields from 100% yellow to 30% yellow.
			13) Fixed first/last notation to use brackets.

Revision History (part 3 of 6)			
Rev	Date	Author	Description
1b	March 7, 2012	Curtis E. Stevens	1) Made corrections to revision history. 2) Integrated 12-102r1 that is the output from February editorial review. 3) integrated 11-450r5, Mapping SOP over PQI (Annex A). This adds material regarding the queuing layer specific information in the IUs. 4) Integrated 12-033r4. This adds normative material showing information unit sequences. 5) Integrate 12-001r3. This adds capability reporting for the queuing interface. 6) Integrate 12-097r1. Adds clarification to the success IU. 7) Integrate 12-100r0. This unifies the command IU formats. 8) Added references and terms from 12-026r3. The remainder will be added in the next rev.
2	March 8, 2012	Curtis E. Stevens	1) Updated to show rev 2 after plenary confirmation.
2a	March 19, 2012	Curtis E. Stevens	1) Incorporated editorial changes based on E-Mail feedback A) Changed HEAD1 paragraph style to only have 1 line of margin above the paragraph B) 12-102r0 - Bunch of updates C) Changed GlossParagraph to remove the12pts before the paragraph D) Changed Gloss2t to remove the 12pts after the paragraph. E) Clarified definition of nexus identifier in the COMMAND IU and TASK MANAGEMENT IU. Other IUs refer to these descriptions. F) Fixed several occurrences of incorrect small caps G) Fixed reported spelling errors 2) Incorporated 11-394r2. This defines the SET and REPORT CONFIGURATION IUs
2b	March 20, 2012	Curtis E. Stevens	1) Incorporated 11-516r4. This proposal adds error response IUs for errors that do not have an explicit queue. 2) Converted all colors from 30% x to the equivalent Tb_color from the template. Created Tb_Light_Green for blocking the IU Header. 3) Changed the definition of Tb_light_red to match the current doc. 4) Fixed misspelling in the template for Tb_Light_Yellow. Original entry we Td_Light_Yellow.

Revision History (part 4 of 6)			
Rev	**Date**	**Author**	**Description**
2c	March 21, 2012	Curtis E. Stevens	1) Incorporated 12-149r1. Changed several of the bridge IUs to report number of descriptors instead of number of descriptor bytes. 2) Converted all tables to have the table subheading color for the subheadings. This mostly affected Data descriptor lists. 3) Incorporated 12-106r1. This field aligns the BRIDGE IUs and the REPORT GENERAL IU 4) Incorporated 12-026r3. Adds to the bridge architecture model along with updates to Annex C 5) Implemented editors note that suggested moving all of 5.4 (i.e., SCSI transport protocol specific data) into Annex C
2d	April 11, 2012	Curtis E. Stevens	1) Incorporate 12-214. Rob Elliott nits on Rev 2b. 2) Incorporate 12-144r2. Adds definitions for various SOP objects 3) Incorporate 12-197r0. Removes left over sections in Annex A 4) Incorporate 12-211. Adds a vendor specific area to the REPORT GENERAL IU
2e	April 19, 2012	Curtis E. Stevens	1) Incorporate 12-203r2. Clarifies sense data length in the BIDIRECTIONAL RESPONSE IU 2) Incorporate 12-208r0. Merges the Command response IU and Bidirectional response IU. 3) Incorporate 12-199r2. Normalize IU introductory paragraphs 4) Incorporate 12-209r0. Changes the name of the BIDIRECTIONAL COMMAND IU to EXTENDED COMMAND IU. There were inconsistencies in the naming of the BIDIRECTIONAL COMMAND IU that were also normalized while incorporating this proposal. 5) Incorporate 12-205r1. Changes the Additional CDB length to be a coded value instead of a usable byte value. 6) Incorporate 11-389r3. Adds SCSI application layer services. 7) incorporate 12-183r2. Normalized references to host queuing layer and device queuing layer in both PQI and SOP 8) Incorporate 12-206r1. Clarify the use of SGL Segment.
3	May 10, 2012	Curtis E. Stevens	1) Plenary voted to accept 2e as the current working draft. This is published as Rev 3.

Revision History (part 5 of 6)			
Rev	**Date**	**Author**	**Description**
3a	May 10, 2012	Curtis E. Stevens	1) Modified all tables that contained field ranges with no MSB or LSB designation to center the lines inside the field descriptions. This is not marked as a change. 2) Fixed small caps in the List of Tables. Apparently it is not good enough to make them small caps, the need the style field name... 3) Incorporate 12-223r1. Adds a description of data transfers to the model clause. 4) Incorporate 12-204r2. defines the SGL type numbers 5) Incorporate 12-093r2. Defines service delivery subsystem error handling. 6) Incorporate 12-148r3. Provides a model clause describing request identifiers. 7) Incorporate 12-235r1. This cleans up several of the TBD areas in clause 5 and clause 6 8) Incorporate 11-169r4. SPC-4 overflow and underflow handling 9) Incorporate 11-391r2. Adds asynchronous event detection and reporting. 10) Incorporate 12-240r1. Merges the IQ ERROR IU with the EVENT IU along with changing terminology regarding IQ and OQ. This proposal had several sections that were added with the incorporation of 11-391. This material was skipped in 12-240. Did not compare word for word. Assumed that Rob's impeccable style kept them the same.
3b	June 11, 2012	Curtis E. Stevens	1) Accepted all the changes to create a version without changebars.
3c	June 13, 2012	Curtis E. Stevens	1) Performed a full spell check. This should have caught errors associated with changebars and missing spaces. 2) Added missing author names in the integrated proposal list. 3) Incorporated 12-285r1. This cleans up issues associated with SET and REPORT CONFIGURATION 4) Incorporated 12-271r2. Reorganizes some of the error codes for data-in and data-out transfer results. 5) Reincorporated the UML convention, taken from SAM5r11. Beware if you are using visio, the embedded objects can cause frame 10 to crash. Incorporated the illustrations as metafile format instead of embedded objects. This solved all the problems with frame 10. Also found that Visio 10 behaves better if it is set to XP compatibility. 6) Deleted unused state diagram convention. SOP does not have any state transition diagrams at this time. 7) Added editorial updates found in 12-313r0 and 12-314r0. 8) Deleted editors notes that no longer apply due to proposals being delayed until SOP-2
3d	July 10, 2012	Curtis E. Stevens	1) Added editorial updates found in 12-324r0 2) Incorporated editorial updates received in E-Mail

Revision History (part 6 of 6)			
Rev	Date	Author	Description
4	August 8, 2012	Curtis E. Stevens	1) Corrected a header error in several of the structure format tables. 2) Integrated 11-396r1. Defines a new section that describes the PCIe configuration of a SOP device. 3) Integrated 11-305r3. Updates the SOP architecture model. 4) Implemented issue #74 as voted in the WG. Deleted Annex B which describes SOP over NVMe 5) Incorporated 12-088r9. Adds a reference to Annex A and a description of PQI Reset and PQI Nexus Loss conditions 6) Incorporated 12-280r1. Adds a protocol specific port identifier for SOP to SPC-4. The change to SPC-4 is replicated here. 7) Incorporated 12-309r1. Provides a clause 4 definition of SOP IU elements 8) Incorporated 12-331r1. This proposal overlaps both 12-309r1 and 12-088r9. Conflicts were resolved during the incorporation of this proposal. The I_T Nexus loss conditions are documented in A.6 and referenced in 4.7.2 9) Incorporated 12-308r1. Cleans up TBD's associated with TMF Executed Response and other items 10) Incorporated 12-342r0. Adds a vendor specific field to REPORT BRIDGE LOCAL POR T DETAILS 11) Incorporated 12-193r7. Defines SOP Bridge Error responses 12) Incorporated WG review and editorial edits from 12-392r1 13) Normalized spacing 14) Performed a consistency pass to make sure tables are properly continued and conventions are properly followed 15) Added the copyright statement for no declared patents. 16) Set the copyright year to 2012

New Capabilities added to SOP

Integrated Proposal List			
#	Doc	Author	Description
1	11-279r7	Curtis Stevens	Initial definition of SOP IU's
2	11-308r2	Rob Elliott	Bit and byte ordering conventions
3	11-278r5	Mark Evans	Defines SOP domains
4	11-515r1	Rob Elliott	Renumbered and reorganized IUs
5	12-003r1	Rob Elliott	Defines SOP device and port names
6	12-002r2	Rob Elliott	Defines SCSI Port identifiers
7	12-025r1	Rob Elliott	Defines the SOP SCSI Event Model
8	11-395r5	Brad Besmer	Protocol Specific field definition
9	11-341r7	Steve Johnson	Added text for descriptor list. This does not include material for Annex A and Annex B.
10	11-212r6	Rob Elliott	Added IUs and architecture material for SOP Bridging
11	11-450r5	Ie-Wei Njoo	Added the queuing layer specific field information to Annex A
12	12-033r4	Mark Evans	Added information unit sequences
13	12-102r1	Curtis E. Stevens	Editors session review comments
14	12-001r3	Rob Elliott	Adds capability reporting for the queuing interface
15	12-097r1	Brad Besmer	Clarifies use of the SUCCESS IU
16	12-100r0	Rob Elliott	Unifies the command IU formats
17	12-102r0	Curtis E. Stevens	Bundles editorial comments to clean-up Rev 2.
18	11-394r2	Brad Besmer	This defines the SET and REPORT CONFIGURATION IUs
19	11-516r4	Rob Elliott	This proposal adds error response IUs for errors that do not have an explicit queue.
20	12-149r1	Rob Elliott	Changed several of the bridge IUs to report number of descriptors instead of number of descriptor bytes.
21	12-026r3	Rob Elliott	Adds to the bridge architecture model along with updates to Annex C
22	12-214r0	Curtis E. Stevens	Editorial changes and responses to requests from Rob Elliott.
23	12-144r2	Mark Evans	Adds definitions for various SOP objects
24	12-197r0	Ie Wei Njoo	Removes A.1-A.10
25	12-211r0	Satish Vasudeva	Adds a vendor specific area to the REPORT GENERAL IU
26	12-203r2	Rob Elliott	Clarifies sense data length in the BIDIRECTIONAL RESPONSE IU
27	12-208r0	Rob Elliott	Extends the COMMAND RESPONSE IU to cover the bidirectional case and removes the BIDIRECTIONAL RESPONSE IU
28	12-199r2	Rob Elliott	Normalize IU introductory paragraphs
29	12-209r0	Ie Wei Njoo	Changes the name of the BIDIRECTIONAL COMMAND IU to EXTENDED COMMAND IU
30	12-205r1	Rob Elliott	Changes the additional CDB length field to be a coded value
31	11-389r3	Rob Elliott	Adds SCSI application layer services
32	12-183r2	Rob Elliott	Normalizes references to host queuing layer and device queuing layer in both PQI and SOP

Integrated Proposal List			
#	Doc	Author	Description
33	12-206r1	Rob Elliott	Clarify the use of SGL Segment
34	12-223r1	Mark Evans	Data transfer model
35	12-204r2	Ie Wei Njoo	SGL types for SOP in Annex A
36	12-093r3	Dan Colegrove	Device server error handling
37	12-148r3	Rob Elliott	Request identifier model
38	12-235r1	Rob Elliott	TBD Cleanups
39	11-169r4	Rob Elliott	SPC-4 overflow and underflow handling
40	11-391r2	Rob Elliott	Asynchronous events
41	12-240r1	Rob Elliott	Merge IQ ERROR IU into EVENT IU
42	12-285r1	Brad Besmer	Cleanup issues with SET and REPORT CONFIGURATION IUs.
43	12-271r2	Rob Elliott	PQI administrator function error handling with SOP coordination
44	11-396r1	Richard Solomon	SOP: PCI Express Miscellanea
45	11-305r3	Mark Evans	Updates SOP architecture model
46	12-088r9	Ie Wei	SOP PQI Resets
47	12-280r1	Ralph Weber	SPC-4/SOP Designators for Target Port Identifiers
48	12-309r1	Rob Elliott	SOP IU and element clarification
49	12-331r1	Rob Elliott	SOP SPC-4 Partial I_T nexus loss handling
50	12-308r1	Rob Elliott	SOP Received TMF Executed Service Response cleanup
51	12-342r0	Rob Elliott	Add vendor specific bytes to REPORT BRIDGE LOCAL PORT DETAILS
52	12-193r7	Rob Elliott	SOP Bridge error responses
53			
54			
55			
56			
57			
58			
59			
60			

Contents

Tables

Figures

FOREWORD (This foreword is not part of this standard)

SCSI over PCIe® architecture (SOP) defines the upper portion of a SCSI transport protocol for the PCIe architecture (see http://www.pcisig.com). Standardizing this interface is intended to provide a consistent interface between the PCIe hosts and endpoints while preserving the existing SCSI software infrastructure. This ensures that investments in such solutions have a stable and managed migration path as technologies continue to evolve and expand.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, International Committee for Information Technology Standards, Information Technology Industry Council, Suite 610, 1101 K Street, NW, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by the International Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, INCITS had the following members:

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

John B. Lohmeyer, Chair
Mark Evans, Vice-Chair
Ralph O. Weber, Secretary

INTRODUCTION

This standard is divided into the following clauses and annexes:

Clause 1 (Scope) describes the relationship of this standard to the SCSI and PCIe standards.

Clause 2 (Normative references) provides references to other standards and documents.

Clause 3 (Definitions, symbols, abbreviations, keywords, and conventions) defines terms and conventions used throughout this standard.

Clause 4 (Model) describes the operational model.

Clause 5 (SOP Information Unit layer) describes the SOP information unit layer.

Clause 6 (Application layer) describes the Application layer.

Annex A (Mapping SOP over PQI) describes how to map SOP to PQI.

Annex B (Protocol specific requirements) describes requirements for communicating between SOP and other protocols.

1 Scope

The SCSI family of standards provides for many different transport protocols that define the rules for exchanging information between different SCSI devices. This standard defines the rules for exchanging information between SCSI devices using a PCIe queueing mechanism. Other SCSI transport protocol standards define the rules for exchanging information between SCSI devices using other interconnects.

This standard does not define the configuration of the PCIe topology, and the mechanisms for sending and receiving these messages across the PCIe link(s).

Figure 1 shows the relationship of this standard to the other standards and related projects in the SCSI family of standards as of the publication of this standard.

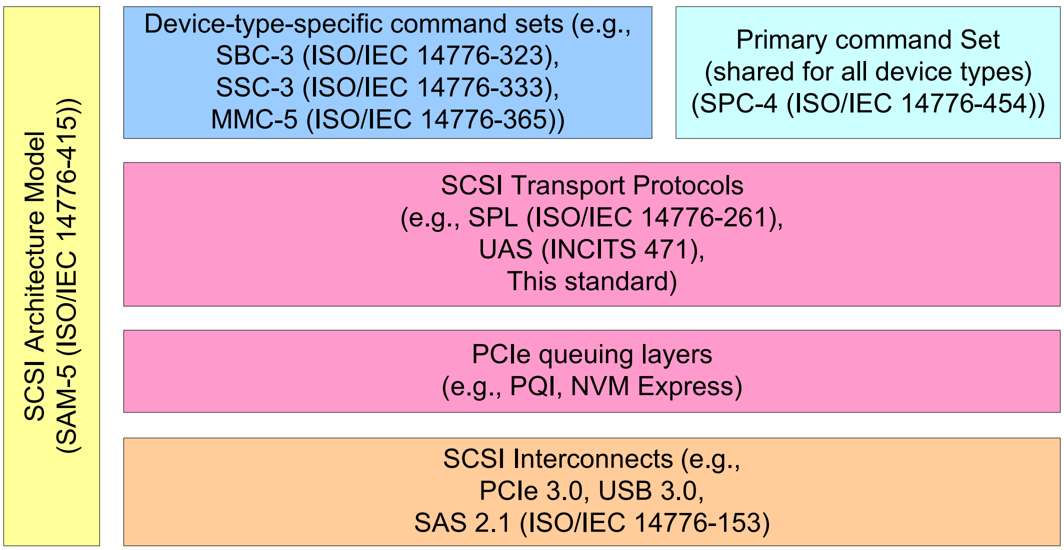


Figure 1 — SCSI document relationships

2 Normative references

2.1 Normative references

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

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

Additional availability contact information is provided below as needed.

2.2 Approved references

ISO/IEC 14776-241, *SCSI RDMA Protocol (SRP)* [ANSI INCITS 365-2002]

ISO/IEC 14776-224, *Fibre Channel Protocol for SCSI - 4 (FCP-4)* [ANSI INCITS 481-2012]

ISO 80000-1:2009, *Quantities and units -- Part 1: General*

IEC 80000-13:2008, *Quantities and units -- Part 13: Information science and technology*

INCITS 470-2011, *Fibre Channel - Framing and Signaling - 3 (FC-FS-3)*

2.3 References under development

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

ISO/IEC 14776-252, *USB Attached SCSI - 2 (UAS-2)* [T10/2229-D]

ISO/IEC 14776-262, *SAS Protocol Layer - 2 (SPL-2)* [T10/2228-D]

ISO/IEC 14776-415, *SCSI Architecture Model - 5 (SAM-5)* [T10/2104-D]

ISO/IEC 14776-923, *SCSI / ATA Translation - 3 (SAT-3)* [T10/2126-D]

ISO/IEC 14776-454, *SCSI Primary Commands - 4 (SPC-4)* [T10/1731-D]

RFC 793, *Transmission Control Protocol - DARPA Internet Program - Protocol Specification*

2.4 Other references

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

PCI Local Bus Specification (PCI) Revision 3.0, February 3, 2004

PCI Express Base Specification (PCIe) Revision 3.0, November 10, 2010

Editor’s Note 1: Check revisions and dates

PCI Code and ID Assignment Specification (PCI-ID) Revision 1.2, March 15, 2012.

PCI Bus Power Management Interface Specification (PCI-PM) revision 1.2, March 3, 2004

PCI Express Card Electromechanical Specification (PCIe-CEM) revision 3.0, version 0.9, May 23, 2011

NOTE 1 - For more information on the current status of PCI documents, contact the PCI-SIG (see http://www.pcisig.com).

RFC 3720, *Internet Small Computer Systems Interface (iSCSI)*

NOTE 2 - Copies of approved IETF standards may be obtained through the Internet Engineering Task Force (IETF) at www.ietf.org.

Serial ATA (SATA) Revision 3.1, 18 July 2011.

NOTE 3 - For information on the current status of Serial ATA documents, contact the Serial ATA International Organization (see http://www.sata-io.org).

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 aggregation
when used in class diagrams, a form of association that defines a whole-part relationship between the whole (i.e., aggregate) and its parts

3.1.2 association
when used in class diagrams a relationship between two or more classes that specifies connections among their objects (i.e., relationship that specifies that objects of one class are connected to objects of another class)

3.1.3 attribute
when used in class diagrams, a named property of a class that describes the range of values that the class or its objects may hold. When used in object diagrams, a named property of an instance of a class

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

3.1.5 bridge (see 4.1.2.9)
object in a SOP device (see 3.1.60) that connects the SOP port (see 3.1.64) in the SOP device with a different SCSI domain (e.g., a SAS domain (see SPL-2))

3.1.6 bridge application client (see 4.1.2.9.2)
object in a bridge (see 3.1.5) that routes information either ~~between~~to and from a bridge device server (see 3.1.7) and a SCSI initiator port (see SAM-5) or ~~between~~to and from a bridge task manager (see 3.1.9) and a SCSI~~SOP~~ initiator port ~~(see 3.1.63) and a bridge device server~~

3.1.7 bridge device server (see 4.1.2.9.4)
object in a bridge (see 3.1.5) that routes information ~~either~~ between a SOP target port (see 3.1.67) and a bridge application client (see 3.1.6)~~ or between a bridge application client and a SCSI target port (see SAM-5)~~

3.1.8 bridge port identifier (see B.3.1)
identifier for a bridge's local port or a remote port accessed through the bridge

3.1.9 bridge task manager (see 4.1.2.5.3)
object in a bridge (see 3.1.5) that controls the sequencing of commands and processes task management functions in the bridge

3.1.10 bus number (see PCI and PCIe)
encoded value used to select one of the PCI buses in a PCI fabric

3.1.11 byte
sequence of eight contiguous bits considered as a unit

3.1.12 class
description of a set of objects that share the same attributes, operations, relationships, and semantics. Classes may have attributes and may support operations

3.1.13 constraint
when used in class diagrams and object diagrams, a mechanism for specifying semantics or conditions that are maintained as true between entities (e.g., a required condition between associations)

3.1.14 $D3_{hot}$ (see PCI-PM)
PCI function power management state that consumes minimum power for a PCI function that is receiving power

3.1.15 $D0_{uninitialized}$ (see PCI-PM)
PCI function power management state that consumes maximum power for a PCI function that has not been initialized by system software

3.1.16 Data-In Buffer (see SAM-5 and SPC-4)
buffer specified by the application client to receive data from the device server during the processing of a SCSI command or SOP management function

3.1.17 Data-Out Buffer (see SAM-5 and SPC-4)
buffer specified by the application client to supply data that is transferred to the device server during the processing of a SCSI command or SOP management function

3.1.18 Data Buffer
for a SCSI command or SOP management function that transfers data from the Data-In Buffer, the Data-In Buffer; for a SCSI command or SOP management function that transfers data from the Data-Out Buffer, the Data-Out Buffer

3.1.19 device number (see PCI and PCIe)
encoded value used to select one of the PCI devices on a PCI bus

3.1.20 device queuing layer (see 4.1.2.8.4)
object that transfers SOP IUs (see 5.2) to an outbound queue and removes SOP IUs from an inbound queue

3.1.21 field
group of one or more contiguous bits

3.1.22 function number (see PCI and PCIe)
encoded value used to select one of the functions in a PCI device

3.1.23 generalization
when used in class diagrams, a relationship among classes where one class (i.e., superclass) shares the attributes and/or operations on one or more classes (i.e., subclasses)

3.1.24 host queuing layer (see 4.1.2.8.3)
object that transfers SOP IUs (see 5.2) to an intbound queue and removes SOP IUs from an outbound queue

3.1.25 inbound queue (see 4.4)
a queue that is used to transfer SOP IUs (see 5.2) from a host queuing layer to a device queuing layer

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

3.1.27 little-endian
format for storage of binary data in which the least significant byte appears first. In a multi-byte value, the byte containing the least significant bit is stored in the lowest memory address and the byte containing the most significant bit is stored in the highest memory address

3.1.28 local port (see 4.3.1)
a SCSI port in a bridge used to access a remote port

3.1.29 SOP device (see 4.1.2.3)
class whose objects are, or an object that is, connected to a service delivery subsystem, supports the SOP protocol, and may serve as a SCSI device (see SAM-5)

3.1.30 SOP initiator device (see 4.1.2.4)
class whose objects originate, or an object that originates, device service and task management requests, receives device service and task management responses, and may serve as a SCSI initiator device (see SAM-5)

3.1.31 SOP management application client (see 4.1.2.10)
object in a SOP initiator device (see 3.1.62) that specifies management tasks to be performed by SOP management device servers (see 3.1.32)

3.1.32 SOP management device server (see 4.1.2.11)
object in a SOP target device (see 3.1.66) that performs management tasks specified by SOP management application clients (see 3.1.31)

3.1.33 SOP target device (see 4.1.2.5)
class whose objects receive, or an object that receives, device service and task management requests for processing, sends device service and task management responses, and may serve as a SCSI target device (see SAM-5)

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

3.1.35 multiplicity
when used in class diagrams, an indication of the range of allowable instances that a class or an attribute may have

3.1.36 nexus identifier (see 4.3.2)
identifier assigned by a bridge for an I_T nexus on the far side of the bridge in which the bridge is able to participate (see 4.3.2)

3.1.37 object
entity with a well-defined boundary and identity that encapsulates state and behavior. All objects are instances of classes (i.e., a concrete manifestation of a class is an object)

3.1.38 operation
when used in class diagrams, a service that may be requested from any object of the class in order to effect behavior. Operations describe what a class is allowed to do and may be a request or a question. A request may change the state of the object but a question should not

3.1.39 outbound queue (see 4.4)
a queue that is used to transfer SOP IUs (see 5.2) from a device queuing layer to a host queuing layer

3.1.40 PCI Express cold reset (see PCIe)
PCI Express reset triggered by application of power to the component

3.1.41 PCI Express device (see PCIe)
collection of one or more PCI functions identified by a common bus number and device number, if any

3.1.42 PCI Express fabric (see 4.1.2.2.2)
object in a SOP service delivery subsystem (see 3.1.65) that connects all SOP devices (see 3.1.60) in a SOP domain (see 3.1.61)

3.1.43 PCI Express function (see PCIe)
set of logic in a PCI Express device that is represented by a single configuration space

3.1.44 PCI Express function-level reset (see PCIe)
PCI Express mechanism for resetting a PCI Express function

3.1.45 PCI Express hard reset (see PCIe)
PCI Express reset triggered by a hardware mechanism provided by the system (e.g., PERST# in PCIe-CEM) or generated autonomously by the component

3.1.46 PCI Express hot reset (see PCIe)
PCI Express reset triggered by one of the following in-band mechanisms:

a) using the TS1 and TS2 Ordered sets' Training Control field Hot Reset bit set to one;
b) disabling a link; or
c) other events that cause the data link layer to report DL_Down status

3.1.47 PCI Express soft reset (see PCI-PM and PCIe)
PCI Express reset triggered by a $D3_{hot}$ to $D0_{uninitialized}$ transition if the Power Management Control/Status Register No_Soft_Reset bit is set to zero

3.1.48 PCI Express warm reset (see PCIe)
PCI Express reset triggered by hardware without the removal and re-application of power to the component

3.1.49 queue (see 4.4)
set of queue elements

3.1.50 queue element (see 4.4)
a single entry in a queue

3.1.51 queue structure (see 4.1.2.2.3)
object in a SOP service delivery subsystem (see 3.1.65) that contains one or more inbound queues and one or more outbound queues

3.1.52 3.1.53 remote port (see 4.3.1)
SCSI port in a SCSI domain accessed through a bridge's local port

3.1.54 request IU
IU with an IU type greater than or equal to 01h and less than or equal to 7Fh (see 5.2.1)

3.1.55 response IU
IU with an IU type greater than or equal to 80h and less than or equal to FFh (see 5.2.1)

3.1.56 role
when used in class diagrams and object diagrams, a label at the end of an association or aggregation that defines a relationship to the class on the other side of the association or aggregation

3.1.57 routing ID (see 4.2.8 and PCIe)
PCI Express identifier consisting of a bus number, a device number if any, and a function number

3.1.58 scatter gather list
a data structure in an IU and memory address space used to describe a data buffer

3.1.59 SGL segment
a portion of a scatter gather list contiguous in an IU or contiguous in memory address space

3.1.60 SOP device (see 4.1.2.3)
SOP initiator device (see 3.1.62) or SOP target device (see 3.1.66) that may serve as a SCSI device (see SAM-5)

3.1.61 SOP domain (see 4.1.2)
I/O system defined by this standard that may serve as a SCSI domain (see SAM-5)

3.1.62 SOP initiator device (see 4.1.2.3)
SOP device (see 3.1.60) containing one or more SOP initiator ports (see 3.1.63) in a SOP domain (see 3.1.61)

3.1.63 SOP initiator port (see 4.1.2.8.5)
SCSI initiator port (see SAM-5) in a SOP domain (see 3.1.61)

3.1.64 SOP port (see 4.1.2.8)
object containing a host queuing layer (see 3.1.24) or device queuing layer (see 3.1.20), and a SOP initiator port (see 3.1.63) or a SOP target port (see 3.1.67)

3.1.65 SOP service delivery subsystem (see 4.1.2.2)
part of a SOP I/O system that transmits information between a SOP initiator port (see 3.1.63) and a SOP target port (see 3.1.67) and may serve as a SCSI domain (see SAM-5)

3.1.66 SOP target device (see 4.1.2.3)
SOP device (see 3.1.60) containing one or more SOP target ports (see 3.1.67) in a SOP domain (see 3.1.61)

3.1.67 SOP target port (see 4.1.2.8.5)
SCSI target port (see SAM-5) in a SOP domain (see 3.1.61)

3.2 Symbols and abbreviations

Abbreviation	Meaning
XOR	exclusive logical OR
^	exclusive logical OR
×	multiplication
/	division
≠ or NE	not equal
≤ or LE	less than or equal to
±	plus or minus
≈	approximately
×	multiply
+	add
-	subtract
< or LT	less than
= or EQ	equal
> or GT	greater than
≥ or GE	greater than or equal to
DL_Down	Data Link Down (see PCIe)
ID	identifier
iSCSI	Internet SCSI (see iSCSI)
IU	Information Unit (see 5.2)
FCP	Fibre Channel Protocol for SCSI (see FCP-4)
LSB	least significant bit (see 3.1.26)

LUN	logical unit number
MSB	most significant bit (see 3.1.34)
NVMe	NVM Express
PCI	Peripheral Component Interconnect
PERST#	PCI Express reset auxiliary signal (see PCIe-CEM)
PQI	PCIe architecture Queuing Interface (see PQI)
SAS SSP	Serial Attached SCSI Serial SCSI Protocol (see SPL-2)
SBC-2	SCSI Block Commands-2 (see 2.3)
SOP	SCSI over PCIe architecture (i.e., this standard)
SPC-3	SCSI Primary Commands-3 (see 2.3)
SRP	SCSI RDMA Protocol (see SRP)
TCP	Transmission Control Protocol (see RFC 793)
TS1	Training Sequence 1 (see PCIe)
TS2	Training Sequence 2 (see PCIe)
UAS	USB Attached SCSI (see UAS-2)
USB	Universal Serial Bus

3.3 Keywords

3.3.1 invalid
A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt by a device server of an invalid bit, byte, word, field or code value shall be reported as error.

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

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

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

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

3.3.6 option, optional
Keywords that describe features that are not required to be implemented by this standard. However, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

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

3.3.8 reserved
A keyword referring to bits, bytes, words, fields, and code values that are set aside for future standardization. A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words, or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

3.3.9 restricted
A keyword referring to bits, bytes, words, and fields that are set aside for other identified standardization purposes. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field in the context where the restricted designation appears.

3.3.10 shall
A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

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

3.3.12 vendor specific
Something (e.g., a bit, field, code value) that is not defined by this standard. Specification of the referenced item is determined by the SCSI device vendor and may be used differently in various implementations.

3.4 Editorial conventions

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

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

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

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

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

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

a) red (i.e., one of the following colors):
   A) crimson: or
   B) amber;
b) blue; or
c) green.

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

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

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

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

Notes and examples do not constitute any requirements for implementation and notes are numbered consecutively throughout this standard.

3.5 Numeric and character conventions

3.5.1 Numeric conventions

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

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

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

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

This standard uses the following conventions for representing decimal numbers:

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

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

Table 1 — Numbering conventions

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

3.5.2 Byte encoded character strings conventions

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

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

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

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

3.6 Bit and byte ordering

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

Tables defining data structures are shown with one row per byte and one column per bit. The lowest byte offset is at the top and the highest byte offset is at the bottom. The least significant bit (LSB) of each byte is numbered 0 and is shown on the right, and the most significant bit (MSB) of each byte is numbered 7 and shown on the left.

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

In a big-endian field, the byte containing the MSB is at the lowest byte offset and the byte containing the LSB is at the highest byte offset. The bits in big-endian fields are not shaded.

In a little-endian field, the byte containing the MSB is at the highest byte offset and the byte containing the LSB is at the lowest byte offset. The bits in little-endian fields are shaded.

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels for that field. The MSB and LSB of each subfield may be shown in another table.

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

Multiple byte fields are represented with three rows, with the non-sequentially increasing byte numbers separated by a row labeled '...'.

Table 2 shows how this standard depicts a 64-bit field with subfields. The subfields may be defined in a table defining just that field.

Table 2 — Example of a 64-bit field with subfields

Bit Byte	7	6	5	4	3	2	1	0
•••	Other field(s), if any							
n	Field name							
•••								
n + 7								
•••	Other field(s), if any							

Table 3 shows how this standard depicts a 32-bit big-endian field.

Table 3 — Example of a 32-bit big-endian field

Byte Bit	7	6	5	4	3	2	1	0
•••	Other field(s), if any							
n	(MSB) Field name							
•••								
n + 3								(LSB)
•••	Other field(s), if any							

Table 4 shows the bit numbers for the field shown in table 3.

Table 4 — Bit assignments in a 32-bit big-endian field

Bit / Byte	7	6	5	4	3	2	1	0
•••	Other field(s), if any							
n	(MSB) Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
n + 1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
n + 2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
n + 3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
•••	Other field(s), if any							

EXAMPLE 1 - If the field in table 3 and table 4 contains a value of 00010203h, then:

a) byte n contains 00h;
b) byte n+1 contains 01h;
c) byte n+2 contains 02h; and
d) byte n+3 contains 03h.

Table 5 shows how this standard depicts a 32-bit little-endian field.

Table 5 — Example of a 32-bit little-endian field

Bit / Byte	7	6	5	4	3	2	1	0
•••	Other field(s), if any							
n								(LSB)
•••	Field name							
n + 3	(MSB)							
•••	Other field(s), if any							

Table 6 shows the bit numbers for the field shown in table 5.

Table 6 — Bit numbers for a 32-bit little-endian field

Bit / Byte	7	6	5	4	3	2	1	0
•••	Other field(s), if any							
n	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	(LSB) Bit 0
n + 1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
n + 2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
n + 3	Bit 31 (MSB)	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
•••	Other field(s), if any							

EXAMPLE 3 -If the field in table 5 and table 6 contains a value of 00010203h, then:

a) byte n contains 03h;
b) byte n+1 contains 02h;
c) byte n+2 contains 01h; and
d) byte n+3 contains 00h.

3.7 Notation for procedure calls

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

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

Where:

Result:A single value representing the outcome of the procedure call.

Procedure Name:A descriptive name for the function modeled by the procedure call.

Input-1, Input-2, …:A comma-separated list of names identifying caller-supplied input arguments.

Output-1, Output-2, …:A comma-separated list of names identifying output arguments to be returned by the procedure call.

"[…]": Brackets enclosing optional or conditional arguments.

This notation allows arguments to be specified as inputs and outputs. The following is an example of a procedure call specification:

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

Where:

Found = Flag

Flag, if set to one, indicates that a matching item was located.

Input Arguments:

Pattern = … /* Definition of **Pattern** argument */

Argument containing the search pattern.

Item List = Item<NN> /* Definition of **Item List** as an array of NN **Item** arguments*/

Contains the items to be searched for a match.

Output Arguments:

Item Found = Item … /* Item located by the search procedure call */

This argument is only returned if the search succeeds.

3.8 UML notation convention

3.8.1 Notation conventions overview

This standard uses class diagrams and object diagrams with notation that is based on the Unified Modeling Language (UML).

See 3.8.3 for the conventions used for class diagrams.

See 3.8.4 for the conventions used for object diagrams.

3.8.2 Constraint and note conventions

Class diagrams and object diagrams may include constraints, which specify requirements, and notes, which are informative.

Table 7 shows the notation used for constraints and notes.

Table 7 — Constraint and note notation

Notation	Description
{Constraint text}	The presence of the curly brackets (i.e. {}) defines constraint that is a normative requirement. An example of a constraint is shown in figure 3.
note text	The absence of curly brackets defines a note that is informative. An example of a note is shown in figure 4.

3.8.3 Class diagram conventions

Table 8 shows the notation used for classes in class diagrams.

Table 8 — Class diagram notation for classes

Notation	Description
Class Name **Class Name** **Class Name**	A class with no attributes or operations.
Class Name Attribute01[1] Attribute02[1] **Class Name** Attribute01[1] Attribute02[1]	A class with attributes and no operations.
Class Name Operation01() Operation02()	A class with operations and no attributes.
Class Name Attribute01[1] Attribute02[1] Operation01() Operation02()	A class with attributes and operations.
Class Name Attribute01[1..*] Attribute02[1] Operation01() Operation02()	A class with attributes that have a specified multiplicity (see table 9) and operations.

Table 9 shows the notation used to indicate multiplicity in class diagrams.

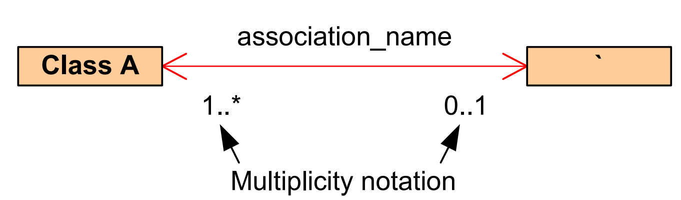
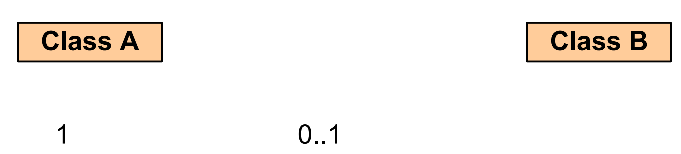
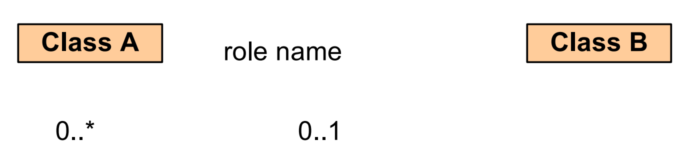
Table 9 — Multiplicity notation

Notation	Description
not specified	The number of instances of an attribute is not specified.
1	One instance of the class or attribute exists.
0..*	Zero or more instances of the class or attribute exist.
1..*	One or more instances of the class or attribute exist.
0..1	Zero or one instance of the class or attribute exists.
n..m	n to m instances of the class or attribute exist (e.g., 2..8).
x, n..m	Multiple disjoint instances of the class or attribute exist (e.g., 2, 8..15).

Table 10 shows the notation used to denote association (i.e., “knows about”) relationships between classes.

Unless the two classes in an association relationship also have an aggregation relationship, association relationships have multiplicity notation (see table 4) at each end of the relationship line.

Table 10 — Class diagram notation for associations

Notation	Description
	Class A knows about Class B (i.e., read as “Class A association_name Class B”) and Class B knows about Class A (i.e., read as “Class B association name Class A”).
	Class B knows about Class A (i.e., read as “Class B knows about Class A”) but Class A does not know about Class B.
	Class A knows about Class B (i.e., read as “Class A uses the role name attribute of Class B”) but Class B does not know about Class A.
Note - The use of role names and association names are optional.	

See figure 2 for examples of association relationships between classes.

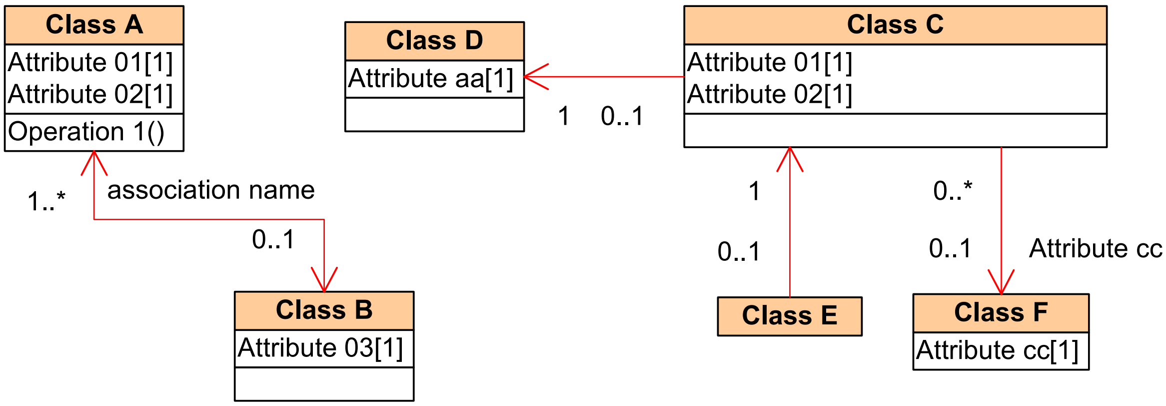
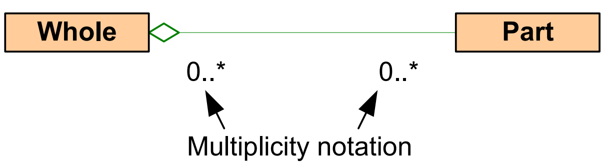


Figure 2 — Examples of association relationships in class diagrams

Table 11 shows the notation used to denote aggregation (i.e., “is a part of” or “contains”) relationships between classes. The aggregation relationship is a specific type of association and always include multiplicity notation (see table 4) at each end of the relationship line.

Table 11 — Class diagram notation for aggregations

Notation	Description
	The Part class is part of the Whole class and may continue to exist even if the Whole class is removed (i.e, read as “the whole contains the part”).
	The Part class is part of the Whole class, shall only belong to one Whole class, and shall not continue to exist if the Whole class is removed (i.e., read as “the whole contains the part”).

See figure 3 for examples of aggregation relationships between classes.

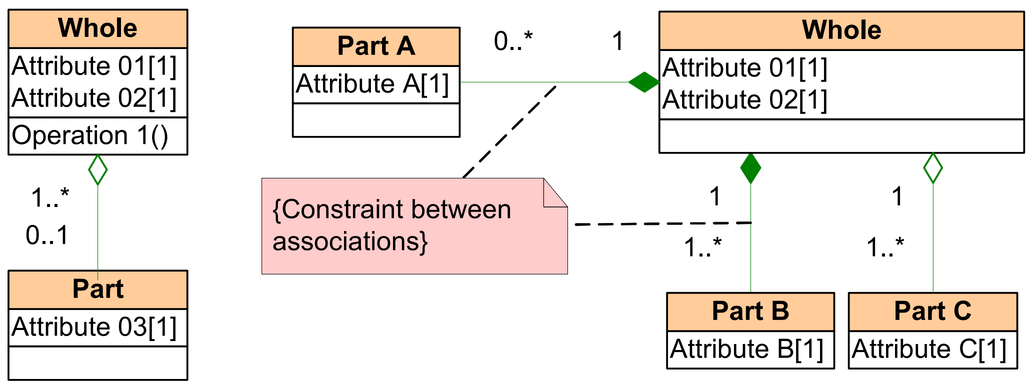


Figure 3 — Examples of aggregation relationships in class diagrams

Table 12 shows the notation used to denote generalization (i.e., “is a kind of”) relationships between classes.

Table 12 — Class diagram notation for generalizations

Notation	Description
	Subclass is a kind of superclass. A subclass shares all the attributes and operations of the superclass (i.e., the subclass inherits from the superclass).

See figure 4 for examples of generalization relationships between classes.

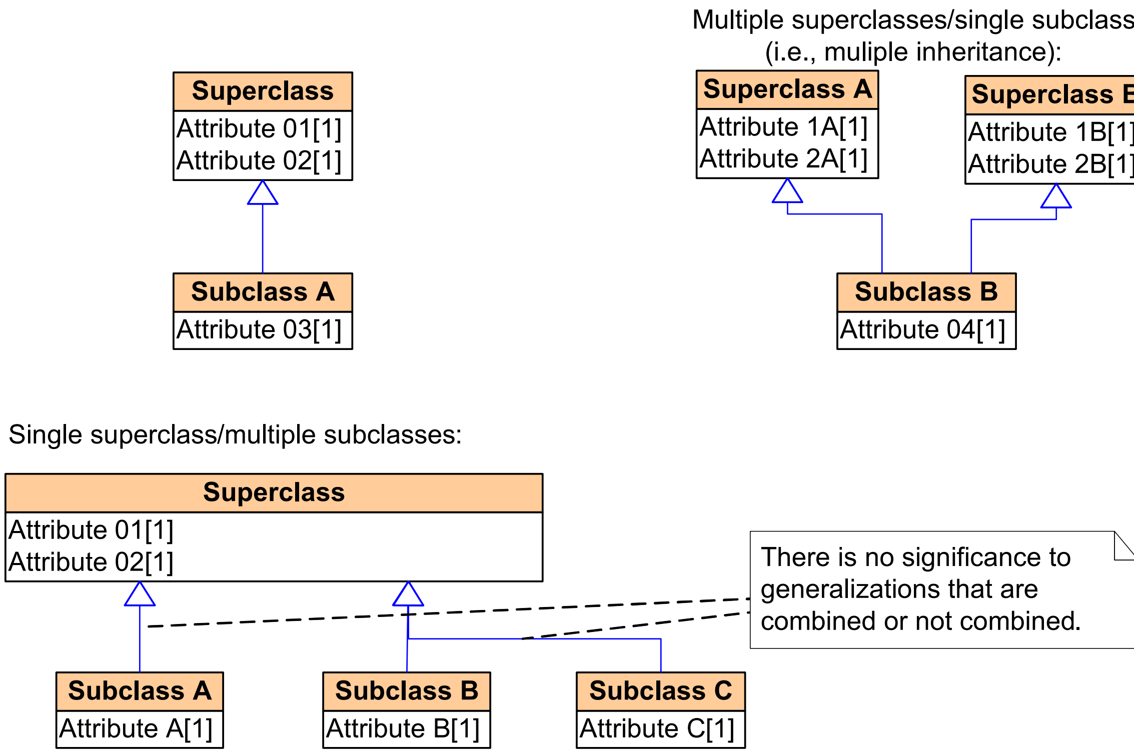


Figure 4 — Example of generalization relationships in class diagrams

Table 13 shows the notation used to denote dependency (i.e., "depends on") relationships between classes.

Table 13 — Class diagram notation for dependency

Notation	Description
	Class A depends on class B. A change in class B may cause a change in class A.

See figure 5 for an example of a dependency relationship between classes.

Figure 5 — Example of a dependency relationship in class diagrams

3.8.4 Object diagram conventions

Table 14 shows the notation used for objects in object diagrams.

Table 14 — Object diagram notation for objects

Notation	Description
label : Class Name	Notation for a named object with no attributes.
label : Class Name Attribute01 = x Attribute02 = y	Notation for a named object with attributes.
: Class Name	Notation for a anonymous object with no attributes.
: Class Name Attribute01 = x Attribute02 = y	Notation for a anonymous object with attributes.

Table 15 shows the notation used to denote link relationships between objects.

Table 15 — Object diagram notation for link

Notation	Description
: Object A ———————— : Object B	An instance of an association between object A and object B.

See figure 6 for examples of a link relationships between objects.

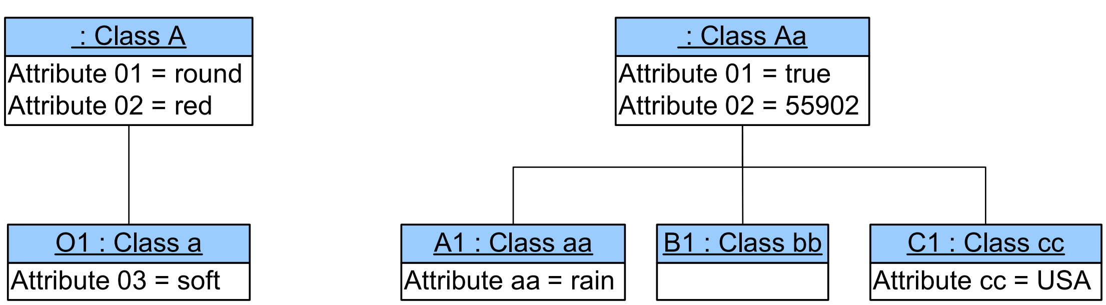


Figure 6 — Examples of link relationships for object diagrams

4 Model

4.1 SOP classes

4.1.1 SOP classes overview

Figure 7 shows the main functional classes of the SOP domain. This standard defines these classes in greater detail.

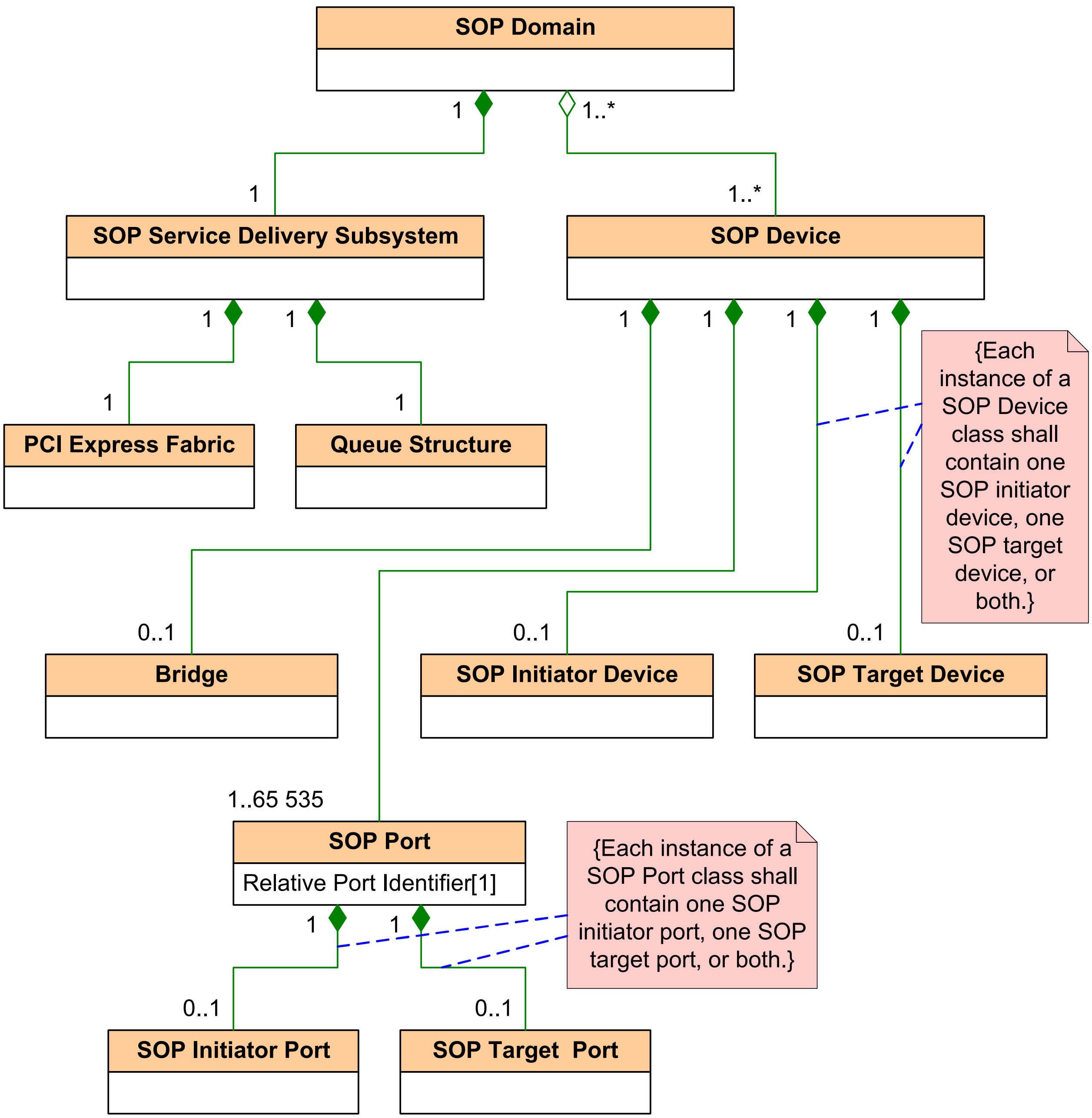


Figure 7 — The SOP domain class diagram overview

Figure 8 shows the relationship between some of the SOP classes and the corresponding SCSI classes.

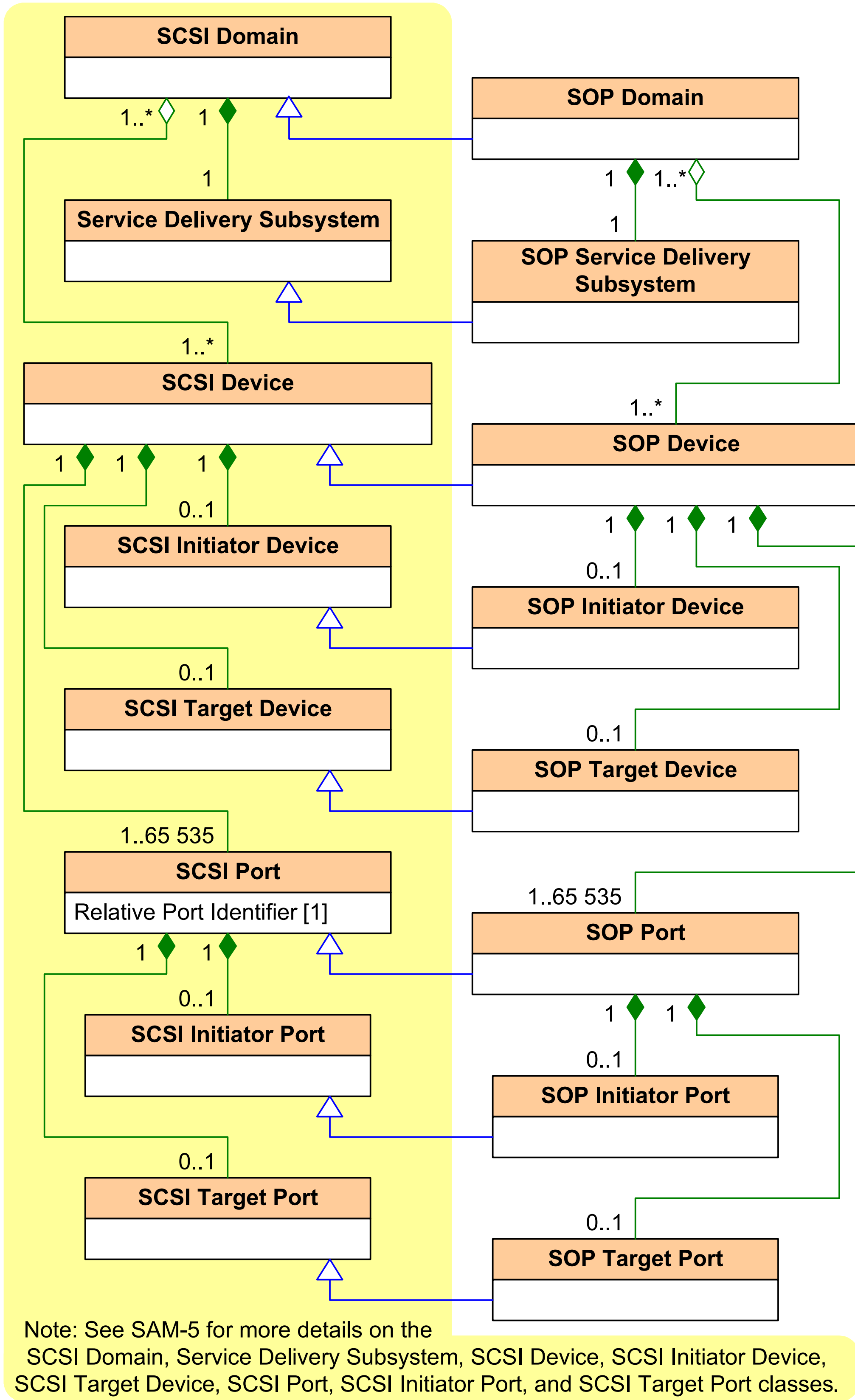


Figure 8 — Relationship between SOP classes and SCSI classes

4.1.2 SOP domain class

4.1.2.1 SOP domain class overview

The SOP Domain class (see figure 9) contains:

a) the SOP Service Delivery Subsystem class (see 4.1.2.2); and
b) the SOP Device class (see 4.1.2.3).

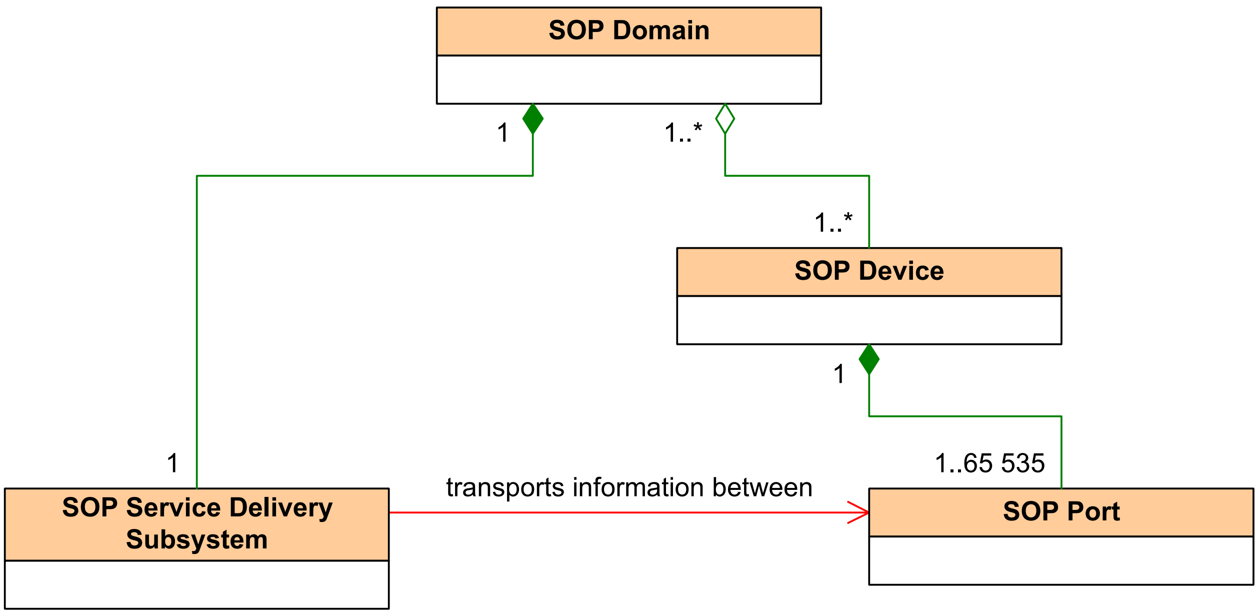


Figure 9 — SOP Domain class diagram

Each instance of a SOP Domain class shall contain the following objects:

a) one SOP service delivery subsystem;
b) one or more SOP devices; and
c) from one to 65 535 SOP ports per SOP device.

4.1.2.2 SOP Service Delivery Subsystem class

4.1.2.2.1 SOP Service Delivery Subsystem class overview

The SOP Service Delivery Subsystem class (see figure 10) contains:

d) the PCI Express Fabric class (see 4.1.2.2.2); and
e) the Queue Structure class (see 4.1.2.2.3).

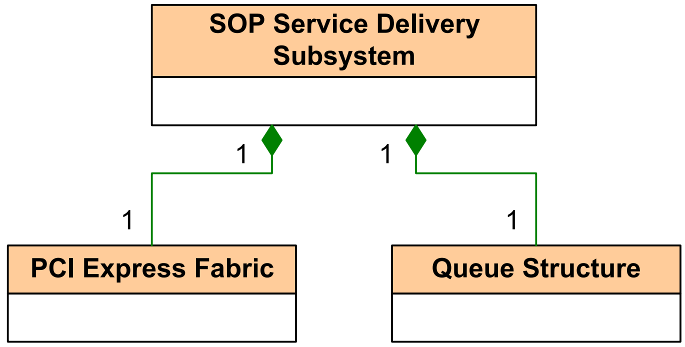


Figure 10 — SOP Service Delivery Subsystem class diagram

Each instance of a SOP Domain class shall contain the following objects:

a) one PCI Express fabric; and
b) one queue structure.

4.1.2.2.2 PCI Express Fabric class

The PCI Express Fabric class connects all the SOP devices in the SOP domain, providing a mechanism through which application clients communicate with device servers and task managers.

The PCI Express Fabric class includes backplane traces, cables, and PCIe switches (see PCIe).

4.1.2.2.3 Queue Structure class

Each instance of a Queue Structure class shall contain:

a) one or more inbound queues (see 4.4); and
b) one or more outbound queues (see 4.4).

4.1.2.3 SOP Device class

The SOP Device class (see figure 7) contains:

a) the SOP Initiator Device class (see 4.1.2.4);
b) the SOP Target Device class (see 4.1.2.5);
c) the SOP Port class (see 4.1.2.8); and
d) the Bridge class (see 4.1.2.9).

Each instance of a SOP Device class:

a) shall contain:
   A) one SOP initiator device;
   B) one SOP target device; or
   C) both;

   and

b) shall contain:
   A) from one to 65 535 SOP ports; and
   B) zero or one bridge.

4.1.2.4 SOP Initiator Device class

The SOP Initiator Device class (see figure 11) contains:

a) the Application Client class (see SAM-5); and
b) the SOP Management Application Client (see 4.1.2.6); ~~and~~
c) ~~the SOP Management Device Server class (see 4.1.2.7)~~.

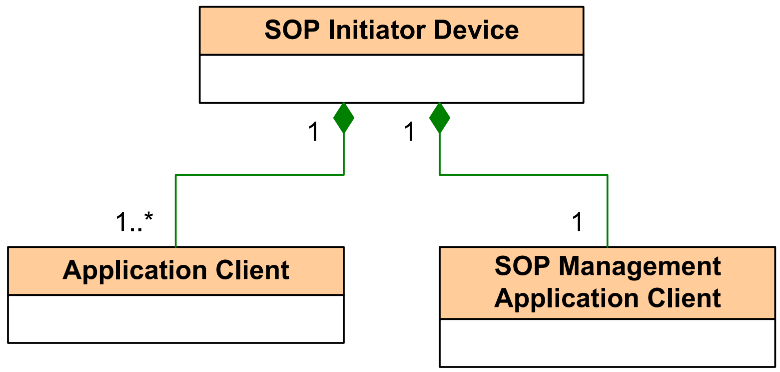


Figure 11 — SOP Initiator Device class diagram

Each instance of a SOP Initiator Device class shall contain only one ~~of the following:~~

a) ~~a~~ one or more application clients; and
b) one SOP management application client~~; or~~
c) ~~a SOP management device server~~.

~~If an instance of a SOP device (see 4.1.2.3) contains:~~

a) ~~a SOP initiator device; and~~
b) ~~a SOP initiator port that contains a host queuing layer,~~

~~then the SOP initiator device shall contain a SOP management application client.~~

~~If an instance of a SOP device contains:~~

a) ~~a SOP initiator device; and~~
b) ~~a SOP initiator port that contains a device queuing layer,~~

~~then the SOP initiator device shall contain a SOP management device server.~~

4.1.2.5 SOP Target Device class

The SOP Target Device class (see figure 12) contains:

a) the Logical Unit class (see SAM-5);
b) ~~the SOP Management Application Client (see 4.1.2.6);~~ and
c) the SOP Management Device Server class (see 4.1.2.7).

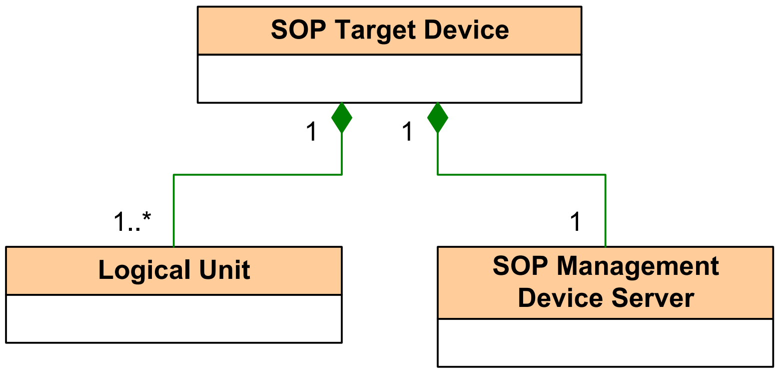


Figure 12 — SOP Target Device class diagram

Each instance of a SOP Target Device class shall contain ~~only one of the following~~:

a) ~~a SOP management application client; or~~one or more logical units; and
b) ~~a~~ one SOP management device server.

~~If an instance of a SOP device contains:~~

a) ~~a SOP target device; and~~
b) ~~a SOP target port that contains a device queuing layer,~~

~~then the SOP target device shall contain a SOP management device server.~~

~~If an instance of a SOP device (see 4.1.2.3) contains:~~

a) ~~a SOP target device; and~~
b) ~~a SOP target port that contains a host queuing layer,~~

~~then the SOP target device shall contain a SOP management application client.~~

4.1.2.6 SOP Management Application Client class

The SOP Management Application Client class specifies management tasks to be performed by SOP management device servers.

4.1.2.7 SOP Management Device Server class

The SOP Management Device Server class performs management tasks specified by SOP management application clients.

4.1.2.8 SOP Port class

4.1.2.8.1 SOP Port class overview

The SOP Port class (see figure 7) contains:

a) the SOP Initiator Port class (see 4.1.2.8.2); and
b) the SOP Target Port class (see 4.1.2.8.5).

Each instance of a SOP Port class shall contain:

a) one SOP initiator port;
b) one SOP target port; or
c) both.

4.1.2.8.2 SOP Initiator Port class

4.1.2.8.2.1 SOP Initiator Port class overview

The SOP Initiator Port class (see figure 13) contains:

a) the Host Queuing Layer class (see 4.1.2.8.6)~~; and~~
b) ~~the Device Queuing Layer class (see 4.1.2.8.4)~~.

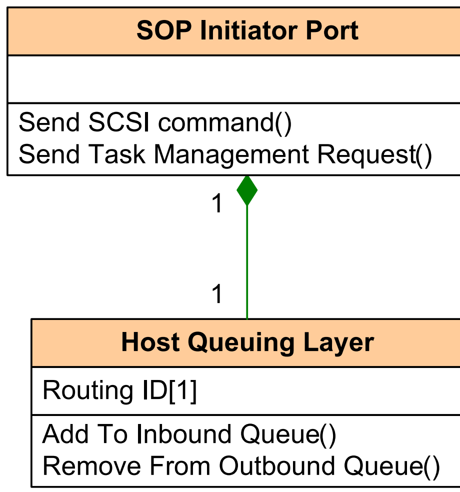


Figure 13 — SOP Initiator Port class diagram

Each instance of a SOP Initiator Port class shall contain only one ~~of the following:~~

a) ~~a~~ host queuing layer~~; or~~
b) ~~a device queuing layer~~.

~~If an instance of a SOP device (see 4.1.2.3) contains:~~

a) ~~a SOP initiator port; and~~
b) ~~a SOP initiator device that contains a SOP management application client,~~

~~then the SOP initiator port shall contain a host queuing layer.~~

~~If an instance of a SOP device contains:~~

a) ~~a SOP initiator port; and~~
b) ~~a SOP initiator device that contains a SOP management device server,~~

~~then the SOP initiator port shall contain a device queuing layer.~~

The SOP Initiator Port class processes:

a) the Send SCSI Command operation (see 4.1.2.8.2.2); and
b) the Send Task Management Request operation (see 4.1.2.8.2.3).

4.1.2.8.2.2 Send SCSI Command operation

The SOP initiator port processes this operation by:

1) creating a command IU containing the SCSI CDB; and

2) invoking the Add To Inbound Queue operation of the host queuing layer.

This operation is inherited from the SCSI Initiator Port class (see SAM-5).

4.1.2.8.2.3 Send Task Management Request operation

The SOP initiator port processes this operation by:

1) creating a command IU containing the task management function; and
2) invoking the Add To Inbound Queue operation of the device queuing layer.

This operation is inherited from the SCSI Initiator Port class (see SAM-5).

4.1.2.8.3 ~~Host Queuing Layer class~~

4.1.2.8.3.1 ~~Host Queuing Layer class overview~~

~~The Host Queuing Layer class provides queuing layer services to the SOP port.~~

4.1.2.8.3.2 ~~Routing ID attribute~~

~~The Routing ID attribute contains the routing ID (see 4.2.8).~~

4.1.2.8.3.3 ~~Add To Inbound Queue operation~~

~~This operation places SOP IUs in an inbound queue for transmission to a queuing layer device.~~

~~This operation is modeled by the following procedure call:~~

~~Add To Inbound Queue (IUs, Number Of IUs)~~

~~This operation is defined in A.2.1.1 and Annex B.~~

4.1.2.8.3.4 ~~Remove From Outbound Queue operation~~

~~This operation removes SOP IUs from an outbound queue that received them from a device queuing layer.~~

~~This operation is modeled by the following procedure call:~~

~~IUs = Remove From Outbound Queue (Number Of IUs)~~

~~This operation is defined in A.2.1.2 and Annex B.~~

4.1.2.8.4 ~~Device Queuing Layer class~~

4.1.2.8.4.1 ~~Device Queuing Layer class overview~~

~~The Device Queuing Layer class provides queuing layer services to the SOP port.~~

4.1.2.8.4.2 ~~Routing ID attribute~~

~~The Routing ID attribute contains the routing ID (see 4.2.8).~~

4.1.2.8.4.3 ~~Add To Outbound Queue operation~~

~~This operation places SOP IUs in an outbound queue for transmission to a host queuing layer.~~

~~This operation is modeled by the following procedure call:~~

~~Add To Outbound Queue (IUs, Number Of IUs)~~

~~This operation is defined in A.2.2.1 and Annex B.~~

4.1.2.8.4.4 ~~Remove From Inbound Queue operation~~

~~This operation removes SOP IUs from an inbound queue that received them from a host queuing layer.~~

~~This operation is modeled by the following procedure call:~~

~~IUs = Remove From Inbound Queue (IUs, Number Of IUs)~~

~~This operation is defined in A.2.2.2 and Annex B.~~

4.1.2.8.5 SOP Target Port class

4.1.2.8.5.1 SOP Target Port class overview

The SOP Target Port class (see figure 11) contains:

a) ~~the Host Queuing Layer class (see 4.1.2.8.3); and~~
b) the Device Queuing Layer class ~~(see 4.1.2.8.4)~~(see 4.1.2.8.7).

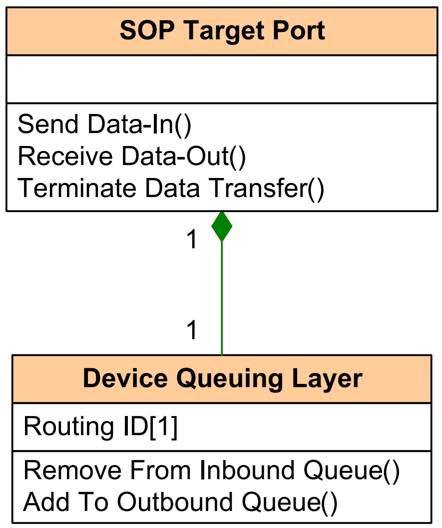


Figure 14 — SOP Target Port class diagram

Each instance of a SOP Target Port class shall contain only one~~ of the following:~~

a) ~~a host queuing layer; or~~
b) ~~a~~ device queuing layer.

~~If an instance of a SOP device (see 4.1.2.3) contains:~~

a) ~~a SOP target port; and~~
b) ~~a SOP target device that contains a SOP management device,~~

~~then the SOP target port shall contain a device queuing layer.~~

~~If an instance of a SOP device contains:~~

a) ~~a SOP target port; and~~
b) ~~a SOP target device that contains a SOP management application client,~~

~~then the SOP target port shall contain a host queuing layer.~~

The SOP Target Port class processes:

a) the Send Data-In operation (see 4.1.2.8.5.2);
b) the Receive Data-Out operation (see 4.1.2.8.5.3); and
c) the Terminate Data Transfer operation (see 4.1.2.8.5.4).

4.1.2.8.5.2 Send Data-In operation

The SOP target port processes this operation by performing PCI memory writes based on the scatter gather list for the command.

This operation is inherited from the SCSI Target Port class (see SAM-5).

4.1.2.8.5.3 Receive Data-Out operation

The SOP target port processes this operation by performing PCI memory reads based on the scatter gather list for the command.

This operation is inherited from the SCSI Target Port class (see SAM-5).

4.1.2.8.5.4 Terminate Data Transfer operation

The SOP target port processes this operation by performing PCI memory reads based on the scatter gather list for the command.

This operation is inherited from the SCSI Target Port class (see SAM-5).

4.1.2.8.6 Host Queuing Layer class

4.1.2.8.6.1 Host Queuing Layer class overview

The Host Queuing Layer class provides queuing layer services to the SOP port.

4.1.2.8.6.2 Routing ID attribute

The Routing ID attribute contains the routing ID (see 4.2.8).

4.1.2.8.6.3 Add To Inbound Queue operation

This operation places SOP IUs in an inbound queue for transmission to a queuing layer device.

This operation is modeled by the following procedure call:

Add To Inbound Queue (IUs, Number Of IUs)

This operation is defined in A.2.1.1 and Annex B.

4.1.2.8.6.4 Remove From Outbound Queue operation

This operation removes SOP IUs from an outbound queue that received them from a device queuing layer.

This operation is modeled by the following procedure call:

IUs = Remove From Outbound Queue (Number Of IUs)

This operation is defined in A.2.1.2 and Annex B.

4.1.2.8.7 Device Queuing Layer class

4.1.2.8.7.1 Device Queuing Layer class overview

The Device Queuing Layer class provides queuing layer services to the SOP port.

4.1.2.8.7.2 Routing ID attribute

The Routing ID attribute contains the routing ID (see 4.2.8).

4.1.2.8.7.3 Add To Outbound Queue operation

This operation places SOP IUs in an outbound queue for transmission to a host queuing layer.

This operation is modeled by the following procedure call:

Add To Outbound Queue (IUs, Number Of IUs)

This operation is defined in A.2.2.1 and Annex B.

4.1.2.8.7.4 Remove From Inbound Queue operation

This operation removes SOP IUs from an inbound queue that received them from a host queuing layer.

This operation is modeled by the following procedure call:

IUs = Remove From Inbound Queue (IUs, Number Of IUs)

This operation is defined in A.2.2.2 and Annex B.

4.1.2.9 Bridge class

4.1.2.9.1 Bridge class overview

The Bridge class (see figure 15) contains:

a) the SCSI Port class (see SAM-5);
b) the Bridge Application Client class (see 4.1.2.9.2);
c) the Bridge Task Manager class (see 4.1.2.9.3);
d) the Bridge Device Server class (see 4.1.2.9.4); and

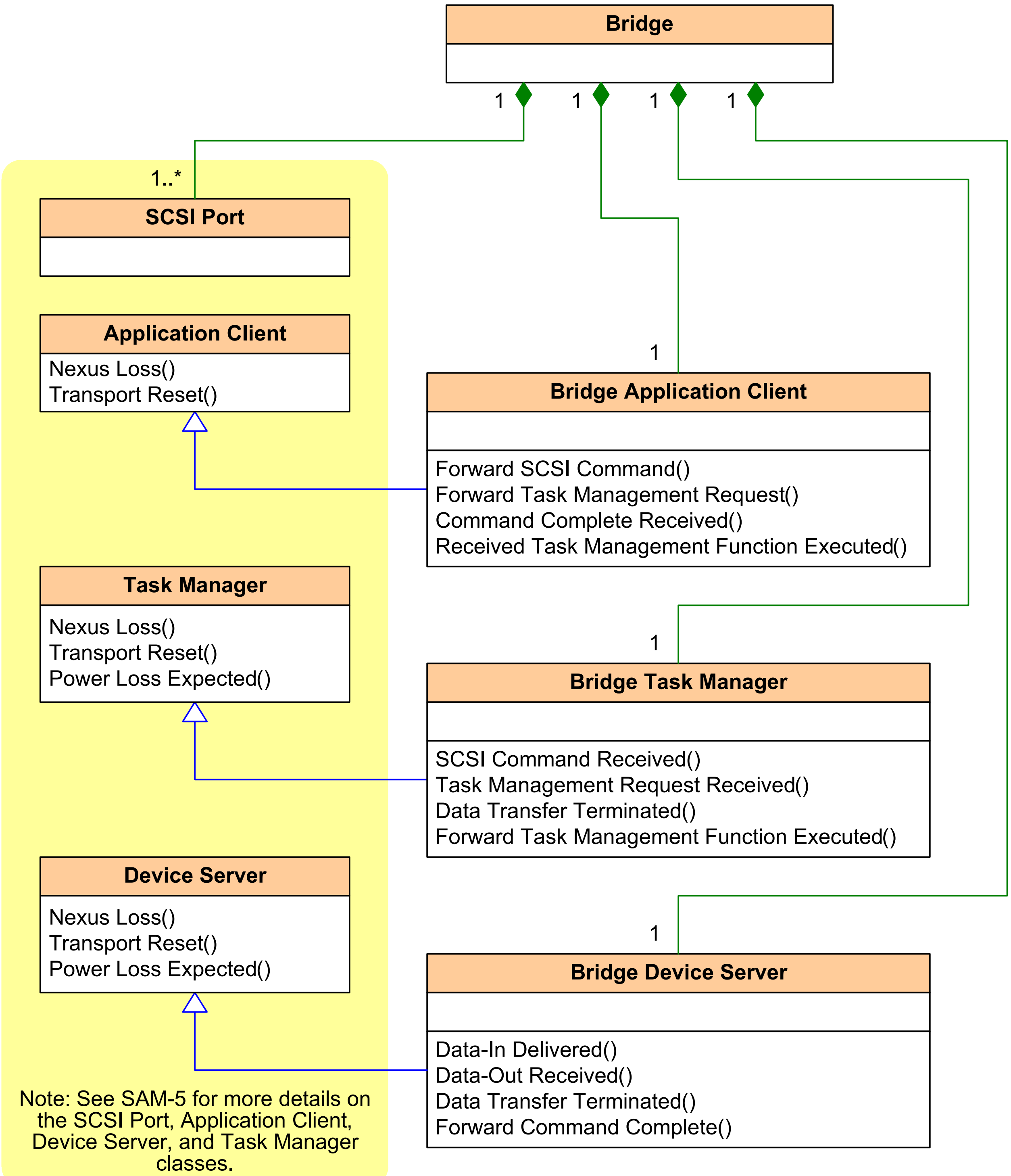


Figure 15 — Bridge class diagram

Each instance of a Bridge class shall contain:

a) one or more SCSI ports;
b) one bridge application client;
c) one bridge task manager; and
d) one bridge device server.

4.1.2.9.2 Bridge Application Client class

4.1.2.9.2.1 Bridge Application Client class overview

~~This standard defines the bridge application client requirements for a bridge that supports initiator mode.~~

The bridge application client~~ requirements for a bridge that supports target mode are outside the scope of this standard.~~:

a) processes requests received from the bridge task manager (see 4.1.2.9.3) and the bridge device server (see 4.1.2.9.4);
b) performs operations for the application client class defined in SAM-5.

4.1.2.9.2.2 Forward SCSI Command operation

The bridge application client processes this operation by invoking the Send SCSI Command operation (see 6.2.1.2) of a SCSI initiator port.

4.1.2.9.2.3 Forward Task Management Request operation

The bridge application client processes this operation by invoking the Send Task Management Request operation (see 6.2.2.2) of a SCSI initiator port.

4.1.2.9.2.4 Command Complete Received operation

The bridge application client processes this operation by invoking the Forward Command Complete operation (see 4.1.2.9.4.5) of the bridge device server.

This operation is inherited from the Application Client class (see SAM-5).

4.1.2.9.2.5 Received Task Management Function Executed operation

The bridge application client processes this operation by invoking the Forward Task Management Function Executed operation (see 4.1.2.9.3.5) of the bridge task manager.

This operation is inherited from the Application Client class (see SAM-5).

4.1.2.9.2.6 Nexus Loss operation

This operation is inherited from the Application Client class (see SAM-5).

4.1.2.9.2.7 Transport Reset operation

This operation is inherited from the Application Client class (see SAM-5).

4.1.2.9.3 Bridge Task Manager class

4.1.2.9.3.1 Bridge Task Manager class overview

~~This standard defines the bridge task manager requirements for a bridge that supports initiator mode.~~

~~The bridge task manager requirements for a bridge that supports target mode are outside the scope of this standard.~~

The Bridge Task Manager class invokes the Terminate Data Transfer operation (see 6.2.1.10) in the SOP target port for each terminate data transfer request received from the SCSI initiator port.

4.1.2.9.3.2 SCSI Command Received operation

The bridge task manager processes this operation by invoking the Forward SCSI Command operation 4.1.2.9.2.2 of the bridge application client.

This operation is inherited from the Task Manager class (see SAM-5).

4.1.2.9.3.3 Task Management Request Received operation

The bridge task manager processes this operation by invoking the Forward Task Management Request operation (see 4.1.2.9.2.3) of the bridge application client.

This operation is inherited from the Task Manager class (see SAM-5).

4.1.2.9.3.4 Data Transfer Terminated operation

The bridge device server processes this by forwarding data transfer terminated information to the SCSI initiator port.

This operation is inherited from the Task Manager class (see SAM-5).

4.1.2.9.3.5 Forward Task Management Function Executed operation

The bridge task manager processes this operation by invoking the Task Management Function Executed operation (see 6.2.2.4) of a SCSI initiator port.

4.1.2.9.3.6 Nexus Loss operation

This operation is inherited from the Task Manager class (see SAM-5).

4.1.2.9.3.7 Transport Reset operation

This operation is inherited from the Task Manager class (see SAM-5).

4.1.2.9.3.8 Power Loss Expected operation

This operation is inherited from the Task Manager class (see SAM-5).

4.1.2.9.4 Bridge Device Server class

4.1.2.9.4.1 Bridge Device Server class overview

~~This standard defines the bridge device server requirements for a bridge that supports initiator mode.~~

~~The bridge device server requirements for a bridge that supports target mode are outside the scope of this standard.~~

The Bridge Device Server class:

a) invokes the Send Data-In operation (see 6.2.1.6) in the SOP target port for each data-in request received;
b) invokes the Request Data-Out operation (see 6.2.1.8) in the SOP target port for each data-out request received from the SCSI initiator port; and
c) invokes the Terminate Data Transfer operation (see 6.2.1.10) in the SOP target port for each terminate data transfer request received from the SCSI initiator port.

4.1.2.9.4.2 Data-In Delivered operation

The bridge device server processes this operation by forwarding data-in delivered information to the SCSI initiator port.

This operation is inherited from the Device Server class (see SAM-5).

4.1.2.9.4.3 Data-Out Received operation

The bridge device server processes this by forwarding data-out received information to the SCSI initiator port.

This operation is inherited from the Device Server class (see SAM-5).

4.1.2.9.4.4 Data Transfer Terminated operation

The bridge device server processes this by forwarding data transfer terminated information to the SCSI initiator port.

This operation is inherited from the Device Server class (see SAM-5).

4.1.2.9.4.5 Forward Command Complete operation

The bridge device server processes this operation by invoking the Send Command Complete operation (see 6.2.1.4) of a SCSI target port.

4.1.2.9.4.6 Nexus Loss operation

This operation is inherited from the Device Server class (see SAM-5).

4.1.2.9.4.7 Transport Reset operation

This operation is inherited from the Device Server class (see SAM-5).

4.1.2.9.4.8 Power Loss Expected operation

This operation is inherited from the Device Server class (see SAM-5).

4.1.2.9.4.9 SCSI Initiator Port class

A SCSI initiator port in an SOP device forwards:

a) data-in requests to the bridge device server;
b) data-out requests to the bridge device server; and
c) terminate data transfer requests to the bridge device server and the bridge task manager.

4.1.2.10 SOP Management Application Client class

The SOP Management Application Client class specifies management tasks to be performed by SOP management device servers.

4.1.2.11 SOP Management Device Server class

The SOP Management Device Server class performs management tasks specified by SOP management application clients.

4.1.3 SOP domain example~~s~~

4.1.3.1 ~~SOP domain example, SOP target device contains a bridge~~

Figure 16 shows an example of a simple SOP domain in which one SOP initiator port is connected to one SOP target port via a SOP service delivery subsystem. ~~In this example, the SOP device containing the SOP target device contains a bridge in initiator mode.~~

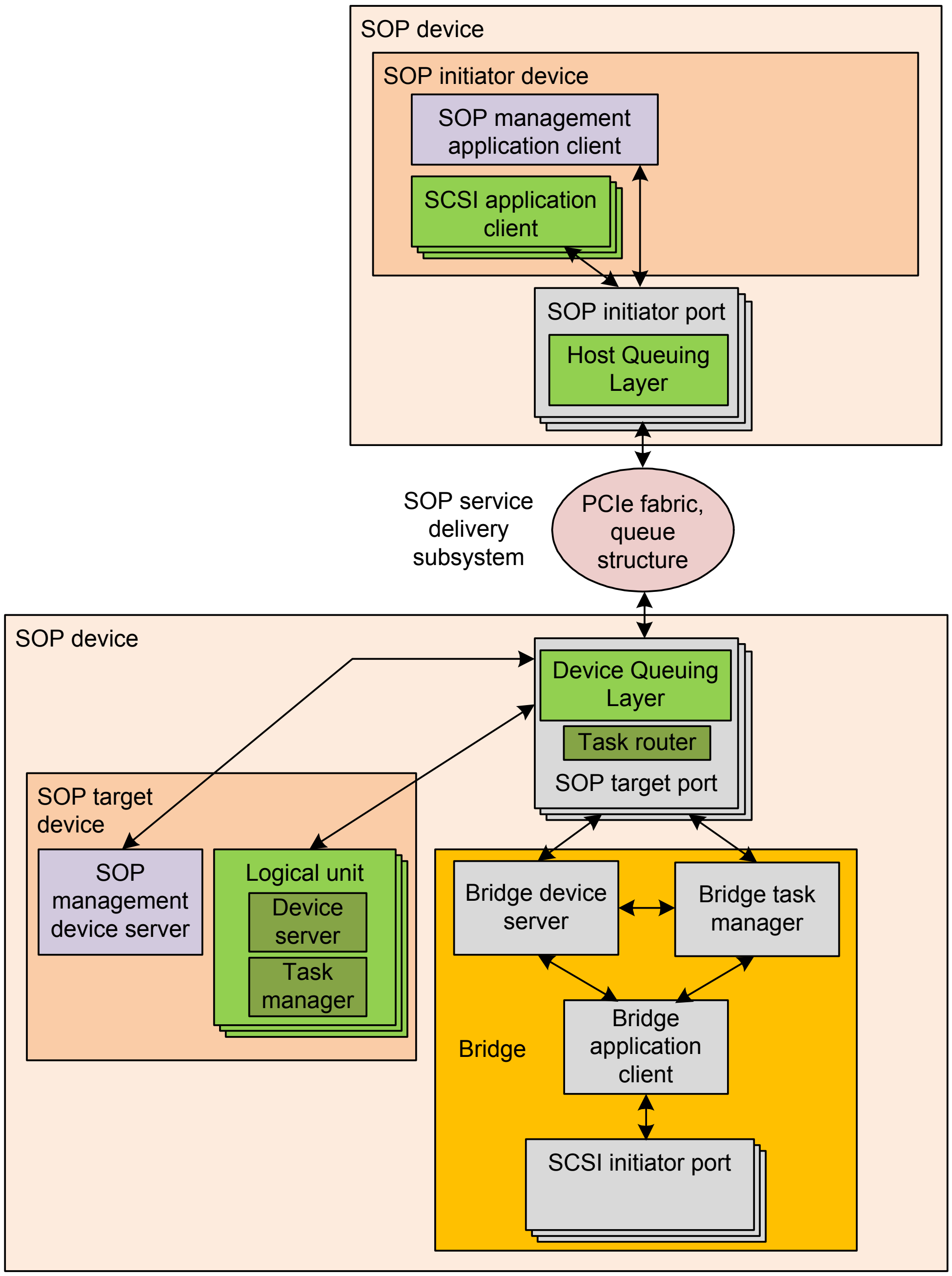


Figure 16 — SOP domain example~~, bridge in initiator mode~~

Figure 17 shows the UML instantiation of the example shown in figure 16.

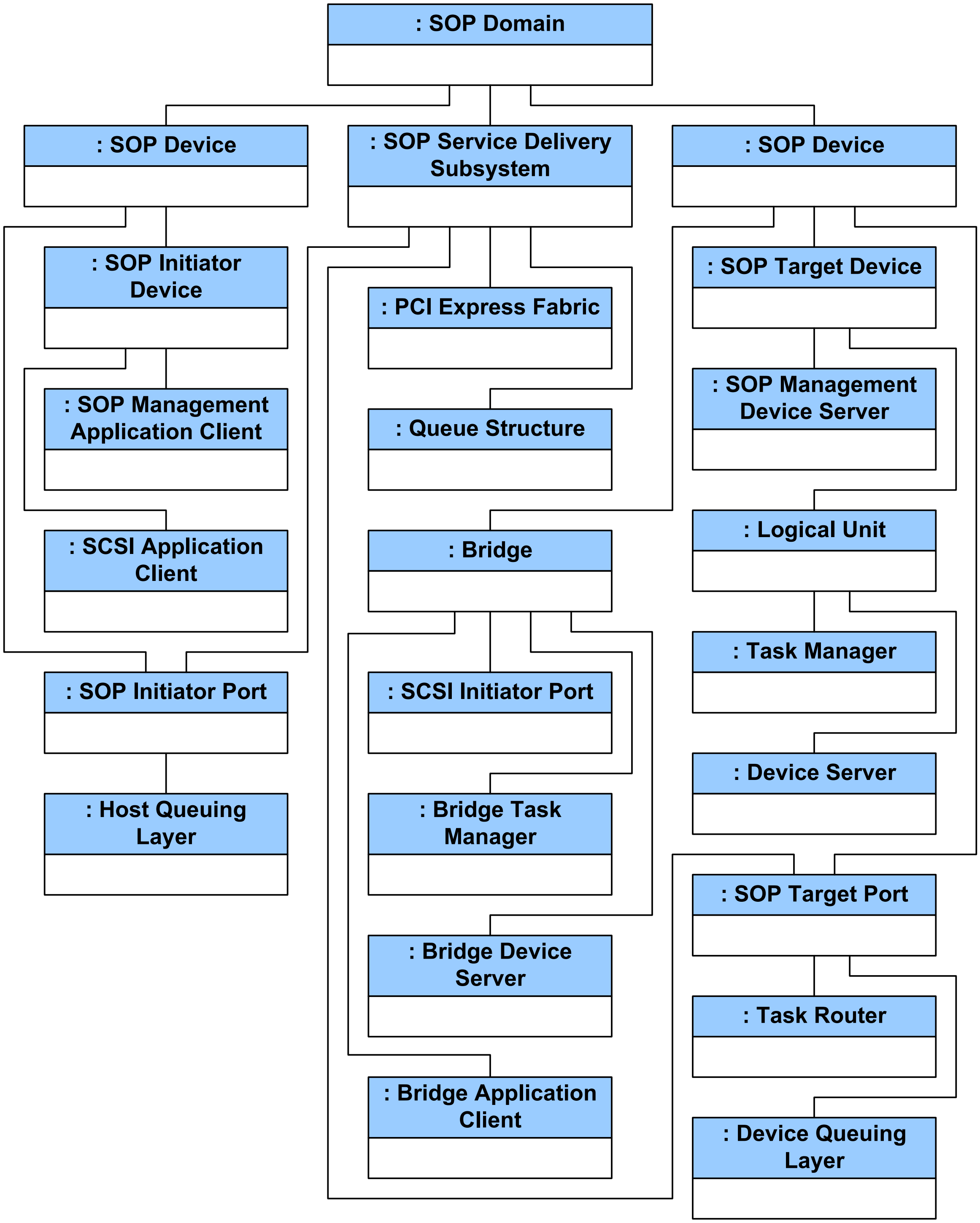


Figure 17 — UML instantiation of SOP domain example~~, bridge in initiator mode~~

4.1.3.2 ~~SOP domain example: SOP initiator device contains a bridge~~

~~Figure 18 shows an example of a simple SOP domain in which one SOP target device is connected to one SOP initiator device via a SOP service delivery subsystem. In this example, the SOP device containing the initiator device contains a bridge in target mode.~~

Editor's Note 2: Figures 18 and 19 were deleted to save space.

Figure 18 — ~~SOP domain example, SOP initiator device contains a bridge~~

~~Figure 19 shows the UML instantiation of the example shown in figure 18.~~

Figure 19 — ~~UML instantiation of SOP domain example, bridge in target mode~~

4.2 Names and identifiers

4.2.1 Names and identifiers overview

Device names are worldwide unique names for devices within a transport protocol. Port names are worldwide unique names for ports within a transport protocol. Port identifiers are the values by which ports are identified within a domain.

Table 16 describes the definitions of names and identifiers for SOP.

Table 16 — Names and identifiers

Attribute	Format	SOP Usage	Reference
Device name	NAA IEEE Registered format identifier (see 4.2.3) or NAA Locally Assigned format identifier (see 4.2.4)	Reported in the Device Identification VPD page (see 6.8.2)	4.2.5
Port Name	NAA IEEE Registered format identifier (see 4.2.3) or NAA Locally Assigned format identifier (see 4.2.4)	Reported in the Device Identification VPD page (see 6.8.2)	4.2.6
Port identifier	Routing ID (see 4.2.8)	Reported in the Device Identification VPD page (see 6.8.2)	4.2.7

Table 17 describes how various SAM-5 attributes are implemented in SOP.

Table 17 — SAM-5 attribute mapping

SAM-5 attribute	SOP implementation
SCSI device name	Device name (see table 16) of an SOP device containing an SOP port
Initiator port identifier	Port identifier (see table 16) of an SOP initiator port
Initiator port name	Port name (see table 16) of an SOP initiator port
Target port identifier	Port identifier (see table 16) of an SOP target port
Target port name	Port name (see table 16) of an SOP target port

4.2.2 PCI Express IDs

See PCI-ID for values that SOP devices may use as Class Codes in configuration space (see PCI).

4.2.3 NAA IEEE Registered format identifier

Table 18 defines the NAA IEEE Registered format identifier used by device names and port names. This format is the same as that defined in SPC-4.

Table 18 — NAA IEEE Registered format

Byte \ Bit	7	6	5	4	3	2	1	0
0	NAA (5h)				(MSB)			
1	IEEE COMPANY ID							
2								
3				(LSB)				
4	VENDOR-SPECIFIC IDENTIFIER							
•••								
7								

The NAA field is set to the value defined in table 18.

The IEEE COMPANY ID field contains a 24-bit canonical form company identifier (i.e., organizationally unique identifier (OUI)) assigned by the IEEE.

Bit 5 of byte 1, which serves as the UNIVERSALLY/LOCALLY ADMINISTERED ADDRESS bit, shall be set to zero.

Bit 4 of byte 1, which serves as the INDIVIDUAL/GROUP ADDRESS bit, shall be set to zero.

NOTE 4 - Information about IEEE company identifiers, the UNIVERSALLY/LOCALLY ADMINISTERED ADDRESS bit, and the INDIVIDUAL/GROUP ADDRESS bit may be obtained from the IEEE Registration Authority web site (see http://standards.ieee.org/regauth/oui).

The VENDOR-SPECIFIC IDENTIFIER field contains a 36-bit value that is assigned by the organization associated with the company identifier in the IEEE COMPANY ID field. The VENDOR-SPECIFIC IDENTIFIER field shall be assigned so the NAA IEEE Registered format identifier is worldwide unique.

4.2.4 NAA Locally Assigned format identifier

Table 19 defines the NAA Locally Assigned format identifier used by device names and port names. This format is the same as that defined in SPC-4.

Table 19 — NAA Locally Assigned format

Bit / Byte	7	6	5	4	3	2	1	0
0	NAA (3h)				LOCALLY ADMINISTERED VALUE			
1								
•••								
7								

The NAA field is set to the value defined in table 19.

The LOCALLY ADMINISTERED VALUE field contains a 60-bit value that is assigned by an administrator to be unique within the set of SCSI domains that are accessible by a common instance of an administration tool or tools.

4.2.5 Device names

Each SOP device shall include an NAA format identifier as its device name. An NAA format identifier used as a device name shall not be used as any other name or identifier (e.g., a port name or logical unit name).

Logical units accessed through SOP target ports report SOP target device names through SCSI vital product data (see 6.8.2).

4.2.6 Port names

Each SOP port shall include an NAA format identifier as its port name. An NAA format identifier used as a port name shall not be used as any other name or identifier (e.g., a device name or logical unit name).

Logical units accessed through SOP target ports report SOP target port names through SCSI vital product data (see 6.8.2).

4.2.7 Port identifiers

Each SOP port shall include a routing ID (see 4.2.8) as its port identifier.

Logical units accessed through SOP target ports report SOP target port identifiers through SCSI vital product data (see 6.8.2).

4.2.8 Routing ID

Table 20 defines the routing ID for a PCI Express device that supports the alternative routing ID Interpretation.

Table 20 — Routing ID for a PCI Express device with alternative routing ID Interpretation

Bit / Byte	7	6	5	4	3	2	1	0
0	BUS NUMBER							
1	FUNCTION NUMBER							

The BUS NUMBER field contains the bus number.

The FUNCTION NUMBER field contains the function number.

Table 21 defines the routing ID for a PCI Express device that does not support the alternative routing ID Interpretation.

Table 21 — Routing ID for a PCI Express device without alternative routing ID Interpretation

Bit / Byte	7	6	5	4	3	2	1	0
0	BUS NUMBER							
1	DEVICE NUMBER					FUNCTION NUMBER		

The BUS NUMBER field contains the bus number.

The DEVICE NUMBER field contains the device number.

The FUNCTION NUMBER field contains the function number.

4.3 Bridge

4.3.1 Bridge overview

An SOP device may contain a bridge to provide access to one or more SCSI domains (e.g., an SOP to SAS bridge).

SCSI ports in the bridge are called local ports. SCSI ports in the SCSI domain accessed through the bridge are called remote ports.

~~Table 22 defines two modes of operation for the bridge.~~

Table 22 — ~~**Bridge modes of operation**~~

~~**Type of SOP device containing the bridge**~~	~~**Type of SOP port interfacing to the bridge**~~	~~**Bridge facilitates communication between**~~		~~**SCSI command and task management function IUs**~~
		~~**SOP port side**~~	~~**Remote port(s) accessed through the local port(s)**~~	
~~SOP initiator device~~	~~SOP initiator port~~	~~Application client(s) in SOP device(s)~~	~~Device server(s) in the SCSI domain(s)~~	~~5.2~~
~~SOP target device~~	~~SOP target port~~	~~Device server(s) in SOP device(s)~~	~~Application client(s) in the SCSI domain(s)~~	~~Outside the scope of this standard~~

Figure 20 shows an example where a SOP device contains a bridge~~ and supports initiator mode~~.

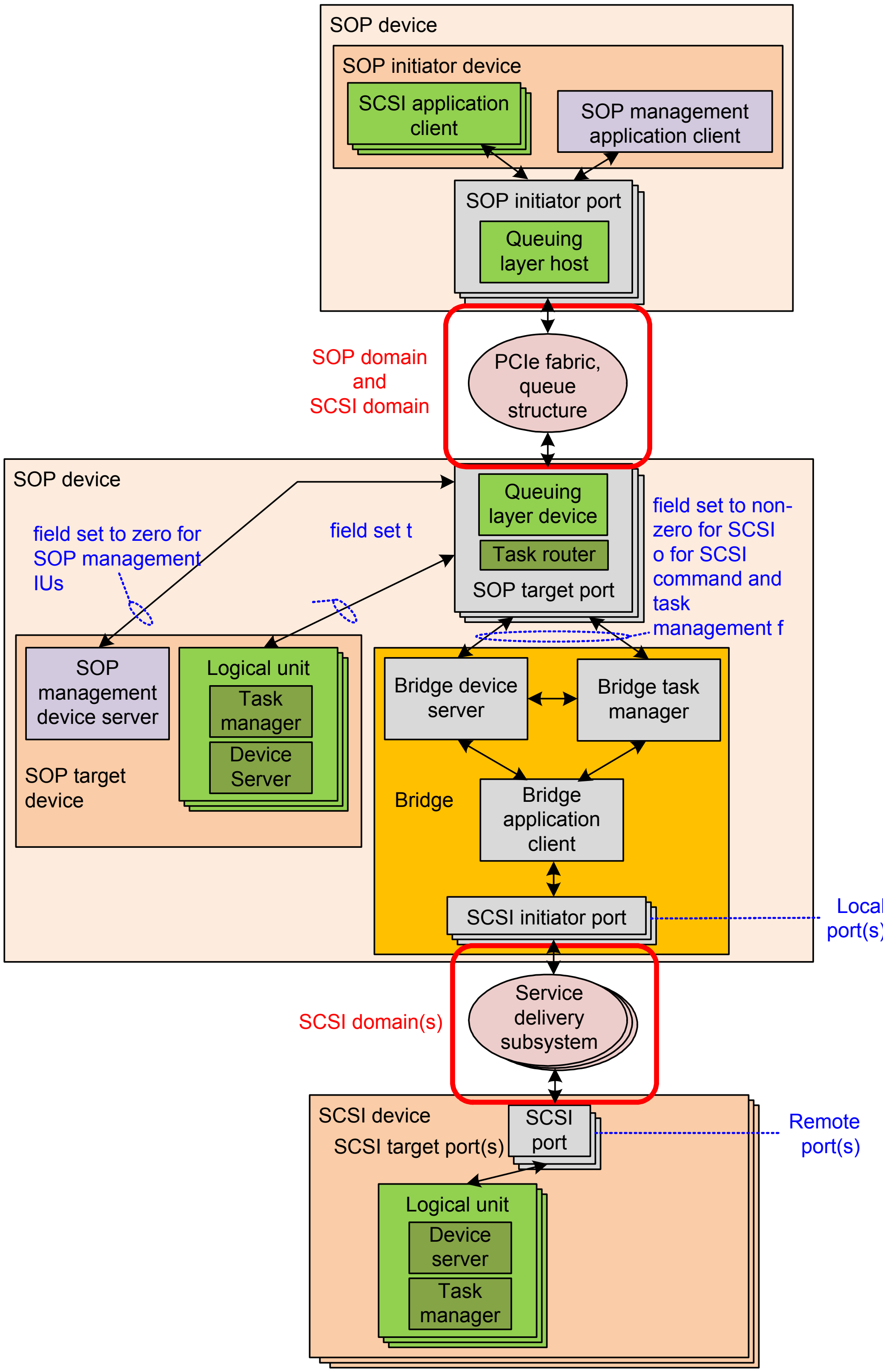


Figure 20 — Example where a SOP device contains a bridge~~ and supports initiator mode~~

~~Figure 21 shows an example where a SOP device contains a bridge and supports target mode.~~

Editor's Note 3: Figures 21 and 22 were deleted to save space.

Figure 21 — ~~Example where a SOP device contains a bridge and supports target mode~~

~~Figure 22 shows the objects used if the SOP device that contains a bridge shown in figure 20 is acting as an SOP initiator device.~~

Figure 22 — ~~SOP initiator device containing a bridge~~

The bridge may or may not make logical units contained inside the same SOP device as the bridge accessible to remote SCSI initiator ports. Indication and control of such a capability is outside the scope of this standard.

The bridge may or may not support bridging between the SCSI domains containing the remote ports (i.e., allowing an application client in one SCSI domain to access a device server in another SCSI domain). Indication and control of such a capability is outside the scope of this standard.

4.3.2 Nexus identifier

The bridge shall maintain a non-zero 16-bit nexus identifier for every I_T nexus in which it is able to participate on the far side of the bridge (i.e., involving a local port and a remote port). The nexus identifier is included in SCSI command request IUs and TASK MANAGEMENT IUs (see 5.2). A nexus identifier of 0000h is used to address logical units that are not behind the bridge (i.e., are contained within the SOP device).

NOTE 5 - This standard does not define how a bridge detects I_T nexuses and assigns nexus identifiers (e.g., automatically detected by the bridge due to a SAS discover process or an FCP login, or programmed by an application client using vendor-specific SOP IUs).

Each nexus identifier value from 0001h through FFFFh is either:

a) assigned for an I_T nexus that is not lost;
b) assigned for an I_T nexus that is lost; or
c) not assigned.

If a local port is used as both a SCSI initiator port and a SCSI target port, then the bridge shall assign different nexus identifiers for each of those roles. Figure 23 shows an example of two nexus identifiers being assigned for one SCSI port.

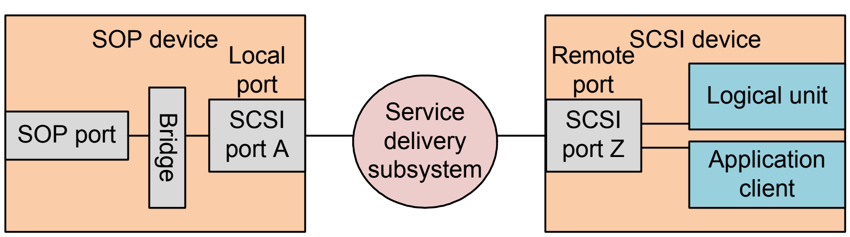


Figure 23 — Two nexus identifiers for a SCSI port example

If SCSI port A and SCSI port Z both support being SCSI initiator ports and SCSI target ports, then the bridge assigns two nexus identifiers for I_T nexuses involving those SCSI ports:

a) one representing SCSI initiator port A and SCSI target port Z; and
b) another representing SCSI target port A and SCSI initiator port Z.

Once a nexus identifier is assigned, the bridge shall maintain that nexus identifier until the nexus identifier is removed with a REMOVE BRIDGE NEXUS IU (see 5.2.6.5) or until a hard reset.

If the bridge detects an I_T nexus loss for an I_T nexus represented by a nexus identifier, then the bridge shall report that nexus identifier with the NEXUS LOST bit set to one in the REPORT BRIDGE NEXUSES IU (see 5.2.6.3). If the nexus identifier is still assigned and the I_T nexus is reestablished, then the bridge shall report the nexus identifier with the NEXUS LOST bit set to zero in the REPORT BRIDGE NEXUSES IU.

The REMOVE BRIDGE NEXUSES IU supports removing a nexus identifier for a nexus identifier that is lost. Removal of a nexus identifier for an I_T nexus that is not lost is outside the scope of this standard.

4.3.3 Nexus generation count

The bridge shall maintain a non-zero 32-bit bridge nexus generation count. The bridge shall set the bridge nexus generation count to at least 00000001h at power on. The bridge shall increment the bridge nexus generation count by at least one every time that it changes the nexus identifiers, wrapping to at least 00000001h after reaching the maximum value (i.e., FFFFFFFFh).

NOTE 6 - Application clients that use the bridge nexus generation count should read the value often enough to ensure that the value does not increment a multiple of 4 294 967 295 times.

4.3.4 Forwarding by a bridge supporting a SCSI initiator port

The bridge device server, bridge task manager, and bridge application client shall forward SCSI commands by mapping the transport protocol services as described in table 23.

Table 23 — Execute Command procedure call mapping by a bridge

SOP target port side transport protocol service	Class	Dir	SCSI initiator port side transport protocol service	Class	Ref
Command and status					
SCSI Command Received indication	BTM	⇒	Send SCSI Command request	SIP	4.3.5
Send Command Complete response	STP	⇐	Command Complete Received confirmation	BAC	4.3.6
Data-in transfer					
Send Data-In request	STP	⇐	None [a] [b]		
Data-In Delivered confirmation	BDS	⇒	None [a] [c]		
Data-out transfer					
Receive Data-Out request	STP	⇐	None [a] [d]		
Data-Out Received confirmation	BDS	⇒	None [a] [e]		
Terminate data transfer					
Terminate Data Transfer request	STP	⇐	None [a]		
Data Transfer Terminated confirmation	BSM	⇒	None [a]		
Task management					
Task Management Request Received indication	BTM	⇒	Send Task Management Request request	SIP	4.3.7
Task Management Function Executed response	STP	⇐	Received Task Management Function Executed confirmation	BAC	4.3.8

Class key:
- Dir = Direction
- Ref = Reference
- BAC = bridge application client
- BDS = bridge device server
- BTM = bridge task manager
- BSM = bridge device server or bridge task manager
- SIP = SCSI initiator port
- STP = SCSI target port

[a] Data transfer transport protocol services for SCSI initiator ports are not specified by SAM-5.
[b] The addresses for data-in received by the SCSI port (e.g., in read DATA IUs) are calculated according to the command's scatter gather list.
[c] The SCSI port may acknowledge reception of data-in before forwarding.
[d] The addresses for data-out requests received by the SCSI port (e.g., in XFER_RDY IUs) are calculated according to the command's scatter gather list.
[e] Data-out is sent by the SCSI port as data-out transfers (e.g., write DATA IUs).

4.3.5 SCSI Command Received to Send SCSI Command mapping

If a bridge task manager processes the following SCSI transport protocol service indication:

SCSI Command Received (IN (I_T_L_Q Nexus, CDB, Task Attribute, [CRN], [Command Priority], [First Burst Enabled])),

then the bridge application client shall invoke the Send SCSI Command SCSI transport protocol service request with the arguments described in table 24.

Table 24 — SCSI Command Received to Send SCSI Command argument mapping

Argument	Value
I_T_L_Q Nexus	Set as follows: a) I specifying the local port; b) T specifying the remote port; c) L set to the corresponding value from the SCSI Command Received() I_T_L_Q Nexus argument; and d) Q set to a command identifier mapped from the SCSI Command Received() I_T_L_Q Nexus argument request identifier (see 4.3.9)
CDB	Set to the corresponding value from SCSI Command Received()
Task Attribute	Set to the corresponding value from SCSI Command Received()
[Data-In Buffer Size]	Set to: a) if the command was received in a LIMITED COMMAND IU or a COMMAND IU and the DATA DIRECTION field is set to 10b (i.e. data-in), then the DATA TRANSFER SIZE field; b) if the command was received in an EXTENDED COMMAND IU, then the DATA-IN TRANSFER SIZE field; or c) if neither a) nor b) is true, then nothing (i.e., no argument)
[Data-Out Buffer]	If the command has data-out, then set to the data-out buffer contents (e.g., represented by the data-out buffer address)
[Data-Out Buffer Size]	Set to: a) if the command was received in a LIMITED COMMAND IU or a COMMAND IU and the DATA DIRECTION field is set to 01b (i.e. data-out), then the DATA TRANSFER SIZE field; b) if the command was received in an EXTENDED COMMAND IU, then the DATA-OUT TRANSFER SIZE field; or c) if neither a) nor b) is true, then nothing (i.e., no argument)
[CRN]	Set to the corresponding value, if any, from SCSI Command Received()
[Command Priority]	Set to the corresponding value, if any, from SCSI Command Received()
[First Burst Enabled]	Set to the corresponding value, if any, from SCSI Command Received()

4.3.6 Command Complete Received to Send Command Complete mapping

If a bridge device server processes the following SCSI transport protocol service confirmation:

> Command Complete Received (IN (I_T_L_Q Nexus, [Data-In Buffer], [Sense Data], [Sense Data Length], Status, Service Response, [Status Qualifier])),

then the bridge application client shall invoke the Send Command Complete SCSI transport protocol service response with the arguments described in table 25.

Table 25 — Command Complete Received to Send Command Complete argument mapping

Argument	Value
I_T_L_Q Nexus	Set as follows: a) I specifying the SOP initiator port from which the command was received; b) T specifying the SOP target port through which the command was received; c) L set to the corresponding value from the Command Complete Received() I_T_L_Q Nexus argument; and d) Q set to the request identifier of the command
[Data-In Buffer]	If the command has data-in, then set to data-in buffer contents (e.g., represented by the data-in buffer address).
[Sense Data]	Set to the corresponding value, if any, from SCSI Command Received()
[Sense Data Length]	Set to the corresponding value, if any, from SCSI Command Received()
Status	Set to the corresponding value from SCSI Command Received()
Service Response	Set to the corresponding value from SCSI Command Received()
[Status Qualifier]	Set to the corresponding value, if any, from SCSI Command Received()

4.3.7 Task Management Request Received to Send Task Management Request mapping

If a bridge task manager processes the following SCSI transport protocol service indication:

Task Management Request Received (IN (Nexus, Function Identifier)),

then the bridge application client shall invoke the Send Task Management Request SCSI transport protocol service request with the arguments described in table 26.

Table 26 — Task Management Request Received to Send Task Management Request argument mapping

Argument	Value
I_T_L_Q Nexus	Set as follows: a) I specifying the local port; b) T specifying the remote port; c) L set to the corresponding value from the Task Management Request Received() I_T_L_Q Nexus argument; and d) Q set to a command identifier mapped from the Task Management Request Received() I_T_L_Q Nexus argument request identifier (see 4.3.9)
Function Identifier	Set to the corresponding value from Task Management Request Received()

4.3.8 Received Task Management Function Executed to Task Management Function Executed mapping

If a bridge application client processes the following SCSI transport protocol service confirmation:

Received Task Management Function Executed (IN (Nexus, Service Response, [Additional Response Information])),

then the bridge task manager shall invoke the Task Management Function Executed SCSI transport protocol service response with the arguments described in table 27.

Table 27 — Received Task Management Function Executed to Task Management Function Executed argument mapping

Argument	Value
Nexus	Set as follows: a) I specifying the SOP initiator port from which the task management function was received; b) T specifying the SOP target port that received the task management function; c) L set to the corresponding value, if any, from the Received Task Management Function Executed() Nexus argument; and d) Q set to the request identifier of the task management function
Service Response	Set to the corresponding value from Received Task Management Function Executed()
[Additional Response Information]	Set to the corresponding value, if any, from Received Task Management Function Executed()

4.3.9 Request identifier mapping

The bridge device server, bridge task manager, and bridge application client shall map request identifiers used by the SOP port to command identifiers and task management associations used by the local port based on the SCSI transport protocol of the local port as defined in B.2.

4.3.10 ~~Forwarding by a bridge supporting a SCSI target port~~

~~This standard does not define the behavior of bridges supporting SCSI target ports (i.e., target mode).~~

4.4 PCI Express queuing layer services

The PCI Express queuing layer defines:

a) the queue structure, which:
   A) contains inbound queues and outbound queues; and
   B) is represented by the Queue Structure class (see 4.1.2.2.3);
b) the host queuing layer, which:
   A) provides operations for an SOP port to access the inbound queues and outbound queues; and
   B) is represented by the Host Queuing Layer class (see 4.1.2.8.3);

   and

c) the device queuing layer, which:
   A) provides operations for an SOP port to access the inbound queues and outbound queues; and
   B) is represented by the Device Queuing Layer class (see 4.1.2.8.4).

An inbound queue is a queue that is used to transfer SOP IUs (see 5.2) from a host queuing layer to a device queuing layer.

An outbound queue is a queue that is used to transfer SOP IUs (see 5.2) from a device queuing layer to a host queuing layer.

See A.1 for the description of how PQI implements inbound queues and outbound queues.

Each queue consists of queue elements. An SOP IU either:

a) fits in one queue element; or
b) if supported by the queuing layer, spans multiple queue elements.

All bytes in a queue element following an IU shall be ignored by the recipient.

If an outbound queue is deleted and there are one or more commands in the task set specifying that outbound queue, then the logical unit shall

a) abort all those commands;
b) create a unit attention condition with the additional sense code set to SOME COMMANDS CLEARED BY QUEUING LAYER EVENT.

NOTE 7 - If all outbound queues are deleted, then an I_T nexus loss condition exists (see 4.7.2).

4.5 Data transfers

Data (e.g., user data and protection information, if any, for a WRITE command, or parameter data for a MODE SENSE command) is transferred to and from data buffers. An application client in an SOP initiator device uses the data buffer descriptor area(s) specified in each IU requesting a data transfer to specify the location and size of one or more data buffers to which and/or from which data is to be transferred for the IU.

For an IU specifying a data-out operation (e.g., an IU containing a WRITE command or a MODE SELECT command), the application client writes the data for the IU to the Data-Out Buffer before writing the IU to the inbound queue. When the device server processes the IU, the SOP target port reads the data from the Data-Out Buffer, and the SOP target port writes a response IU to the specified outbound queue.

For an IU specifying a data-in operation (e.g., an IU containing a READ command or an INQUIRY command), the SOP target port writes the data requested by the IU to the Data-In Buffer. When all data has been written for the IU, the SOP target port writes a response IU to the specified outbound queue. When the application client processes the response IU, the application client reads the data from the Data-In Buffer.

For an IU specifying a data-out operation and a data-in operation (e.g., an IU containing an XDWRITEREAD command), the application client writes the data to the Data-Out Buffer before writing the IU to the inbound queue. When the device server processes the IU, the SOP target port reads data from the Data-Out Buffer, and the SOP target port writes data to the Data-In Buffer. When all data for the IU has been written and read, the SOP target port writes a response IU to the specified outbound queue. When the application client processes the response IU, the application client reads the data from the Data-In Buffer.

4.6 Request identifier

SCSI command request IUs (see 5.2.5), the TASK MANAGEMENT request IU (see 5.2.5.4), and SOP management request IUs (see 5.2.7) each contain a REQUEST IDENTIFIER field specifying a request identifier that the SOP target port shall use in the response IUs for those the request IUs. The request identifier allows the SOP initiator port to determine the context for each response IU.

The request identifier uniqueness requirements are as follows:

a) for a SCSI command or task management function, the request identifier shall be unique across:
   A) all commands and task management functions to the same SOP target port specifying the same nexus identifier and any LUN; and
   B) if the nexus identifier is 0000h, then all SOP management functions to the same SOP target port;

   and

b) for an SOP management function, the request identifier shall be unique across:
   A) all commands and task management functions to the same SOP target port specifying a nexus identifier of 0000h and any LUN; and
   B) all SOP management functions to the same SOP target port.

NOTE 8 - The NEXUS IDENTIFIER field that exists in some SOP management function request IUs is not used for selecting the destination of the request IU, and does not factor into the uniqueness requirements for the request identifier.

EXAMPLE - If a SCSI request identifier is used for a SCSI command to a LUN behind an SOP target port, then it is neither used for a concurrent task management function to any other LUN behind that SOP target port nor used for a concurrent SOP management function to the management device server behind that SOP target port.

The SOP initiator port shall not reuse a request identifier until it receives indication from the SOP target port that the request identifier is no longer in use.

The request identifier in a SCSI command request IU contains the command identifier defined in SAM-5.

The request identifier in a TASK MANAGEMENT REQUEST IU is the association between a Send Task Management Request operation (see 6.2.2.2) and a Received Task Management Function Executed operation (see 6.2.2.5).

Request identifier overlap handling is described in 4.6.

4.7 SCSI events

4.7.1 Hard reset

Each of the following PCI Express reset events is considered to be a SCSI reset event (e.g., generating a Transport Reset event notification to the SCSI application layer) (see 6.2.3.1 and SAM-5):

a) PCI Express cold reset;
b) PCI Express function-level reset;

c) PCI Express hard reset;
d) PCI Express hot reset;
e) PCI Express soft reset; and
f) PCI Express warm reset.

See A.5 for queuing layer events that are considered to be reset events.

After processing a hard reset, each logical unit to which the SOP target port that detected the reset event has access shall establish a unit attention condition for all SOP initiator ports with the additional sense code set to SCSI BUS RESET OCCURRED (see SAM-5 and SPC-4).

A bridge device server, bridge task manager, and bridge application client shall process a hard reset by deleting all outstanding request identifier mappings associated with that SOP port receiving the hard reset.

4.7.2 I_T nexus loss

There is no PCI Express event that is considered to be an I_T nexus loss event (e.g., generating a Nexus Loss event notification to the SCSI application layer) (see 6.2.3.2 and SAM-5).

See A.6 for queuing layer events that are considered to be I_T nexus loss events.

4.7.3 Power loss expected

There is no PCI Express event that is considered to be a power loss expected event (e.g., generating a Power Loss Expected event notification to the SCSI application layer) (see 6.2.3.3 and SAM-5).

4.8 Null data field termination and zero padding requirements

A data field that is described as being null-terminated shall have one byte containing an ASCII or UTF-8 null (00h) character in the last used byte (i.e., highest offset) of the field and no other bytes in the field shall contain the ASCII/UTF-8 null character.

A data field may be specified to be a fixed length. The length specified for a data field may be greater than the length required to contain the contents of the field. A data field may be specified to have a length that is a multiple of a given value (e.g., a multiple of four bytes). When such fields are described as being null-padded, the bytes at the end of the field that are not needed to contain the field data shall contain ASCII or UTF-8 null (00h) characters. When such fields are described as being zero-padded, the bytes at the end of the field that are not needed to contain the field data shall contain zeros.

NOTE 9 - There is no difference between the pad byte contents in null-padded and zero-padded fields. The difference is in the format of the other bytes in the field.

A data field that is described as being both null-terminated and null-padded shall have at least one byte containing an ASCII or UTF-8 null (00h) character in the end of the field (i.e., highest offset) and may have more than one byte containing ASCII or UTF-8 null characters, if needed, to meet the specified field length requirements. If more than one byte in a null-terminated, null-padded field contains the ASCII or UTF-8 null character, then all the bytes containing the ASCII or UTF-8 null character shall be at the end of the field (i.e., only the highest offsets).

4.9 Asynchronous events

4.9.1 Asynchronous events overview

Asynchronous events are:

a) managed with the REPORT EVENT CONFIGURATION IU (see 5.2.4.4) and the SET EVENT CONFIGURATION IU (see 5.2.4.5);
b) reported with the EVENT IU (see 5.2.7.2) on a specified outbound queue; and
c) if requested, acknowledged with the EVENT ACKNOWLEDGE IU (see 5.2.7.3) on any inbound queue.

Asynchronous events are reported using four layers of event information:

1) 1-byte event type;
2) 2-byte event identifier;
3) 4-byte additional event identifier; and
4) variable-length event data, if any.

The event identifier is defined in the description of the event type.

The additional event identifier is defined in the description of the event identifier.

The event data, if any, is defined in the description of the additional event identifier.

Table 28 describes the event types.

Table 28 — Event types

Code	Description	Support	Reference
00h	Reserved		
Queuing layer events			
01h	INBOUND QUEUE ERROR	M	4.9.2
02h to 2Fh	Reserved		
SOP device events			
30h to 5Fh	Reserved		
SOP target device events			
60h to 7Fh	Reserved		
Bridge events			
80h	Bridge local port detected error event	BR	B.6
81h to FFh	Reserved		
Support key: M = mandatory. BR = mandatory if the SOP device includes a bridge (see 4.3); prohibited if the SOP device does not include a bridge			

If an outbound queue to which event types are assigned is deleted, then reporting of events those event types shall be disabled.

If the management device server requests an event acknowledgement while reporting an event, then it may not report another similar event until it receives the EVENT ACKNOWLEDGE IU.

4.9.2 INBOUND QUEUE ERROR event type

The INBOUND QUEUE ERROR event type is used to report an error that has caused the SOP port in the SOP device containing a SOP management device server to stop consuming from an inbound queue.

Table 29 defines the EVENT IDENTIFIER field for the INBOUND QUEUE ERROR event type.

Table 29 — EVENT IDENTIFIER field for the INBOUND QUEUE ERROR event type

Code	Name	Description
0000h	UNKNOWN ERROR	An inbound queue has failed for an unknown reason and the SOP target port is no longer consuming from it.
0001h	INVALID IU TYPE	An inbound queue has failed because it contained an IU with an invalid IU type (e.g., reserved or not supported).
0002h	INVALID IU LENGTH	An inbound queue has failed because it contained an IU with an invalid IU length.
0003h to FFFEh	Reserved	
FFFFh	Vendor specific	

Table 30 defines the event data for the INBOUND QUEUE ERROR event type.

Table 30 — Event data for the INBOUND QUEUE ERROR event type

Bit / Byte	7	6	5	4	3	2	1	0
0 ••• 7	Queuing-layer specific inbound queue error information							
8 ••• n	LEADING BYTES OF BAD IU							

The queuing-layer specific inbound queue error information is defined in A.8.

The LEADING BYTES OF BAD IU field indicates the value of the first bytes in the bad IU. If the IU length is shorter than the length indicated by the IU LENGTH field, then this field shall be padded with zeros.

NOTE 10 - The first byte contains the IU TYPE field of the bad IU, the second byte contains the COMPATIBLE FEATURES field of the bad IU, etc.

4.9.3 Bridge local port detected error event

For an event type of 80h (i.e., bridge local port detected error event):

a) if the event is for a SCSI command or task management function, then the event shall be reported on that response queue rather than the event queue specified in the SET EVENT CONFIGURATION function;

b) if the event is not for a SCSI command or task management function, then the event shall be reported on the event queue specified in the SET EVENT CONFIGURATION function; and
c) the EVENT IDENTIFIER field, ADDITIONAL EVENT IDENTIFIER field, and event data are defined in B.6.

4.10 Parameter rounding

Certain parameters sent to a SOP management device server with various commands contain a range of values. SOP management device servers may choose to implement only selected values from this range. If the SOP management device server receives a value that it does not support, it either rejects the command (i.e., return a MANAGEMENT RESPONSE IU with the RESULT CODE field set to INVALID FIELD IN REQUEST IU) or it rounds the value received to a supported value.

If parameter rounding is implemented, a SOP management device server that receives a parameter value that is not an exact supported value shall adjust the value to one that it supports and shall return a MANAGEMENT RESPONSE IU with the RESULT CODE field set to GOOD. The SOP management application client should send an appropriate SOP management function to learn what value the SOP management device server has selected.

The SOP management device server shall reject unsupported values unless rounding is permitted in the description of the parameter. When the description of a parameter states that rounding is permitted, the SOP management device server should adjust maximum-value fields down to the next lower supported value than the one specified by the SOP management application client. Minimum-value fields should be rounded up to the next higher supported value than the one specified by the SOP management application client. In some cases, the type of rounding (i.e., up or down) is described in the definition of the parameter.

4.11 PCI Express Configuration

This subclause describes information associated with the PCIe configuration of a SOP device to support this standard.

A SOP port is a PCI Function (see PCIe).

A SOP port shall be a PCI Express Endpoint (see PCIe) or a Root Complex Integrated Endpoint (see PCIe).

If a SOP port is a PCI function in a PCI Multi-Function device that contains more than eight PCI functions, then the PCI Multi-Function device shall support the Alternative Routing-ID Interpretation (see PCIe) (i.e., be an ARI Device, support the ARI Capability, and not support Phantom Functions).

5 SOP information unit layer

5.1 SOP information unit layer overview

The SOP information unit layer defines the structures and protocols for transmission of device service requests, device service responses, task management requests, and task management responses between SCSI initiator devices and SCSI target devices over a queuing layer. There are also special IUs used for general management and bridge management.

5.2 SOP IU definitions

5.2.1 SOP IU definitions overview

Table 31 describes the SOP IU types.

Table 31 — SOP IU types (part 1 of 3)

IU type	Name	Minimum length [b]	Queue type	Support	Reference
00h	NULL	4	IO	TM	5.2.3
Request IUs					
General management request IUs					
01h	REPORT GENERAL	16	I	M	5.2.4.1
02h	REPORT CONFIGURATION	32	I	M	5.2.4.2
03h	SET CONFIGURATION	32	I	M	5.2.4.3
04h	REPORT EVENT CONFIGURATION	16	I	M	5.2.4.4
05h	SET EVENT CONFIGURATION	16	I	M	5.2.4.5
06h to 0Fh	Reserved				

Note - This standard does not use the 60h to 6Fh and E0h to EFh ranges, which some queuing layers (e.g., PQI) use for administrative IUs.

Queue type key:
- I = inbound queue.
- O = outbound queue.
- IO = inbound queue and outbound queue.

Support key [a]:
- M = mandatory.
- TM = mandatory if the SOP device is a SOP target device, prohibited if it is not.
- TO = optional if the SOP device is a SOP target device, prohibited if it is not.
- BR = mandatory if the SOP device includes a bridge (see 4.3); prohibited if the SOP device does not include a bridge.

[a] Mandatory, optional, and prohibited are from the perspective of the SOP device receiving the IU.
[b] The minimum length is measured in bytes. The value in this column is the length of the entire IU, not the value in the IU LENGTH field.

Table 31 — SOP IU types (part 2 of 3)

IU type	Name	Minimum length [b]	Queue type	Support	Reference
SCSI command and task management function request IUs for SOP target devices					
10h	LIMITED COMMAND	32	I	TO	5.2.5.1
11h	COMMAND	64	I	TM	5.2.5.2
12h	EXTENDED COMMAND	56	I	TO	5.2.5.3
13h	TASK MANAGEMENT	32	I	TM	5.2.5.4
14h to 1Fh	Reserved				
Bridge management request IUs					
20h	REPORT BRIDGE LOCAL PORTS	16	I	BR	5.2.6.1
21h	REPORT BRIDGE LOCAL PORT DETAILS	16	I	BR	5.2.6.2
22h	REPORT BRIDGE NEXUSES	16	I	BR	5.2.6.3
23h	REPORT BRIDGE NEXUS DETAILS	16	I	BR	5.2.6.4
24h	REMOVE BRIDGE NEXUS	16	I	BR	5.2.6.5
25h to 2Fh	Reserved				
Other request IUs					
30h to 6Fh	Reserved				
70h to 7Fh	Vendor specific request IUs				
Response IUs					
General response IUs					
80h	Reserved				

Note - This standard does not use the 60h to 6Fh and E0h to EFh ranges, which some queuing layers (e.g., PQI) use for administrative IUs.

Queue type key:
- I = inbound queue.
- O = outbound queue.
- IO = inbound queue and outbound queue.

Support key [a]:
- M = mandatory.
- TM = mandatory if the SOP device is a SOP target device, prohibited if it is not.
- TO = optional if the SOP device is a SOP target device, prohibited if it is not.
- BR = mandatory if the SOP device includes a bridge (see 4.3); prohibited if the SOP device does not include a bridge.

[a] Mandatory, optional, and prohibited are from the perspective of the SOP device receiving the IU.
[b] The minimum length is measured in bytes. The value in this column is the length of the entire IU, not the value in the IU LENGTH field.

Table 31 — SOP IU types (part 3 of 3)

IU type	Name	Minimum length [b]	Queue type	Support	Reference
81h	MANAGEMENT RESPONSE	16	O	M	5.2.7.1
82h	EVENT	16	O	M	5.2.7.2
83h	EVENT ACKNOWLEDGE	16	I	M	5.2.7.3
84h to 8Fh	Reserved				
SCSI command and task management function response IUs for SOP target devices					
90h	SUCCESS	16	O	TM	5.2.8.1
91h	COMMAND RESPONSE	32	O	TM	5.2.8.2
92h	Reserved				
93h	TASK MANAGEMENT RESPONSE	16	O	TM	5.2.8.3
94h to 9Fh	Reserved				
Other response IUs					
A0h to EFh	Reserved				
F0h to FFh	Vendor specific response IUs				

Note - This standard does not use the 60h to 6Fh and E0h to EFh ranges, which some queuing layers (e.g., PQI) use for administrative IUs.

Queue type key:
- I = inbound queue.
- O = outbound queue.
- IO = inbound queue and outbound queue.

Support key [a]:
- M = mandatory.
- TM = mandatory if the SOP device is a SOP target device, prohibited if it is not.
- TO = optional if the SOP device is a SOP target device, prohibited if it is not.
- BR = mandatory if the SOP device includes a bridge (see 4.3); prohibited if the SOP device does not include a bridge.

[a] Mandatory, optional, and prohibited are from the perspective of the SOP device receiving the IU.
[b] The minimum length is measured in bytes. The value in this column is the length of the entire IU, not the value in the IU LENGTH field.

5.2.2 IU header

All IUs include the header defined in table 32.

Table 32 — IU header

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH							(LSB)
3	(MSB)							

Byte labels for the IU header are shaded in tables defining IUs.

The IU TYPE field contains the SOP IU type (see table 31).

The COMPATIBLE FEATURES field shall be set to the value shown in table 32 and shall be ignored by the recipient.

NOTE 11 - Future versions of this standard may redefine bits in the COMPATIBLE FEATURES field, with the expectation that SOP ports compliant with this standard ignore them.

The IU LENGTH field contains the number of bytes following the IU header.

If the IU LENGTH field is greater than the value specified in the IU definition then the SOP port shall ignore the additional bytes.

NOTE 12 - Future versions of this standard may add additional fields to an IU by increasing the IU length, with the expectation that SOP ports compliant with this standard ignore the additional bytes.

Table 33 defines how an SOP target port and an SOP initiator port handle errors in the common IU header.

Table 33 — Common IU header error handling

Common IU header problem	SOP target port handling	SOP initiator port handling
The IU TYPE field is set to a reserved or unsupported value	The SOP target port shall: a) stop consuming from the inbound queue; and b) if configured, return an event with the EVENT TYPE field set to INBOUND QUEUE ERROR and the EVENT IDENTIFIER field set to INVALID IU TYPE	The SOP initiator port should: a) stop consuming from the outbound queue; and b) delete the outbound queue
The IU LENGTH field is set to a supported value and either: a) the IU LENGTH field bits 1:0 are not set to 00b (i.e., the IU length is not a multiple of 4); b) the IU LENGTH field is greater than 4 092 (i.e., the IU length is larger than 4 096 bytes); c) the IU LENGTH field is greater than the largest IU size supported by the queue; d) the IU LENGTH field is less than the minimum length in bytes specified in table 31 (see 5.2.2) minus 4; or e) the IU LENGTH field is set to a value that causes a field (e.g., the ADDITIONAL CDB field in the EXTENDED COMMAND IU) or descriptor (e.g., a data descriptor) in the IU to be truncated.	The SOP target port shall: a) stop consuming from the inbound queue; and b) if configured, return an event with the EVENT TYPE field set to INBOUND QUEUE ERROR and the EVENT IDENTIFIER field set to INVALID IU LENGTH	The SOP initiator port should: a) stop consuming from the outbound queue; and b) delete the outbound queue

5.2.3 NULL IU

The NULL IU is an IU that shall be ignored by the recipient.

Table 34 defines the NULL IU.

Table 34 — NULL IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (00h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (0000h)							(LSB)
3	(MSB)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 34.

5.2.4 General management request IUs

5.2.4.1 REPORT GENERAL IU

5.2.4.1.1 REPORT GENERAL IU overview

The REPORT GENERAL IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that general information be reported.

Table 35 defines the REPORT GENERAL IU.

Table 35 — REPORT GENERAL IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (01h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 9	REQUEST IDENTIFIER							
10 11	Reserved							
12	ALLOCATION LENGTH							(LSB)
15	(MSB)							
16 ••• 31	Reserved							
32 ••• n	Data-In Buffer descriptor area (if any)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 35.

The QUEUING INTERFACE SPECIFIC field contains queuing interface specific parameters and is defined in A.4.2.1 and Annex B.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The ALLOCATION LENGTH field is defined in 5.2.6.1.

The Data-In Buffer descriptor area, if any, specifies the Data-In Buffer and is defined in A.4.3.

NOTE 13 - The length of the Data-In Buffer descriptor area is the IU LENGTH field plus 4 minus the offset of the Data-In Buffer descriptor area.

5.2.4.1.2 REPORT GENERAL parameter data

The format of the parameter data returned in the Data-In Buffer is shown in table 36.

Table 36 — REPORT GENERAL parameter Data-In Buffer (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0 ••• 3	Reserved							
~~Initiator mode~~SOP device capabilities								
4	Reserved						LOGICAL UNITS PRESENT	SOP TARGET ~~DEVICE~~AND BRIDGE PRESENT
5 ••• 7	Reserved							
~~Target mode capabilities~~								
8	Reserved						APPLICATION CLIENTS PRESENT	~~TARGET INITIATOR BRIDGE PRESENT~~ Reserved
9 ••• 11	Reserved							
PCI Express capabilities								
12 ••• 15	Reserved							

Table 36 — REPORT GENERAL parameter Data-In Buffer (part 2 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
Incoming IU capabilities								
16	Reserved							INCOMING SPANNING SUPPORTED
17	Reserved							
18	MAXIMUM INCOMING IU SIZE							(LSB)
19	(MSB)							
20	MAXIMUM INCOMING EMBEDDED DATA BUFFERS							(LSB)
21	(MSB)							
22	MAXIMUM DATA BUFFERS							(LSB)
23	(MSB)							
24	Reserved							
•••								
31								
32	INCOMING IU TYPE SUPPORT BITMASK							
•••								
63								
Vendor specific capabilities								
64	Vendor specific							
•••								
71								
Queuing layer capabilities								
72	Reserved							
73								
74	QUEUING LAYER SPECIFIC DATA LENGTH (n - 75)							(LSB)
75	(MSB)							
76	Queuing layer specific data							
•••								
n								

A SOP TARGET ~~DEVICE~~AND BRIDGE PRESENT bit set to one indicates that the SOP device ~~is~~contains a~~n~~ SOP target device ~~that contains~~and a bridge (e.g., the SOP port in the device~~it~~ accepts SCSI command request IUs with the NEXUS IDENTIFIER field set to non-zero values and supports bridge management IUs). A SOP TARGET DEVICE

BRIDGE PRESENT bit set to zero indicates that the SOP device ~~is~~does not ~~a~~contain a SOP target device ~~that contains~~and a bridge.

A LOGICAL UNITS PRESENT bit set to one indicates that the SOP device contains a SOP target device that is a SCSI target device ~~that~~ contain~~s~~ing one or more logical units (e.g., ~~it~~the SOP target device accepts SCSI command request IUs with the NEXUS IDENTIFIER field set to 0000h). A LOGICAL UNITS PRESENT bit set to zero indicates that the SOP deviSOP target device is not a SCSI target device that contains one or more logical units.

~~A SOP INITIATOR DEVICE BRIDGE PRESENT bit set to one indicates that the SOP device is a SOP initiator device that contains a bridge. A SOP INITIATOR DEVICE BRIDGE PRESENT bit set to zero indicates that the SOP device is not a SOP initiator device that contains a bridge.~~

An APPLICATION CLIENTS PRESENT bit set to one indicates that the SOP device ~~is~~contains a SOP initiator device ~~and~~that is a SCSI initiator device ~~that~~ contain~~s~~ing one or more SCSI application clients. An APPLICATION CLIENTS PRESENT bit set to zero indicates that the SOP device ~~is~~does not contain a SOP initiator device that contains~~and~~ a SCSI initiator device ~~that~~ contain~~s~~ing one or more SCSI application clients.

An INCOMING SPANNING SUPPORTED bit set to one indicates that the SOP device supports spanning an incoming IU across multiple elements up to the maximum incoming IU size indicated by the MAXIMUM INCOMING IU SIZE field. An INCOMING SPANNING SUPPORTED bit set to zero indicates that the SOP device does not support spanning an incoming IU.

The MAXIMUM INCOMING EMBEDDED DATA BUFFERS field indicates the maximum number of data descriptors in the data descriptor list embedded in an incoming IU that the SOP device supports, not counting any implicit restrictions caused by the overall incoming IU size. A MAXIMUM INCOMING EMBEDDED DATA BUFFERS field set to 0000h indicates that there is no limit.

The MAXIMUM DATA BUFFERS field indicates the maximum number of data descriptors in a scatter gather list that the SOP device supports. A MAXIMUM DATA BUFFERS field set to 0000h indicates that there is no limit.

The MAXIMUM INCOMING IU SIZE field indicates the maximum size in bytes in an incoming IU that the SOP device supports, not counting any restrictions implied by the queuing layer (e.g., the queue size, or the element size if spanning is not supported). A MAXIMUM INCOMING IU SIZE field set to 0000h indicates that there is no limit.

The INCOMING IU TYPE SUPPORT BITMASK field indicates the IU types that the SOP device supports. A bit set to one indicates that the corresponding IU type is supported. A bit set to zero indicates that the corresponding IU type is not supported. The first bit (i.e., byte 32 bit 0) corresponds to an IU type of 00h (i.e, the NULL IU); the last bit (i.e., byte 63 bit 7) corresponds to an IU type of FFh.

The QUEUING LAYER SPECIFIC DATA LENGTH field indicates the number of bytes in the queuing layer specific data. The queuing layer specific data length is defined in Annex A for PQI and Annex B for NVMe.

The queuing layer specific data is defined in Annex A for PQI and Annex B for NVMe.

5.2.4.2 REPORT CONFIGURATION IU

5.2.4.2.1 REPORT CONFIGURATION IU overview

The REPORT CONFIGURATION IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that configuration parameters be reported. The REPORT CONFIGURATION IU is complementary to the SET CONFIGURATION IU (see 5.2.4.3).

Table 37 defines the REPORT CONFIGURATION IU.

Table 37 — REPORT CONFIGURATION IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU TYPE (02h)							
1	COMPATIBLE FEATURES (00h)							
2				IU LENGTH (n-3)				(LSB)
3	(MSB)							
4				QUEUING INTERFACE SPECIFIC				
•••								
7								
8				REQUEST IDENTIFIER				
9								
10				Reserved				
11								
12				ALLOCATION LENGTH				(LSB)
•••								
15	(MSB)							
16	Reserved				PAGE CONTROL			
17	PAGE CODE							
18	SUBPAGE CODE							
19	Reserved							
20				INSTANCE				
•••								
23								
24				Reserved				
•••								
31								
32				Data-In Buffer descriptor area (if any)				
•••								
n								

The IU TYPE field, COMPATIBLE FEATURES field and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 37.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The ALLOCATION LENGTH field is defined in 5.2.6.1.1.

The PAGE CONTROL field specifies the type of configuration parameters to be returned in the configuration pages and is defined in table 38.

Table 38 — PAGE CONTROL field

Code	Parameter Type	Reference
0h	Current values	**5.2.4.2.2**
1h	Changeable values	**5.2.4.2.3**
2h	Default values	**5.2.4.2.4**
3h	Saved values	**5.2.4.2.5**
4h to Fh	Reserved	

The PAGE CONTROL field only affects the configuration parameters within the configuration pages, however the PS bit (see 6.3.4), PAGE CODE field and SUBPAGE CODE field should return current values (i.e., as if page control is set to 00h).

The PAGE CODE field and SUBPAGE CODE field (see 6.3.4) specify which configuration pages to return.

For a given page code and subpage code, there may be multiple instances of the specific configuration page (i.e., one instance for each phy for the SSP/PHY configuration page). The INSTANCE field is used to address a specific instance of a page.

The Data-In Buffer descriptor area, if any, is defined in 5.2.4.1.1.

If a SOP management application client issues a REPORT CONFIGURATION IU with a page code value or subpage code value not implemented by the SOP management device server, then the SOP management device server shall return a MANAGEMENT RESPONSE IU with a RESULT CODE field set to INVALID FIELD IN REQUEST IU.

The configuration parameter list for all device types for REPORT CONFIGURATION IU and SET CONFIGURATION IU is defined in 6.3.

5.2.4.2.2 Current values

A page control value of 00h requests that the SOP management device server return the current values of the configuration parameters. The current values are:

a) the current values of the configuration parameters established by the last successful SET CONFIGURATION IU request;
b) the saved values of the configuration parameters if a SET CONFIGURATION IU request has not successfully completed since the configuration parameters were restored to their saved values (see 5.2.4.2.5);

   or

c) the default values of the configuration parameters if a SET CONFIGURATION IU request has not successfully completed since the configuration parameters were restored to their default values (see 5.2.4.2.4).

5.2.4.2.3 Changeable values

A PAGE CONTROL field value of 01h requests that the SOP management device server return a mask denoting those configuration parameters that are changeable. In the mask, the bits in the fields in the configuration parameters that are changeable all shall be set to one and the bits in the fields of the configuration parameters that are non-changeable all shall be set to zero.

An attempt to change a non-changeable configuration parameter using the SET CONFIGURATION IU request shall result in an error condition (see 5.2.4.3).

The SOP management application client should issue a REPORT CONFIGURATION IU request with the PAGE CONTROL field set to 01h to determine which configuration parameters within the configuration page is changeable, and the supported length of the configuration page prior to issuing any SET CONFIGURATION IU requests.

5.2.4.2.4 Default values

A PAGE CONTROL field value of 02h requests that the SOP management device server return the default values of the configuration parameters. Unsupported parameters shall be set to zero.

5.2.4.2.5 Saved values

A PAGE CONTROL field value of 03h requests that the SOP management device server return the saved values of the configuration parameters. Configuration parameters not supported by the SOP management device server shall be set to zero.

The method of saving parameters is vendor specific. The parameters are preserved in such a manner that they are retained when the device is powered down. All saveable configuration pages should be considered saved when a SET CONFIGURATION IU request is issued with the SP bit set to one has completed successfully.

5.2.4.2.6 Initial responses

After a queueing layer interface reset, the SOP management device server shall respond in the following manner:

a) if default values are requested, then report the default values;
b) if saved values are requested, then report valid restored configuration parameters, or restore the configuration parameters and report them; or
c) if current values are requested and the current values have been sent by the SOP management application client via a SET CONFIGURATION IU request, then the current values shall be returned. If the current values have not been sent, then the device server shall return:
    A) the saved values, if saved values are available; or
    B) the default values.

5.2.4.3 SET CONFIGURATION IU

The SET CONFIGURATION IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that configuration parameters be set. The SET CONFIGURATION IU is complementary to the REPORT CONFIGURATION IU (see 5.2.4.2).

Table 39 defines the SET CONFIGURATION IU.

Table 39 — SET CONFIGURATION IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (03h)							
1	COMPATIBLE FEATURES (00h)							
2								(LSB)
	IU LENGTH (n-3)							
3	(MSB)							
4								
•••	QUEUING INTERFACE SPECIFIC							
7								
8								
	REQUEST IDENTIFIER							
9								
10								
	Reserved							
11								
12								(LSB)
•••	ALLOCATION LENGTH							
15	(MSB)							
16	SP	Reserved						
17	PAGE CODE							
18	SUBPAGE CODE							
19	Reserved							
20								
•••	INSTANCE							
23								
24								
•••	Reserved							
31								
32								
•••	Data-Out Buffer descriptor area (if any)							
n								

The IU TYPE field, COMPATIBLE FEATURES field and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 39.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

An SP bit (save pages bit) set to zero specifies that the SOP management device server shall perform the specified SET CONFIGURATION IU request, and shall not save the configuration page. An SP bit set to one specifies that the SOP management device server shall perform the specified SET CONFIGURATION IU request to a nonvolatile vendor specific location. Configuration pages that are saved are specified by the page saveable SP bit (pages saveable bit) that is returned in the configuration page by the REPORT CONFIGURATION IU request (see 5.2.8). If the PS bit is set to one in the REPORT CONFIGURATION IU data, then the configuration page shall be saveable by issuing a SET CONFIGURATION IU request with the SP bit set to one.

The PAGE CODE field and the SUBPAGE CODE field are defined in 5.2.4.2.

For a given page code and subpage code, there may be multiple instances of the specific configuration page (i.e., one instance for each phy for the SSP/PHY configuration page). The INSTANCE field is used to address a specific instance of a page.

If a SOP management application client issues a SET CONFIGURATION IU with a page code or subpage code value not implemented by the SOP management device server, the SOP management device server shall return a MANAGEMENT RESPONSE IU with a RESULT CODE field set to INVALID FIELD IN REQUEST IU.

The configuration parameter list for all device types for REPORT CONFIGURATION IU and SET CONFIGURATION IU is defined in 6.3.

The SOP management device server shall terminate the SET CONFIGURATION IU request with a MANAGEMENT RESPONSE IU with a RESULT CODE field set to INVALID FIELD IN REQUEST IU and shall not change any mode parameters in response to any of the following conditions:

a) if the SOP management application client sets any field that is reported as not changeable by the SOP management device server to a value other than its current value;
b) if the SOP management application client sends a configuration page with a page length that is not equal to the page length returned by the REPORT CONFIGURATION IU request for that configuration page;
c) if the SOP management application client sends an unsupported value for a configuration parameter and rounding is not implemented for that configuration parameter; or
d) if the SOP management application client sets any reserved field in the configuration parameter list to a non-zero value and the SOP management device server checks reserved fields.

If the SOP management application client sends a value for a configuration parameter that is outside the range supported by the SOP management device server and rounding is implemented for that configuration parameter, the SOP management device server handles the condition by either:

a) rounding the parameter to an acceptable value as described in (see 4.10); or
b) terminating the command with a MANAGEMENT RESPONSE IU with a RESULT CODE field set to INVALID FIELD IN REQUEST IU.

The Data-Out Buffer descriptor area, if any, specifies the Data-Out Buffer and is defined in A.4.4.

NOTE 14 - The length of the Data-Out Buffer descriptor area is the IU LENGTH field plus 4 minus the offset of the Data-Out Buffer descriptor area.

5.2.4.4 REPORT EVENT CONFIGURATION IU

5.2.4.4.1 REPORT EVENT CONFIGURATION IU overview

The REPORT EVENT CONFIGURATION IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that event configuration information be reported. The REPORT EVENT CONFIGURATION IU is complementary to the SET CONFIGURATION IU. SOP management device servers shall implement the REPORT EVENT CONFIGURATION IU request.

Table 40 defines the REPORT EVENT CONFIGURATION IU.

Table 40 — REPORT EVENT CONFIGURATION IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (04h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 ••• 9	REQUEST IDENTIFIER							
10 ••• 11	Reserved							
12	DATA-IN BUFFER SIZE							(LSB)
•••								
15	(MSB)							
16 ••• n	Data-In Buffer descriptor area (if any)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and are set to the values shown in table 40.

The QUEUING INTERFACE SPECIFIC field and the Data-In Buffer descriptor area, if any, are defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field and the DATA-IN BUFFER SIZE field are defined in 5.2.4.2.1.

5.2.4.4.2 REPORT EVENT CONFIGURATION Data-In Buffer contents

Table 41 defines the contents returned to the Data-In Buffer.

Table 41 — REPORT EVENT CONFIGURATION parameter Data-In Buffer

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	NUMBER OF REPORT EVENT CONFIGURATION DESCRIPTORS							
3	Reserved							
Report event configuration descriptor list								
4	Report event configuration descriptor [first] (see table 42)							
•••								
7								
•••								
n-4	Report event configuration descriptor [last] (see table 42)							
•••								
n								

The NUMBER OF REPORT EVENT CONFIGURATION DESCRIPTORS field indicates the number of report event configuration descriptors contained in the report event configuration descriptor list.

The report event configuration descriptor list contains a report event configuration descriptor for every supported event type, sorted in ascending order by event type.

Table 42 defines the report event configuration descriptor.

Table 42 — Report event configuration descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	EVENT TYPE							
1	Reserved							
2	QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE							
3								

The EVENT TYPE field indicates the event type (see 4.99).

A QUEUING INTERFACE SPECIFIC EVEN OUTBOUND QUEUE field set to 0000h indicates that the event type is disabled. A QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field set to a non-zero value indicates the outbound queue for the event type and is defined in A.4.1.

5.2.4.5 SET EVENT CONFIGURATION IU

5.2.4.5.1 SET EVENT CONFIGURATION IU overview

The SET EVENT CONFIGURATION IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that event configuration parameters be set. The SET EVENT CONFIGURATION IU is complementary to the REPORT CONFIGURATION IU request. SOP management device servers shall implement the SET EVENT CONFIGURATION IU request.

Table 43 defines the SET EVENT CONFIGURATION IU.

Table 43 — SET EVENT CONFIGURATION IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (05h)							
1	COMPATIBLE FEATURES (00h)							
2								(LSB)
	IU LENGTH (n-3)							
3	(MSB)							
4								
•••	QUEUING INTERFACE SPECIFIC							
7								
8								
	REQUEST IDENTIFIER							
9								
10								
	GLOBAL QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE							
11								
12								(LSB)
•••	DATA-OUT BUFFER SIZE							
15	(MSB)							
16								
•••	Data-Out Buffer descriptor area (if any)							
n								

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and are set to the values shown in table 43.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field is defined in 5.2.6.1.1.

A GLOBAL QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field is defined in 5.2.4.5.2.

The DATA-OUT BUFFER SIZE field and the Data-Out Buffer descriptor area, if any, are defined in 5.2.4.3.

5.2.4.5.2 SET EVENT CONFIGURATION Data-Out Buffer contents

Table 44 defines the Data-Out Buffer contents.

Table 44 — SET EVENT CONFIGURATION parameter Data-Out Buffer

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	NUMBER OF REPORT EVENT CONFIGURATION DESCRIPTORS							
3	Reserved							
	Set event configuration descriptor list							
4	Set event configuration descriptor [first] (see table 45)							
•••								
7								
•••								
n-4	Set event configuration descriptor [last] (see table 45)							
•••								
n								

The NUMBER OF SET EVENT CONFIGURATION DESCRIPTORS field specifies the number of set event configuration descriptors contained in the set event configuration descriptor list if the parameter data is transferred for the SET EVENT CONFIGURATION IU.

The set event configuration descriptor list contains a set event configuration descriptor for every event type being changed. The set event configuration descriptor list may be sorted in any order, may contain multiple entries for one event type, and shall be processed in order.

If the Data-Out Buffer is not large enough to hold the specified number of set event configuration descriptors, then the SOP management device server shall return a MANAGEMENT RESPONSE IU with the RESULT CODE field set to INVALID FIELD IN DATA-OUT BUFFER and the result-specific data pointing to the NUMBER OF SET EVENT CONFIGURATION DESCRIPTORS field and shall make no changes to the event configuration parameters.

Table 45 defines the set event configuration descriptor.

Table 45 — Set event configuration descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	EVENT TYPE							
1	Reserved							
2	QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE							
3								

The EVENT TYPE field specifies the event type (see 4.9). If the EVENT TYPE field specifies an unsupported event type, then the SOP management device server shall return a MANAGEMENT RESPONSE IU with the RESULT CODE field set to INVALID FIELD IN DATA-OUT BUFFER and shall make no changes to the event configuration parameters..

Table 46 — defines the GLOBAL QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field and the QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field.

Table 46 — Global queuing interface and queuing interface

<table>
<tr><th>Global queuing interface [a]</th><th>Queuing interface [b]</th><th>Description</th></tr>
<tr><td rowspan="2">0000h</td><td>0000h</td><td>The management device server shall disable the specified event type</td></tr>
<tr><td>Non-zero</td><td>The management device server shall enable the specified event type using the outbound queue specified in the QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field as defined in A.4.1</td></tr>
<tr><td rowspan="2">Non-zero</td><td>0000h</td><td>The management device server shall:
a) disable the specified event type; and
b) set the outbound queue for all enabled event types to the outbound queue specified in the GLOBAL QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field as defined in A.4.1</td></tr>
<tr><td>Non-zero</td><td>The management device server shall:
a) enable the specified event type; and
b) set the outbound queue for all enabled event types to the outbound queue specified in the GLOBAL QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field [c] as defined in A.4.1.
The QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field is ignored.</td></tr>
<tr><td colspan="3">[a] The GLOBAL QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field is in the SET EVENT CONFIGURATION IU.
[b] The QUEUING INTERFACE SPECIFIC EVENT OUTBOUND QUEUE field is in the set event configuration descriptor.
[c] Unless the event type is otherwise defined to use a particular outbound queue (e.g., to use the outbound queue associated with the request),</td></tr>
</table>

If the GLOBAL QUEUING INTERFACE SPECIFIC EVENT QUEUE field or the QUEUING INTERFACE SPECIFIC EVENT QUEUE field specifies an invalid event queueoutbound queue, then the SOP management device server shall return a MANAGEMENT RESPONSE IU with the RESULT CODE field set to INVALID FIELD IN DATA-OUT BUFFER and shall make no changes to the event configuration parameters.

5.2.5 SCSI command request IUs

5.2.5.1 LIMITED COMMAND IU

The LIMITED COMMAND IU is sent by a SOP initiator port to a SOP target port to request that a command be processed.

A command sent with the LIMITED COMMAND IU has the following attributes:

a) a nexus identifier of 0000h (i.e., the logical unit is contained in the SOP device);
b) a LUN of 00000000 00000000h;
c) a command priority of 0h; and
d) a task attribute of SIMPLE.

The LIMITED COMMAND IU only supports commands that:

a) have a CDB length less than or equal to 16 bytes; and
b) transfer no data or transfer data in only one direction (i.e., not bidirectional commands).

NOTE 15 - The COMMAND IU (see 5.2.5.2) and the EXTENDED COMMAND IU (see 5.2.5.3) may be used to transfer commands that are unable to be delivered with the LIMITED COMMAND IU.

Table 47 defines the LIMITED COMMAND IU.

Table 47 — LIMITED COMMAND IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (10h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 9	REQUEST IDENTIFIER							
10	Reserved					PARTIAL	DATA DIRECTION	
11	Reserved							
12	DATA BUFFER SIZE							(LSB)
•••								
15	(MSB)							
16 ••• 31	CDB							
32 ••• n	Data Buffer descriptor area (if any)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 47.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

A PARTIAL bit set to zero specifies that the Data Buffer descriptor area completely describes the Data Buffer (e.g., in PQI, the Data Buffer descriptor area is the last SGL segment). A PARTIAL bit set to one specifies that the Data

Buffer descriptor area may or may not completely describe the Data Buffer (e.g., in PQI, the Data Buffer descriptor area is not the last SGL segment).

If the PARTIAL bit is set to zero and the Data Buffer descriptor area does not completely describe the Data Buffer, then:

a) if the DATA DIRECTION field is set to 10b, then:
   A) if the device server is not a bridge device server, then the device server returns CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to DATA-IN BUFFER ERROR (see 6.5.1); and
   B) the SOP target port shall set the DATA-IN TRANSFER RESULT field to DATA-IN BUFFER ERROR in the COMMAND RESPONSE IU;

   or

b) if the DATA DIRECTION field is set to 01b, then:
   A) if the device server is not a bridge device server, then the device server returns CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to DATA-OUT BUFFER ERROR (see 6.5.1); and
   B) the SOP target port shall set the DATA-OUT TRANSFER RESULT field to DATA-OUT BUFFER ERROR in the COMMAND RESPONSE IU.

The DATA DIRECTION field specifies the direction of the data transfer associated with the CDB and is defined in table 48.

Table 48 — DATA DIRECTION field

Code	Description
00b	Non-data command; no data is transferred
01b	Unidirectional command; data is transferred from the Data-Out Buffer
10b	Unidirectional command; data is transferred to the Data-In Buffer
11b	Reserved [a]
[a] In the EXTENDED COMMAND IU, this value is used to specify a bidirectional command.	

If the DATA DIRECTION field is set to 11b, then the SOP target port shall return a COMMAND RESPONSE IU with response data with the RESPONSE CODE field set to INVALID FIELD IN INFORMATION UNIT.

If:

a) the DATA DIRECTION field is inconsistent with the direction of the data transfer associated with the CDB; and
b) the device server is not a bridge device server,

then the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN COMMAND INFORMATION UNIT.

The DATA BUFFER SIZE field specifies the maximum number of bytes to be transferred to or from the Data Buffer for this command.

The CDB field contains a 16 byte or smaller SCSI Command Descriptor Block (CDB) (see SPC-4). If the CDB contains a bidirectional command, then the device server returns CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN COMMAND INFORMATION UNIT.

The Data Buffer descriptor area, if any, specifies the Data Buffer and is defined in A.4.5.

NOTE 16 - The length of the Data Buffer descriptor area is the IU LENGTH field plus 4 minus the offset of the Data Buffer descriptor area.

5.2.5.2 COMMAND IU

The COMMAND IU is sent by a SOP initiator port to a SOP target port to request that a command be processed.

A command sent with the COMMAND IU has the following attributes:

a) the nexus identifier is specified;
b) the LUN is specified;
c) the command priority is specified; and
d) the task attribute is specified.

The COMMAND IU only supports commands that:

a) have a CDB length less than or equal to 32 bytes; and
b) transfer no data or transfer data in only one direction (i.e., not bidirectional commands).

NOTE 17 - The EXTENDED COMMAND IU (see 5.2.5.3) may be used to deliver commands that are unable to be delivered with the COMMAND IU.

Table 49 defines the COMMAND IU.

Table 49 — COMMAND IU (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (11h)							
1	COMPATIBLE FEATURES (00h)							
2								(LSB)
3	(MSB)			IU LENGTH (n-3)				
4	QUEUING INTERFACE SPECIFIC							
•••								
7								
8	REQUEST IDENTIFIER							
9								
10	NEXUS IDENTIFIER							
11								
12								(LSB)
•••	DATA BUFFER SIZE							
15	(MSB)							
16	LOGICAL UNIT NUMBER							
•••								
23								

Table 49 — COMMAND IU (part 2 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
24	PROTOCOL SPECIFIC							
25								
26	Reserved					PARTIAL	DATA DIRECTION	
27	Reserved							
•••								
29								
30	Reserved	COMMAND PRIORITY				TASK ATTRIBUTE		
31	Reserved			ADDITIONAL CDB BYTES USAGE			Reserved	
32	CDB							
•••								
47								
48	ADDITIONAL CDB BYTES							
•••								
63								
•••								
64	Data Buffer descriptor area (if any)							
•••								
n								

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 49.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The PARTIAL bit, the DATA DIRECTION field, the DATA BUFFER SIZE field, and the Data Buffer descriptor area are defined in 5.2.5.1.

The NEXUS IDENTIFIER field specifies the nexus identifier (see 4.3.2) of the I_T nexus to which the command is to be sent.

The LOGICAL UNIT NUMBER field specifies the logical unit number of the logical unit to which the task router shall route the command. The structure of the LOGICAL UNIT NUMBER field shall be as defined in SAM-5. If the addressed logical unit does not exit, then the task router shall return a COMMAND RESPONSE IU with response data with the RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER.

The PROTOCOL SPECIFIC field is defined in B.1.

The COMMAND PRIORITY field specifies the relative scheduling importance of a command with the TASK ATTRIBUTE field set to 000b (i.e., SIMPLE) in relation to other commands already in the task set with SIMPLE task attributes (see SAM-5).

Table 50 defines the TASK ATTRIBUTE field.

Table 50 — TASK ATTRIBUTE field

Code	Task attribute	Description
000b	SIMPLE	Specifies that the command be managed according to the rules for a SIMPLE task attribute (see SAM-5).
001b	HEAD OF QUEUE	Specifies that the command be managed according to the rules for a HEAD OF QUEUE task attribute (see SAM-5).
010b	ORDERED	Specifies that the command be managed according to the rules for an ORDERED task attribute (see SAM-5).
011b	Reserved	
100b	ACA	Specifies that the command be managed according to the rules for an ACA task attribute (see SAM-5).
101b to 111b	Reserved	

Table 51 defines the ADDITIONAL CDB BYTES USAGE field.

Table 51 — ADDITIONAL CDB BYTES USAGE field

Code	Description
000b	No bytes in the ADDITIONAL CDB BYTES field are valid (i.e., 16 byte CDB)
001b	No more than the first 4 bytes in the ADDITIONAL CDB BYTES field are valid (i.e., 20 byte CDB)
010b	No more than the first 8 bytes in the ADDITIONAL CDB BYTES field are valid (i.e., 24 byte CDB)
011b	No more than the first 12 bytes in the ADDITIONAL CDB BYTES field are valid (i.e., 28 byte CDB)
100b	No more than the first 16 bytes in the ADDITIONAL CDB BYTES field are valid (i.e., 32 byte CDB)
101b to 111b	Reserved. The SOP target port shall process this IU as having an error, transfer no data, and return a COMMAND RESPONSE IU with the RESPONSE CODE field set to INVALID FIELD IN INFORMATION UNIT.

The CDB field and ADDITIONAL CDB BYTES field together contain the CDB to be interpreted by the addressed logical unit. Any bytes after the end of the CDB within the two fields shall be ignored by the device server (e.g., a 24-byte CDB occupies 16 bytes of the CDB field and the first 8 bytes of the ADDITIONAL CDB field; the remaining 8 bytes of the ADDITIONAL CDB field are ignored). The contents of the CDB are defined in the SCSI command standards (e.g., SPC-4). If the CDB contains a bidirectional command, then the device server returns CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN COMMAND INFORMATION UNIT (see 6.5).

If the CDB is truncated (i.e., does not fit into the CDB field and the ADDITIONAL CDB BYTES field as constrained by the ADDITIONAL CDB BYTES USAGE field), then the device server shall transfer no data and return CHECK

CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN COMMAND INFORMATION UNIT.

5.2.5.3 EXTENDED COMMAND IU

The EXTENDED COMMAND IU is sent by a SOP initiator port to a SOP target port to request that a command be processed.

A command sent with the EXTENDED COMMAND IU has the following attributes:

a) the nexus identifier is specified;
b) the LUN is specified;
c) the command priority is specified; and
d) the task attribute is specified.

The COMMAND IU only supports commands that:

a) have a CDB length less than or equal to 32 bytes; and
b) transfer no data, transfer data in only one direction, or transfer data in both directions (i.e., bidirectional commands).

NOTE 18 - The EXTENDED COMMAND IU may be used to deliver commands that are unable to be delivered with the LIMITED COMMAND IU (see 5.2.5.1) or the COMMAND IU (see 5.2.5.2).

Table 52 defines the EXTENDED COMMAND IU.

Table 52 — EXTENDED COMMAND IU (part 1 of 3)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (12h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4	QUEUING INTERFACE SPECIFIC							
•••								
7								

Table 52 — EXTENDED COMMAND IU (part 2 of 3)

Byte \ Bit	7	6	5	4	3	2	1	0
8	REQUEST IDENTIFIER							
9								
10	NEXUS IDENTIFIER							
11								
12								(LSB)
•••	DATA-IN BUFFER SIZE							
15	(MSB)							
16	LOGICAL UNIT NUMBER							
•••								
23								
24	PROTOCOL SPECIFIC							
25								
26	Reserved				PARTIAL OUT	PARTIAL IN	DATA DIRECTION	
27	Reserved							
•••								
29								
30	Reserved	COMMAND PRIORITY				TASK ATTRIBUTES		
31	ADDITIONAL CDB LENGTH (m)						Reserved	
32								(LSB)
•••	DATA-OUT BUFFER SIZE							
35	(MSB)							
36	DATA-IN BUFFER DESCRIPTOR AREA SIZE							(LSB)
37	(MSB)							
38	Reserved							
•••								
47								
48	CDB							
•••								
63								

Table 52 — EXTENDED COMMAND IU (part 3 of 3)

Byte \ Bit	7	6	5	4	3	2	1	0
64	ADDITIONAL CDB (if any)							
•••								
63+(m/4)								
64+(m/4)	Data-In Buffer descriptor area (if any)							
	Data-Out Buffer descriptor area (if any)							
n								

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 52.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The DATA DIRECTION field, and the DATA TRANSFER LENGTH field are defined in 5.2.5.1.

The NEXUS IDENTIFIER field, the PROTOCOL SPECIFIC field, the LOGICAL UNIT NUMBER field, the COMMAND PRIORITY field, the TASK ATTRIBUTE field are defined in 5.2.5.2.

The DATA-IN BUFFER SIZE field specifies the maximum number of bytes to be transferred to the Data-In Buffer for this command.

A PARTIAL OUT bit set to zero specifies that the Data-Out Buffer descriptor area completely describes the Data-Out Buffer (e.g., in PQI, the Data-Out Buffer descriptor area is the last SGL segment). A PARTIAL OUT bit set to one specifies that the Data-Out Buffer descriptor area may or may not completely describe the Data-Out Buffer (e.g., in PQI, the Data-Out Buffer descriptor area is not the last SGL segment). If the PARTIAL OUT bit is set to zero and the Data-Out Buffer descriptor area does not completely describe the Data-Out Buffer, then:

a) if the device server is not a bridge device server, then the device server returns CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to DATA-OUT BUFFER ERROR (see 6.5.2); and
b) the SOP target port shall set the DATA-OUT TRANSFER RESULT field to DATA-OUT BUFFER ERROR in the COMMAND RESPONSE IU.

A PARTIAL IN bit set to zero specifies that the Data-In Buffer descriptor area completely describes the Data-In Buffer (e.g., in PQI, the Data-In Buffer descriptor area is the last SGL segment). A PARTIAL IN bit set to one specifies that the Data-In Buffer descriptor area may or may not completely describe the Data-In Buffer (e.g., in PQI, the Data-In Buffer descriptor area is not the last SGL segment). If the PARTIAL IN bit is set to zero and the Data-In Buffer descriptor area does not completely describe the Data-In Buffer, then:

a) if the device server is not a bridge device server, then the device server returns CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to DATA-IN BUFFER ERROR (see 6.7.1); and
b) the SOP target port shall set the DATA-IN TRANSFER RESULT field to DATA-IN BUFFER ERROR in the COMMAND RESPONSE IU.

The DATA DIRECTION field specifies the direction of the data transfer associated with the CDB and is defined in table 53.

Table 53 — DATA DIRECTION field

Code	Description
00b	Non-data command; no data is transferred
01b	Unidirectional command; data is transferred from the Data-Out Buffer
10b	Unidirectional command; data is transferred to the Data-In Buffer
11b	Bidirectional command; data is transferred from the Data-Out Buffer and data is transferred to the Data-In Buffer

If:

a) the DATA DIRECTION field is inconsistent with the direction of the data transfer associated with the CDB; and
b) the device server is not a bridge device server,

then the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN COMMAND INFORMATION UNIT.

The ADDITIONAL CDB LENGTH field specifies the length in multiples of 4 bytes of the ADDITIONAL CDB BYTES field (e.g., 00h means 0 bytes and supports a 16-byte CDB; 3Fh means 252 bytes and supports a 268 byte CDB).

The DATA-OUT BUFFER SIZE field specifies the maximum number of bytes to be transferred from the Data-Out Buffer for this command.

The CDB field and the ADDITIONAL CDB BYTES field together contain the CDB to be interpreted by the addressed logical unit. Any bytes after the end of the CDB within the two fields shall be ignored by the device server. The contents of the CDB are defined in the SCSI command standards (e.g., SPC-4).

If the CDB is truncated (i.e., does not fit into the CDB field and the ADDITIONAL CDB BYTES field), then the device server shall transfer no data and return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN COMMAND INFORMATION UNIT.

The DATA-IN BUFFER DESCRIPTOR AREA SIZE field specifies the size of the Data-In Buffer descriptor area and is defined in A.4.6.

The Data-In Buffer descriptor area, if any, specifies the Data-In Buffer and is defined in A.4.3.

The Data-Out Buffer descriptor area, if any, specifies the Data-Out Buffer and is defined in A.4.4.

NOTE 19 - The length of the Data-Out Buffer descriptor area is the IU LENGTH field plus 4 minus the offset of the Data-Out Buffer descriptor area.

5.2.5.4 TASK MANAGEMENT IU

The TASK MANAGEMENT IU is sent by a SOP initiator port to a SOP target port to request that a task management function be processed by the task manager addressed with the specified nexus identifier and LUN.

Table 54 defines the TASK MANAGEMENT IU.

Table 54 — TASK MANAGEMENT IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (13h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (001Ch)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 ••• 9	REQUEST IDENTIFIER							
10 ••• 11	NEXUS IDENTIFIER							
12 ••• 15	Reserved							
16 ••• 23	LOGICAL UNIT NUMBER							
24 ••• 25	PROTOCOL SPECIFIC							
26 ••• 27	Reserved							
28 ••• 29	REQUEST IDENTIFIER TO MANAGE							
30	TASK MANAGEMENT FUNCTION							
31	Reserved							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 54.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The PROTOCOL SPECIFIC field is defined in 5.2.5.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The NEXUS IDENTIFIER field specifies the nexus identifier (see 4.3.2) of the I_T nexus to which the task management function is to be sent.

The LOGICAL UNIT NUMBER field specifies the (8-byte) LUN to which the task management request is addressed. The structure of the LOGICAL UNIT NUMBER field shall be as defined in SAM-5. If the addressed logical unit does not exit, then the task router shall return a TASK MANAGEMENT RESPONSE IU with the RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER.

Table 55 defines the TASK MANAGEMENT FUNCTION field.

If the TASK MANAGEMENT FUNCTION field is set to 01h (i.e., ABORT TASK) or 80h (i.e., QUERY TASK), then the REQUEST IDENTIFIER TO MANAGE field specifies the request identifier from the LIMITED COMMAND IU, the COMMAND IU, or EXTENDED COMMAND IU that contained the command to be aborted or queried. If the TASK MANAGEMENT FUNCTION field is not set to 01h or 80h, then the REQUEST IDENTIFIER TO MANAGE field shall be ignored by the recipient.

Table 55 — TASK MANAGEMENT FUNCTION field (part 1 of 2)

Code	Task Management Function	Description
00h	Reserved	
01h	ABORT TASK	The task manager shall perform the ABORT TASK task management function with L set to the value of the LOGICAL UNIT NUMBER field and Q set to the value of the REQUEST IDENTIFIER TO MANAGE field (see SAM-5).
02h	ABORT TASK SET	The task manager shall perform the ABORT TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
03h	Reserved	
04h	CLEAR TASK SET	The task manager shall perform the CLEAR TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
05h to 07h	Reserved	
08h	LOGICAL UNIT RESET	The logical unit shall perform the LOGICAL UNIT RESET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
09h to 0Fh	Reserved	

[a] If the SOP device processes an I_T NEXUS RESET, then the SOP device shall ignore the LOGICAL UNIT NUMBER field.
[b] If ACA is supported by the logical unit (see SAM-5), then CLEAR ACA shall be supported.

Table 55 — TASK MANAGEMENT FUNCTION field (part 2 of 2)

Code	Task Management Function	Description
10h	I_T NEXUS RESET [a]	The SCSI target device shall perform the I_T NEXUS RESET task management function (see SAM-5).
11h to 3Fh	Reserved	
40h	CLEAR ACA [b]	The task manager shall perform the CLEAR ACA task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
41h to 7Fh	Reserved	
80h	QUERY TASK	The task manager shall perform the QUERY TASK task management function with L set to the value of the LOGICAL UNIT NUMBER field and Q set to the value of the REQUEST IDENTIFIER TO MANAGE field (see SAM-5).
81h	QUERY TASK SET	The task manager shall perform the QUERY TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
82h	QUERY ASYNCHRONOUS EVENT	The task manager shall perform the QUERY ASYNCHRONOUS EVENT task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
83h to FFh	Reserved	

[a] If the SOP device processes an I_T NEXUS RESET, then the SOP device shall ignore the LOGICAL UNIT NUMBER field.
[b] If ACA is supported by the logical unit (see SAM-5), then CLEAR ACA shall be supported.

5.2.6 Bridge management request IUs

5.2.6.1 REPORT BRIDGE LOCAL PORTS IU

5.2.6.1.1 REPORT BRIDGE LOCAL PORTS IU overview

The REPORT BRIDGE LOCAL PORTS IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that information about the local port(s) supported by the bridge be reported.

Table 56 defines the REPORT BRIDGE LOCAL PORTS IU.

Table 56 — REPORT BRIDGE LOCAL PORTS IU (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (20h)							
1	COMPATIBLE FEATURES (00h)							

Table 56 — REPORT BRIDGE LOCAL PORTS IU (part 2 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 ... 9	REQUEST IDENTIFIER							
10 ... 11	Reserved							
12	ALLOCATION LENGTH							(LSB)
•••								
15	(MSB)							
16 ••• 31	Reserved							
32 ••• n	Data-In Buffer descriptor area (if any)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 56.

The QUEUING INTERFACE SPECIFIC field and the Data-In Buffer descriptor area, if any, are defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The ALLOCATION LENGTH field specifies the length in bytes of the Data-In Buffer. The contents of the information written to the Data-In Buffer shall not be altered to reflect the truncation, if any, that results from an insufficient allocation length.

5.2.6.1.2 REPORT BRIDGE LOCAL PORTS parameter data

The format of the parameter data returned in the Data-In Buffer is shown in table 57.

Table 57 — REPORT BRIDGE LOCAL PORTS parameter Data-In Buffer

Byte \ Bit	7	6	5	4	3	2	1	0
0	NEXUS GENERATION COUNT							(LSB)
•••								
3	(MSB)							
4	Reserved							
5								
6	NUMBER OF LOCAL PORT DESCRIPTORS (m)							(LSB)
7	(MSB)							
	Local port descriptor list							
8	Local port descriptor [first] (see table 58)							
•••								
11								
•••								
(m×4)+4	Local port descriptor [Last] (see table 58)							
•••								
(m×4)+7								

The NEXUS GENERATION COUNT field indicates the nexus generation count (see 4.3.3).

The NUMBER OF LOCAL PORT DESCRIPTORS field indicates the number of local port descriptors in the local port descriptor list.

The local port descriptor list includes a local port descriptor for every local port currently supported by the bridge. This list may be sorted in any order.

Table 58 defines the local port descriptor.

Table 58 — Local port descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	RELATIVE PORT IDENTIFIER							(LSB)
1	(MSB)							
2					PROTOCOL IDENTIFIER			
3	Reserved						INITIATOR PORT	TARGET PORT

The RELATIVE PORT IDENTIFIER field indicates the relative port identifier (see SAM-5) of the local port being described by this descriptor.

NOTE 20 - Relative port identifiers are from 1 to 65 535 (see SAM-5).

The PROTOCOL IDENTIFIER field indicates the SCSI transport protocol used by the local port and the remote port, and is defined in SPC-4.

An INITIATOR PORT bit set to one indicates that the local port is capable of being a SCSI initiator port (i.e., the SOP device accepts SCSI commands from the SOP target port to be sent by the local port). An INITIATOR PORT bit set to zero indicates that the local port is not capable of being a SCSI initiator port.

A TARGET PORT bit set to one indicates that the local port is capable of being a SCSI target port (i.e., the SOP device sends SCSI commands received by the local port over the SOP initiator port). A TARGET PORT bit set to zero indicates that the local port is not capable of being a SCSI target port.

NOTE 21 - The local port may be both a SCSI initiator port and a SCSI target port.

5.2.6.2 REPORT BRIDGE LOCAL PORT DETAILS IU

5.2.6.2.1 REPORT BRIDGE LOCAL PORT DETAILS IU overview

The REPORT BRIDGE LOCAL PORT DETAILS IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that detailed information about a local port be reported.

Table 59 defines the REPORT BRIDGE LOCAL PORT DETAILS IU.

Table 59 — REPORT BRIDGE LOCAL PORT DETAILS IU (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (21h)							
1	COMPATIBLE FEATURES (00h)							
2				IU LENGTH (n-3)				(LSB)
3	(MSB)							
4				QUEUING INTERFACE SPECIFIC				
•••								
7								
8				REQUEST IDENTIFIER				
9								
10				Reserved				
11								
12				ALLOCATION LENGTH				(LSB)
•••								
15	(MSB)							
16				RELATIVE PORT IDENTIFIER				(LSB)
17	(MSB)							

Table 59 — REPORT BRIDGE LOCAL PORT DETAILS IU (part 2 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
18 ••• 31	Reserved							
32 ••• n	Data-In Buffer descriptor area (if any)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 59.

The QUEUING INTERFACE SPECIFIC field and the Data-In Buffer descriptor area, if any, are defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The ALLOCATION LENGTH field is defined in 5.2.6.1.1.

The RELATIVE PORT IDENTIFIER field specifies the relative port identifier of the local port. If the specified local port does not exist, then the management device server shall return a MANAGEMENT RESPONSE IU with the RESULT field set to INVALID FIELD IN REQUEST IU.

5.2.6.2.2 REPORT BRIDGE LOCAL PORT DETAILS parameter data

The format of the parameter data returned in the Data-In Buffer is shown in table 60.

Table 60 — REPORT BRIDGE LOCAL PORT DETAILS parameter Data-In Buffer

Byte \ Bit	7	6	5	4	3	2	1	0
0								(LSB)
•••	NEXUS GENERATION COUNT							
3	(MSB)							
4								(LSB)
5	(MSB)	RELATIVE PORT IDENTIFIER						
6	Reserved				PROTOCOL IDENTIFIER			
7	Reserved						INITIATOR PORT	TARGET PORT
8	Reserved							
9								
10								(LSB)
11	(MSB)	LOCAL PORT BRIDGE PORT IDENTIFIER LENGTH (k)						
12	Local bridge port identifier							
•••								
k+11								
k+12								(LSB)
k+13	(MSB)	PROTOCOL-SPECIFIC PORT INFORMATION LENGTH (m=0000h)						
k+14	Protocol-specific port information							
•••								
k+m+14								
k+m+15								(LSB)
k+m+16	(MSB)	VENDOR SPECIFIC PORT INFORMATION LENGTH (n)						
k+m+17	Vendor specific							
•••								
k+m+n+15								

The NEXUS GENERATION COUNT field is defined in 5.2.6.1.1.

The RELATIVE PORT IDENTIFIER field, the PROTOCOL IDENTIFIER field, the INITIATOR PORT bit and the TARGET PORT bit are defined in 5.2.6.1.2.

The LOCAL PORT BRIDGE PORT IDENTIFIER LENGTH field indicates the length in bytes of the local port bridge port identifier.

The local port bridge port identifier indicates the bridge port identifier (see 5.3.1) for the local port.

NOTE 22 - The local port may be both a SCSI initiator port and a SCSI target port.

The PROTOCOL-SPECIFIC PORT INFORMATION LENGTH field indicates the length in bytes of the protocol-specific port information and shall be set to the value shown in table 60.

The protocol-specific port information is not defined in this standard.

The VENDOR SPECIFIC PORT INFORMATION LENGTH field indicates the length in bytes of the vendor specific port information.

The vendor specific port information bytes are vendor specific.

5.2.6.3 REPORT BRIDGE NEXUSES IU

5.2.6.3.1 REPORT BRIDGE NEXUSES IU overview

The REPORT BRIDGE NEXUSES IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that a list of the nexus identifiers assigned by the bridge be reported.

Table 61 defines the REPORT BRIDGE NEXUSES IU.

Table 61 — REPORT BRIDGE NEXUSES IU (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (22h)							
1	COMPATIBLE FEATURES (00h)							
2								(LSB)
3	(MSB)	IU LENGTH (n-3)						
4	QUEUING INTERFACE SPECIFIC							
•••								
7								
8	REQUEST IDENTIFIER							
9								
10	Reserved							
11								
12								(LSB)
•••	ALLOCATION LENGTH							
15	(MSB)							

Table 61 — REPORT BRIDGE NEXUSES IU (part 2 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
16 ••• 31	Reserved							
32 ••• n	Data-In Buffer descriptor area (if any)							

The REQUEST IDENTIFIER, the IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 61.

The QUEUING INTERFACE SPECIFIC field and the Data-In Buffer descriptor area, if any, are defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The ALLOCATION LENGTH field is defined in 5.2.6.1.1.

5.2.6.3.2 REPORT BRIDGE NEXUSES parameter data

The format of the parameter data returned in the Data-In Buffer is shown in table 62.

Table 62 — REPORT BRIDGE NEXUSES parameter Data-In Buffer

Bit / Byte	7	6	5	4	3	2	1	0
0								(LSB)
•••	NEXUS GENERATION COUNT							
3	(MSB)							
4 ••• 5	Reserved							
6								(LSB)
7	(MSB)	NUMBER OF LOCAL PORT DESCRIPTORS (m)						
Nexus descriptor list								
8 ••• 11	Nexus descriptor [first] (see table 63)							
•••								
(m×4)+4 ••• (m×4)+7	Nexus descriptor [Last] (see table 63)							

The NEXUS GENERATION COUNT field is defined in 5.2.6.1.2.

The NUMBER OF NEXUS DESCRIPTORS field indicates the number of nexus descriptors in the nexus descriptor list.

The nexus descriptor list includes a nexus descriptor (see table 63) for every nexus identifier currently assigned by the bridge. This list may be sorted in any order.

Table 63 — Nexus descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	NEXUS IDENTIFIER							
1								
2	TARGET MODE	Reserved			PROTOCOL IDENTIFIER			
3	NEXUS LOST	Reserved						

The NEXUS IDENTIFIER field is defined in 5.2.5.2.

NOTE 23 - Nexus identifiers are from 1 to 65 535 (see 4.3.2).

A TARGET MODE bit set to zero indicates that the local port is the SCSI initiator port in the I_T nexus. A TARGET MODE bit set to one indicates that the local port is the SCSI target port in the I_T nexus.

NOTE 24 - If a local port is used as both a SCSI initiator port and a SCSI target port, then the bridge assigns different nexus identifiers for each of those roles. See 4.3.2.

The PROTOCOL IDENTIFIER field indicates the SCSI transport protocol used by the local port and is defined in SPC-4.

A NEXUS LOST bit set to one indicates that an I_T nexus loss exists for the I_T nexus being described. A NEXUS LOST bit set to zero indicates that an I_T nexus loss does not exist for the I_T nexus being described.

5.2.6.4 REPORT BRIDGE NEXUS DETAILS IU

5.2.6.4.1 REPORT BRIDGE NEXUS DETAILS IU overview

The REPORT BRIDGE NEXUS DETAILS IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that detailed information about a nexus identifier be reported.

Table 64 defines the REPORT BRIDGE NEXUS DETAILS IU.

Table 64 — REPORT BRIDGE NEXUS DETAILS IU (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (23h)							
1	COMPATIBLE FEATURES (00h)							

Table 64 — REPORT BRIDGE NEXUS DETAILS IU (part 2 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
2								(LSB)
3	(MSB)			IU LENGTH (n-3)				
4 ••• 7				QUEUING INTERFACE SPECIFIC				
8 ••• 9				REQUEST IDENTIFIER				
10 ••• 11				NEXUS IDENTIFIER				
12								(LSB)
•••				ALLOCATION LENGTH				
15	(MSB)							
16								(LSB)
17	(MSB)			NEXUS IDENTIFIER				
18 ••• 31				Reserved				
32 ••• n				Data-In Buffer descriptor area (if any)				

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 64.

The QUEUING INTERFACE SPECIFIC field and the Data-In Buffer descriptor area, if any, are defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The ALLOCATION LENGTH field is defined in 5.2.6.1.1.

The NEXUS IDENTIFIER field specifies the nexus identifier (see 4.3.2) of the I_T nexus to be described. If the specified nexus is not currently assigned, then the management device server shall return a MANAGEMENT RESPONSE IU with the RESULT field set to INVALID FIELD IN REQUEST IU.

5.2.6.4.2 REPORT BRIDGE NEXUS DETAILS parameter data

The format of the parameter data returned in the Data-In Buffer is shown in table 65.

Table 65 — REPORT BRIDGE NEXUS DETAILS parameter Data-In Buffer

Byte \ Bit	7	6	5	4	3	2	1	0
0	NEXUS GENERATION COUNT							(LSB)
•••								
3	(MSB)							
4	NEXUS IDENTIFIER							
5								
6	TARGET MODE	Reserved			PROTOCOL IDENTIFIER			
7	NEXUS LOST	Reserved						
8	LOCAL PORT RELATIVE PORT IDENTIFIER							(LSB)
9	(MSB)							
10	REMOTE PORT BRIDGE PORT IDENTIFIER LENGTH (n-11)							(LSB)
11	(MSB)							
12	REMOTE PORT BRIDGE PORT IDENTIFIER							
•••								
n								

The NEXUS GENERATION COUNT field is defined 5.2.6.1.2.

The NEXUS IDENTIFIER field indicates the nexus identifier (see 4.3.2) of the I_T nexus being described.

The TARGET MODE bit, the PROTOCOL IDENTIFIER field, and the NEXUS LOST bit are as defined in 5.2.6.3.2.

The LOCAL PORT RELATIVE PORT IDENTIFIER field indicates the relative port identifier of the local port.

NOTE 25 - If the local port is a SCSI initiator port, then this is the I portion of the I_T nexus; if the local port is a SCSI target port, then this is the T portion of the I_T nexus.

NOTE 26 - The remote port may have a different native SCSI transport protocol or SCSI port identifier than the one indicated in this field, if it is behind another SCSI protocol bridge. This field identifies the SCSI port identifier using the protocol known to the local port.

The REMOTE PORT BRIDGE PORT IDENTIFIER LENGTH field indicates the length in bytes of the remote port bridge port identifier.

The remote port bridge port identifier indicate the bridge port identifier (see B.3.1) for the remote port.

5.2.6.5 REMOVE BRIDGE NEXUS IU

5.2.6.5.1 REMOVE BRIDGE NEXUS IU overview

The REMOVE BRIDGE NEXUS IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to request that the bridge remove a nexus identifier for a lost I_T nexus.

Table 66 defines the REMOVE BRIDGE NEXUS IU.

Table 66 — REMOVE BRIDGE NEXUS IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (24h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (001Ch)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8	REQUEST IDENTIFIER							
9								
10 ••• 15	Reserved							
16	NEXUS IDENTIFIER							(LSB)
17	(MSB)							
18 ••• 31	Reserved							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 66.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.4.1.1.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

The NEXUS IDENTIFIER field specifies the nexus identifier (see 4.3.2) to be removed. If the specified nexus identifier is not assigned, then the management device server shall return a MANAGEMENT RESPONSE IU with the RESULT field set to INVALID FIELD IN REQUEST IU. If the specified nexus identifier is currently assigned but the I_T nexus is not lost, then the SOP device shall return a GENERAL RESPONSE IU with a result of NEXUS IDENTIFIER NOT LOST.

5.2.7 General response IUs

5.2.7.1 MANAGEMENT RESPONSE IU

5.2.7.1.1 MANAGEMENT RESPONSE IU overview

The MANAGEMENT RESPONSE IU is sent by a SOP port in a SOP device containing a SOP management device server to a SOP port in a SOP device containing a SOP management application client to deliver response information for a management request (i.e., a general management request (see 5.2.4) or a bridge management request (see 5.2.6)).

Table 67 defines the MANAGEMENT RESPONSE IU.

Table 67 — MANAGEMENT RESPONSE IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (81h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4	QUEUING INTERFACE SPECIFIC							
•••								
7								
8	REQUEST IDENTIFIER							
9								
10	RESULT							
11	Reserved							
•••								
15								
16	Result specific data							
•••								
n								

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 67.

The QUEUING INTERFACE SPECIFIC field contains queuing interface specific parameters and is defined in A.4.2.2 and Annex B.

The REQUEST IDENTIFIER field specifies the request identifier (see 4.6).

Table 68 defines the RESULT field.

Table 68 — RESULT field

Code	Name	Request IUs	Description	Result Specific data
Non-error codes (00h to 3Fh)				
00h	GOOD	All [a]	The SOP management device server completed the function with no error.	None
01h to 3Eh	Reserved			
3Fh	Vendor specific non-error			
Error codes (40h to FFh)				
Request IU-agnostic error codes (40h to 7Fh)				
40h	UNKNOWN ERROR	All	The SOP management function has failed for an unknown reason.	None
41h	INVALID FIELD IN REQUEST IU	All	A field in the request IU was invalid. The result-specific data indicates the field in the IU that was invalid.	5.2.7.1.2
42h	INVALID FIELD IN DATA-OUT BUFFER	SET EVENT CONFIGURATION	A field in the Data-Out Buffer was invalid. The result-specific data indicates the field in the Data-Out Buffer that was invalid	
43h to 7Eh	Reserved			
7Fh	Vendor specific error			
Request IU-specific error codes (80h to FFh)				
80h to FEh	Reserved			
FFh	Vendor specific error			

[a] All IUs except IUs that have specific response IUs defined.

5.2.7.1.2 Result-specific data for INVALID FIELD IN REQUEST IU

If the RESULT field is set to INVALID FIELD IN REQUEST IU, then table 69 defines the result-specific data.

Table 69 — Result-specific data for RESULT field set to INVALID FIELD IN REQUEST IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	BYTE POINTER							(LSB)
1	(MSB)							
2	Reserved							
3	BPV	BIT POINTER			Reserved			

The BYTE POINTER field indicates the offset in the request IU of the first byte (i.e., the lowest byte number) containing the field with the invalid value.

A BPV (bit pointer valid) bit set to one indicates that the BIT POINTER field is valid. A BPV bit set to zero indicates that the BIT POINTER field is not valid.

If the BPV bit is set to one, then the BIT POINTER field indicates the offset in the byte of the request IU pointed to by the BYTE POINTER field of the first bit (i.e., the lowest bit number) containing the field with the invalid value.

5.2.7.1.3 Result-specific data for INVALID FIELD IN DATA-OUT BUFFER

If the RESULT field is set to INVALID FIELD IN DATA-OUT BUFFER, then table 70 defines the result-specific data.

Table 70 — Result-specific data for RESULT field set to INVALID FIELD IN DATA-OUT BUFFER

Byte \ Bit	7	6	5	4	3	2	1	0
0	BYTE POINTER							(LSB)
3	(MSB)							
4	BPV	BIT POINTER			Reserved			
5 ••• 7	Reserved							

The BYTE POINTER field indicates the offset in the Data-Out Buffer of the first byte (i.e., the lowest byte number) containing the field with the invalid value.

A BPV (bit pointer valid) bit set to one indicates that the BIT POINTER field is valid. A BPV bit set to zero indicates that the BIT POINTER field is not valid.

If the BPV bit is set to one, then the BIT POINTER field indicates the offset in the byte of the Data-Out Buffer pointed to by the BYTE POINTER field of the first bit (i.e., the lowest bit number) containing the field with the invalid value.

5.2.7.2 EVENT IU

The EVENT IU is sent by a SOP port in a SOP device containing a SOP management device server to a SOP port in a SOP device containing a SOP management application client to deliver information about a supported and enabled event type.

Table 71 defines the EVENT IU.

Table 71 — EVENT IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (82h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (n-3)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8	EVENT TYPE							
9	REQUEST ACKNOWLEDGE	Reserved						
10 11	EVENT IDENTIFIER							
12 ••• 15	ADDITIONAL EVENT IDENTIFIER							
16 ••• n	Event data (if any)							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and are set to the values shown in table 71.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.7.1.1.

The EVENT TYPE field, the EVENT IDENTIFIER field, the ADDITIONAL EVENT IDENTIFIER field and the Event data, if any, are defined in 4.9.

NOTE 27 - The length of the event data is the IU LENGTH field plus 4 minus the offset of the event data.

A REQUEST ACKNOWLEDGE bit set to one indicates that the SOP application client should send a corresponding EVENT ACKNOWLEDGE IU to the SOP device server with the values contained in the EVENT TYPE field and the EVENT IDENTIFIER field. A REQUEST ACKNOWLEDGE bit set to zero indicates that the SOP management application client shall not send a corresponding EVENT ACKNOWLEDGE IU.

5.2.7.3 EVENT ACKNOWLEDGE IU

The EVENT ACKNOWLEDGE IU is sent by a SOP port in a SOP device containing a SOP management application client to a SOP port in a SOP device containing a SOP management device server to acknowledge receipt of an event in an EVENT IU.

Table 72 defines the EVENT ACKNOWLEDGE IU.

Table 72 — EVENT ACKNOWLEDGE IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (83h)							
1	COMPATIBLE FEATURES (00h)							
2				IU LENGTH (n-3)				(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8	EVENT TYPE							
9	Reserved							
10	EVENT IDENTIFIER							
11								
12 ••• 15	ADDITIONAL EVENT IDENTIFIER							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and are set to the values shown in table 72.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.7.1.1.

The EVENT TYPE field, the EVENT IDENTIFIER field, and the ADDITIONAL EVENT IDENTIFIER FIELD are defined in 4.9.

5.2.8 SCSI command response IUs

5.2.8.1 SUCCESS IU

The SUCCESS IU is sent by a SOP target port to a SOP initiator port to deliver a response for the SCSI command with the indicated nexus identifier and request identifier.

The SUCCESS IU is used to complete a SCSI command that was delivered with the LIMITED COMMAND IU (see 5.2.5.1), the COMMAND IU (see 5.2.5.2), or the EXTENDED COMMAND IU (see 5.2.5.3) and has:

a) a service response of COMMAND COMPLETE;
b) GOOD status;
c) no status qualifier;

d) no sense data or response data;
e) no Data-In Buffer underflow; and
f) no Data-Out Buffer underflow.

NOTE 28 - The COMMAND RESPONSE IU (see 5.2.8.2) may be used to deliver responses that are unable to be delivered with the SUCCESS IU.

Table 73 defines the SUCCESS IU.

Table 73 — SUCCESS IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (90h)							
1	COMPATIBLE FEATURES (00h)							
2	IU LENGTH (000Ch)							(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 9	REQUEST IDENTIFIER							
10 11	NEXUS IDENTIFIER							
12 ••• 15	Reserved							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 73.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.7.1.1.

The NEXUS IDENTIFIER field indicates the nexus identifier (see 4.3.2) of the I_T nexus of the command for which this response is being returned.

The REQUEST IDENTIFIER field indicates the request identifier (see 4.6).

5.2.8.2 COMMAND RESPONSE IU

5.2.8.2.1 COMMAND RESPONSE IU overview

The COMMAND RESPONSE IU is sent by a SOP target port to a SOP initiator port to deliver a response for the SCSI command with the indicated nexus identifier and request identifier.

NOTE 29 - The COMMAND RESPONSE IU may be used to deliver SCSI command responses that are unable to be delivered with the SUCCESS IU (see 5.2.8.1).

Table 74 defines the COMMAND RESPONSE IU.

Table 74 — COMMAND RESPONSE IU (part 1 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (91h)							
1	COMPATIBLE FEATURES (00h)							
2				IU LENGTH (n-3)				(LSB)
3	(MSB)							
4	QUEUING INTERFACE SPECIFIC							
•••								
7								
8	REQUEST IDENTIFIER							
9								
10	NEXUS IDENTIFIER							
11								
12	DATA-IN TRANSFER RESULT							
13	DATA-OUT TRANSFER RESULT							
14	Reserved							
•••								
16								
17	STATUS							
18	STATUS QUALIFIER							
19								

[a] Any contents of the IU after the SENSE DATA field (i.e., after m+p+32 to n) shall be:
a) set to zero by the sender; and
b) ignored by the recipient.

Table 74 — COMMAND RESPONSE IU (part 2 of 2)

Byte \ Bit	7	6	5	4	3	2	1	0
20	SENSE DATA LENGTH (p)							(LSB)
21	(MSB)							
22	RESPONSE DATA LENGTH (m)							(LSB)
23	(MSB)							
24	DATA-IN TRANSFERRED							(LSB)
•••								
27	(MSB)							
28	DATA-OUT TRANSFERRED							(LSB)
•••								
31	(MSB)							
32	RESPONSE DATA (if any)							
•••								
m+31								
m+32	SENSE DATA (if any)							
•••								
m+p+32 [a]								

[a] Any contents of the IU after the SENSE DATA field (i.e., after m+p+32 to n) shall be:
a) set to zero by the sender; and
b) ignored by the recipient.

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 74.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.7.1.1.

The REQUEST IDENTIFIER field indicates the request identifier (see 4.6)

The NEXUS IDENTIFIER field is defined in 5.2.8.1.

Table 75 defines the DATA-IN TRANSFER RESULT field and the DATA-IN TRANSFERRED field.

Table 75 — DATA-IN TRANSFER RESULT field and the DATA-IN TRANSFERRED field (part 1 of 2)

DATA-IN TRANSFER RESULT field		DATA-IN TRANSFERRED field
Code	**Description**	
Results indicating a Service Response of COMMAND COMPLETE		
00h	DATA-IN BUFFER OK	Invalid
01h	DATA-IN BUFFER UNDERFLOW	Indicates the number of contiguous bytes starting with offset 0 in the Data-In Buffer that the device server transferred [a]
02h to 3Fh	Reserved	
Results indicating a Service Response of SERVICE DELIVERY OR TARGET FAILURE [b]		
Miscellaneous errors		
40h	DATA-IN BUFFER ERROR	Invalid
41h	DATA-IN BUFFER OVERFLOW - DATA BUFFER SIZE	Invalid
42h	DATA-IN BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA	Invalid
43h	DATA-IN BUFFER OVERFLOW - BRIDGE LOCAL PORT	Invalid
44h to 5Fh	Reserved	
PCI Express fabric errors		
60h	PCIE FABRIC ERROR	Invalid
61h	PCIE COMPLETION TIMEOUT	Invalid
62h	PCIE COMPLETER ABORT	Invalid
63h	PCIE POISONED TLP RECEIVED	Invalid
64h	PCIE ECRC CHECK FAILED	Invalid
65h	PCIE UNSUPPORTED REQUEST	Invalid
66h	PCIE ACS VIOLATION	Invalid
67h	PCIE TLP PREFIX BLOCKED	Invalid
68h to 6Fh	Reserved	

[a] The device server may or may not have transferred bytes above this offset.
[b] Even if the command completes with GOOD status, the Service Response is SERVICE DELIVERY OR TARGET FAILURE

Table 75 — DATA-IN TRANSFER RESULT field and the DATA-IN TRANSFERRED field (part 2 of 2)

DATA-IN TRANSFER RESULT field		DATA-IN TRANSFERRED field
Code	**Description**	
Other errors		
70h to EFh	Reserved	
Vendor specific errors		
F0h to FFh	Vendor specific	

[a] The device server may or may not have transferred bytes above this offset.
[b] Even if the command completes with GOOD status, the Service Response is SERVICE DELIVERY OR TARGET FAILURE

Table 76 defines the DATA-OUT TRANSFER RESULT field and the DATA-OUT TRANSFERRED field.

Table 76 — DATA-OUT TRANSFER RESULT field and the DATA-OUT TRANSFERRED field (part 1 of 2)

DATA-OUT TRANSFER RESULT field		DATA-OUT TRANSFERRED field
Code	**Description**	
Results indicating a Service Response of COMMAND COMPLETE		
00h	DATA-OUT BUFFER OK	Invalid
01h	DATA-OUT BUFFER UNDERFLOW	Indicates the number of contiguous bytes starting with offset 0 in the Data-In Buffer that the device server transferred [a]
02h to 3Fh	Reserved	
Results indicating a Service Response of SERVICE DELIVERY OR TARGET FAILURE [b]		
Miscellaneous errors		
40h	DATA-OUT BUFFER ERROR	Invalid
41h	DATA-OUT BUFFER OVERFLOW - DATA BUFFER SIZE	Invalid
42h	DATA-OUT BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA	Invalid
43h	DATA-OUT BUFFER OVERFLOW - BRIDGE LOCAL PORT	Invalid
44h to 5Fh	Reserved	

[a] The device server may or may not have transferred bytes above this offset.
[b] Even if the command completes with GOOD status, the Service Response is SERVICE DELIVERY OR TARGET FAILURE

Table 76 — DATA-OUT TRANSFER RESULT field and the DATA-OUT TRANSFERRED field (part 2 of 2)

DATA-OUT TRANSFER RESULT field		DATA-OUT TRANSFERRED field
Code	Description	
PCI Express fabric errors		
60h	PCIE FABRIC ERROR	Invalid
61h	PCIE COMPLETION TIMEOUT	Invalid
62h	PCIE COMPLETER ABORT	Invalid
63h	PCIE POISONED TLP RECEIVED	Invalid
64h	PCIE ECRC CHECK FAILED	Invalid
65h	PCIE UNSUPPORTED REQUEST	Invalid
66h	PCIE ACS VIOLATION	Invalid
67h	PCIE TLP PREFIX BLOCKED	Invalid
68h to 6Fh	Reserved	
Other errors		
70h to EFh	Reserved	
Vendor specific errors		
F0h to FFh	Vendor specific	

[a] The device server may or may not have transferred bytes above this offset.
[b] Even if the command completes with GOOD status, the Service Response is SERVICE DELIVERY OR TARGET FAILURE

The SENSE DATA LENGTH field indicates the length of the SENSE DATA field, if any.

The RESPONSE DATA LENGTH field indicates the length of the RESPONSE DATA field, if any.

The SENSE DATA LENGTH field and the RESPONSE DATA LENGTH field indicate if sense data or response data is present as described in table 77.

Table 77 — SENSE DATA LENGTH field and RESPONSE DATA LENGTH field combinations

SENSE DATA LENGTH field	RESPONSE DATA LENGTH field	Description	Reference
0000h	0000h	No data	5.2.8.2.2
Non-zero	0000h	Sense data	5.2.8.2.3
0000h	Non-zero	Response data	5.2.8.2.4
Non-zero	Non-zero	Reserved	

The STATUS field, the STATUS QUALIFIER field, the RESPONSE DATA field, if any, and the SENSE DATA field, if any, are defined in 5.2.8.2.2, 5.2.8.2.3, and 5.2.8.2.4.

5.2.8.2.2 No sense data or response data

If there is no sense data or response data, then the SOP target port shall:

a) set the IU LENGTH field to 0020h;
b) set the RESPONSE DATA LENGTH field to zero;
c) exclude the RESPONSE DATA field from the IU;
d) set the SENSE DATA LENGTH field to zero;
e) exclude the SENSE DATA field from the IU;
f) set the STATUS field to the status code (see SAM-5) for a completed command; and
g) set the STATUS QUALIFIER field to the status qualifier (see SAM-5) for the command.

5.2.8.2.3 Sense data present

If there is sense data, then the SOP target shall:

a) set the IU LENGTH field to 0020h plus the length of the sense data;
b) set the RESPONSE DATA field to zero;
c) exclude the RESPONSE DATA field from the IU;
d) set the SENSE DATA LENGTH field to a non-zero value indicating the number of bytes in the SENSE DATA field;
e) set the SENSE DATA field to the sense data (see SAM-5);
f) set the STATUS field to the status code (see SAM-5) for a completed command; and
g) set the STATUS QUALIFIER field to the status qualifier (see SAM-5) for the command.

5.2.8.2.4 Response data present

If there is response data, then the SOP target port shall:

a) set the IU LENGTH field to 0020h plus the length of the response data;
b) set the RESPONSE DATA LENGTH field to 0004h indicating the number of bytes in the RESPONSE DATA field;
c) set the RESPONSE DATA field as described in table 78;
d) set the SENSE DATA LENGTH field to 0000h;
e) exclude the SENSE DATA field from the IU;
f) set the STATUS field to the status code 00h; and
g) set the STATUS QUALIFIER field to 0000h.

Table 78 — RESPONSE DATA field

Byte \ Bit	7	6	5	4	3	2	1	0
0	ADDITIONAL RESPONSE INFORMATION							
•••								
2								
3	RESPONSE CODE							

The ADDITIONAL RESPONSE INFORMATION field indicates additional response information for certain task management functions (e.g., QUERY ASYNCHRONOUS EVENT) as defined in SAM-5 and for certain response codes. If the task management function does not define additional response information or the logical unit does not support additional response information, then the SCSI target port shall set the ADDITIONAL RESPONSE INFORMATION field to 000000h.

The RESPONSE CODE field contains the error condition or the completion status of a task management function and is defined in table 79.

Table 79 — RESPONSE CODE field

Code	Description	Additional response information
Task management function specific responses		
00h [a]	TASK MANAGEMENT FUNCTION COMPLETE	Valid
01h to 03h	Reserved	
04h [a]	TASK MANAGEMENT FUNCTION REJECTED	Reserved
05h [a]	TASK MANAGEMENT FUNCTION FAILED	Reserved
06h to 07h	Reserved	
08h [a]	TASK MANAGEMENT FUNCTION SUCCEEDED	Valid
General responses		
09h	INCORRECT LOGICAL UNIT NUMBER	Reserved
0Ah	OVERLAPPED REQUEST IDENTIFIER ATTEMPTED	Reserved
0Bh to 1Fh	Reserved	
20h	INVALID INFORMATION UNIT TYPE	Reserved
21h	INVALID INFORMATION UNIT LENGTH	Reserved
22h	INVALID LENGTH IN INFORMATION UNIT	Reserved
23h	MISALIGNED LENGTH IN INFORMATION UNIT	Reserved
24h	INVALID FIELD IN INFORMATION UNIT	Reserved
25h	INFORMATION UNIT TOO LONG	Reserved
26h to 7Fh	Reserved	
Bridge responses		
80h	BRIDGE LOCAL PORT DETECTED ERROR RESPONSE	Valid
81h to FFh	Reserved	

[a] This response code is only used in a TASK MANAGEMENT RESPONSE IU.

5.2.8.3 TASK MANAGEMENT RESPONSE IU

The TASK MANAGEMENT RESPONSE IU is sent by a SOP target port to a SOP initiator port to deliver a response for the task management function with the indicated nexus identifier and request identifier.

Table 80 defines the TASK MANAGEMENT RESPONSE IU.

Table 80 — TASK MANAGEMENT RESPONSE IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	IU TYPE (93h)							
1	COMPATIBLE FEATURES (00h)							
2				IU LENGTH (000Ch)				(LSB)
3	(MSB)							
4 ••• 7	QUEUING INTERFACE SPECIFIC							
8 9	REQUEST IDENTIFIER							
10 11	NEXUS IDENTIFIER							
12 ••• 15	RESPONSE DATA							

The IU TYPE field, the COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 5.2.2 and shall be set to the values shown in table 80.

The QUEUING INTERFACE SPECIFIC field is defined in 5.2.7.1.1.

The NEXUS IDENTIFIER field indicates the nexus identifier (see 4.3.2) of the I_T nexus of the task management function for which this response is being returned.

The REQUEST IDENTIFIER field indicates the request identifier (see 4.6).

The RESPONSE DATA field is described in 5.2.8.2.4.

5.3 Sequences of SOP communications for SCSI commands and task management functions

5.3.1 Overview

The sequence figures in 5.3 describe communications between a SOP initiator port and a SOP target port.

Lines in the figures with an arrow that points to the SOP target port represent a communication from the SOP initiator port to the SOP target port. The SCSI transport protocol service request is shown to the left of the line, and the SCSI transport protocol service indication is shown to the right of the arrow.

Lines in the figures with an arrow that points to the SOP initiator port represent a communication from the SOP target port to the SOP initiator port. The SCSI transport protocol service response is shown to the right of the line, and the SCSI transport protocol service confirmation is shown to the left of the arrow.

Where a sequence includes one or more data transfers, the SCSI transfer protocol services invoked for the transfer are shown. Where one or more data transfers may occur is shown by ovals encircling arrows. Where more than one data-in transfer and data-out transfer may occur is shown by a larger oval encircling the arrows that show the data-in transfers and data-out transfers.

Table 81 lists the SOP communication sequences supporting the SCSI transport protocol services described in 6.2.

Table 81 — Sequences of SOP communications

Sequence	Reference
Task management function	5.3.2
Non-data command	5.3.3
Read command	5.3.4
Write command	5.3.5
Bidirectional command	5.3.6

5.3.2 Task management function sequence

Figure 24 shows the sequence for a task management function (e.g., ABORT TASK (see SAM-5)), including the SCSI transport protocol services (see 6.2) invoked by the SCSI application layer.

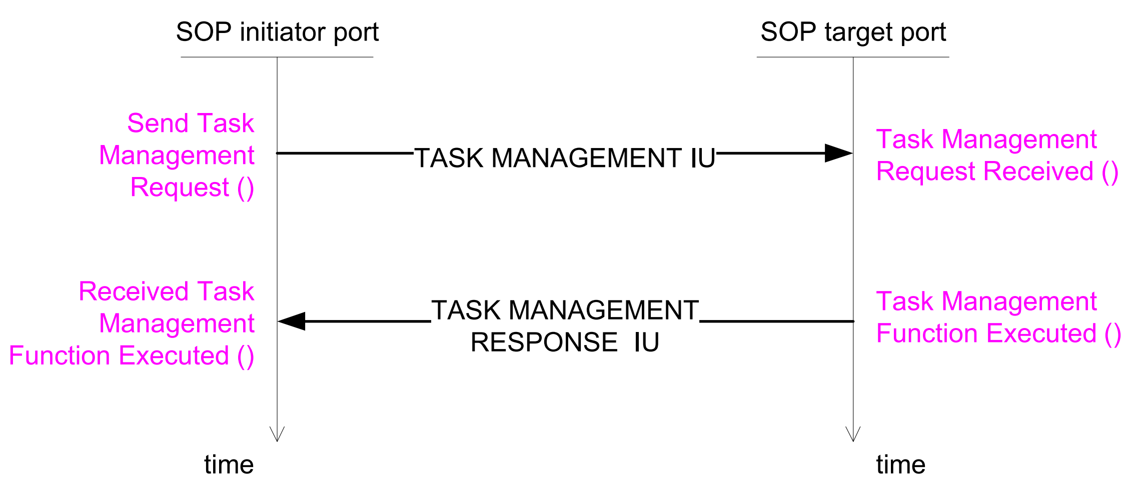


Figure 24 — Task management function sequence

5.3.3 Non-data command sequence

Figure 25 shows the sequence for a non-data command (e.g., TEST UNIT READY (see SPC-4)), including the SCSI transport protocol services (see 6.2) invoked by the SCSI application layer.

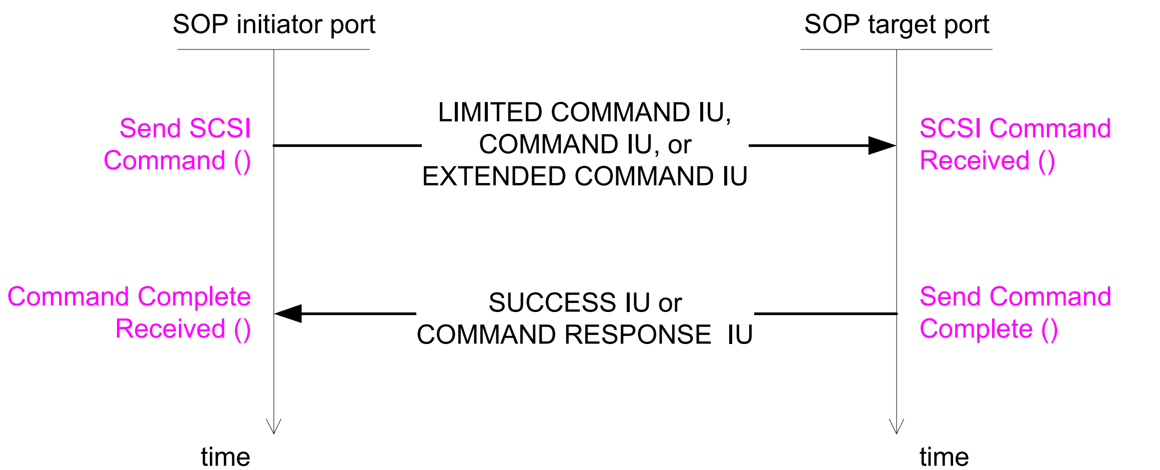


Figure 25 — Non-data command sequence

5.3.4 Read command sequence

Figure 26 shows the sequence for a read command (e.g., INQUIRY, REPORT LUNS, or MODE SENSE (see SPC-4)), including the SCSI transport protocol services (see 6.2) invoked by the SCSI application layer.

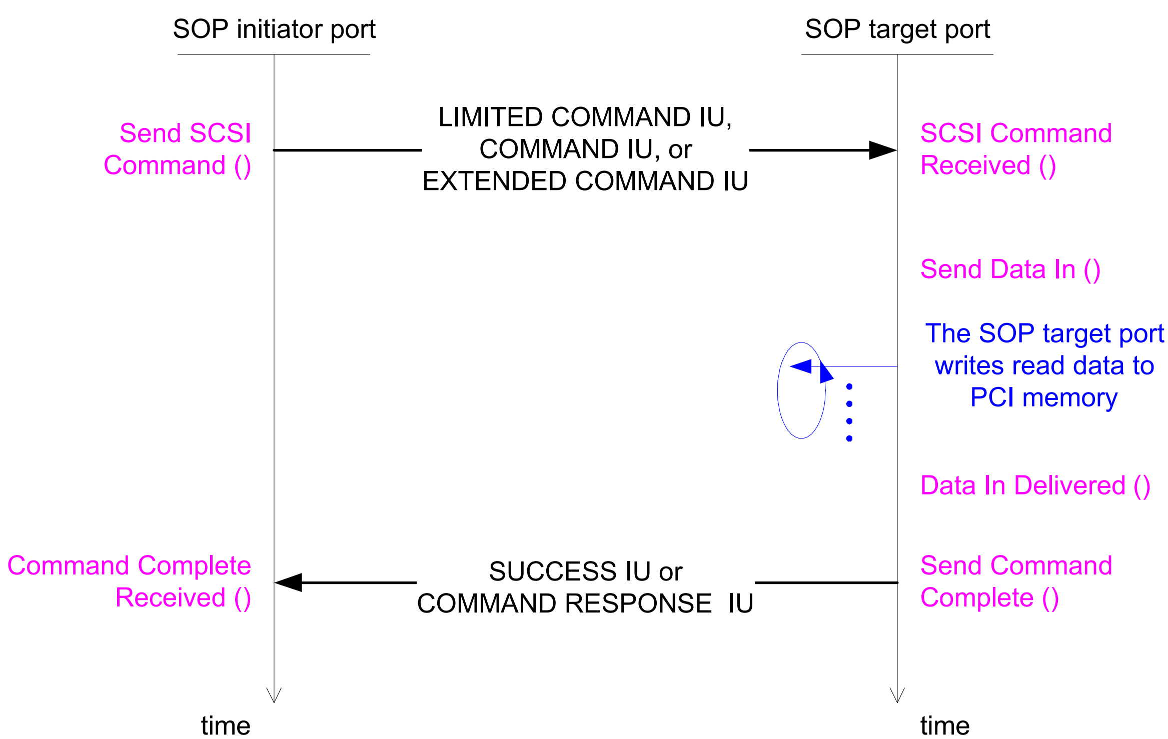


Figure 26 — Read command sequence

5.3.5 Write command sequence

Figure 27 shows the sequence for a write command (e.g., MODE SELECT (see SPC-4)), including the SCSI transport protocol services (see 6.2) invoked by the SCSI application layer.

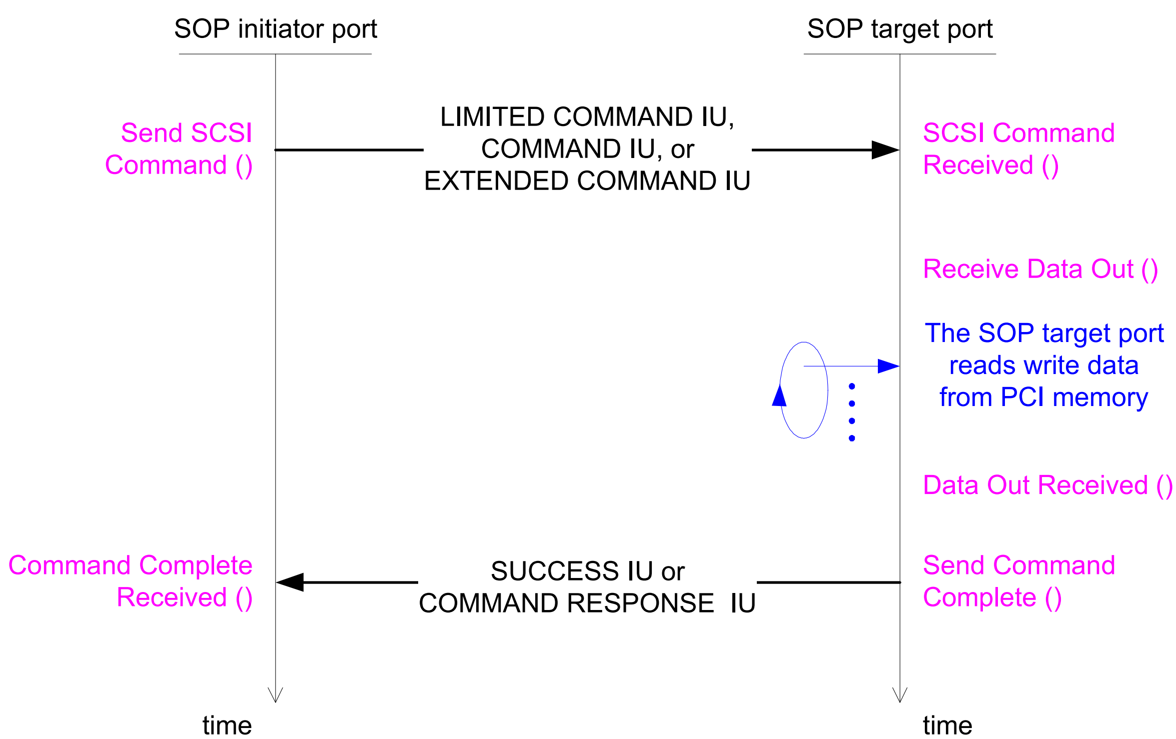


Figure 27 — Write command sequence

5.3.6 Bidirectional command sequence

Figure 28 shows the sequence for a bidirectional command (e.g., an XDWRITEREAD command (see SBC-3)), including the SCSI transport protocol services (see 6.2) invoked by the SCSI application layer.

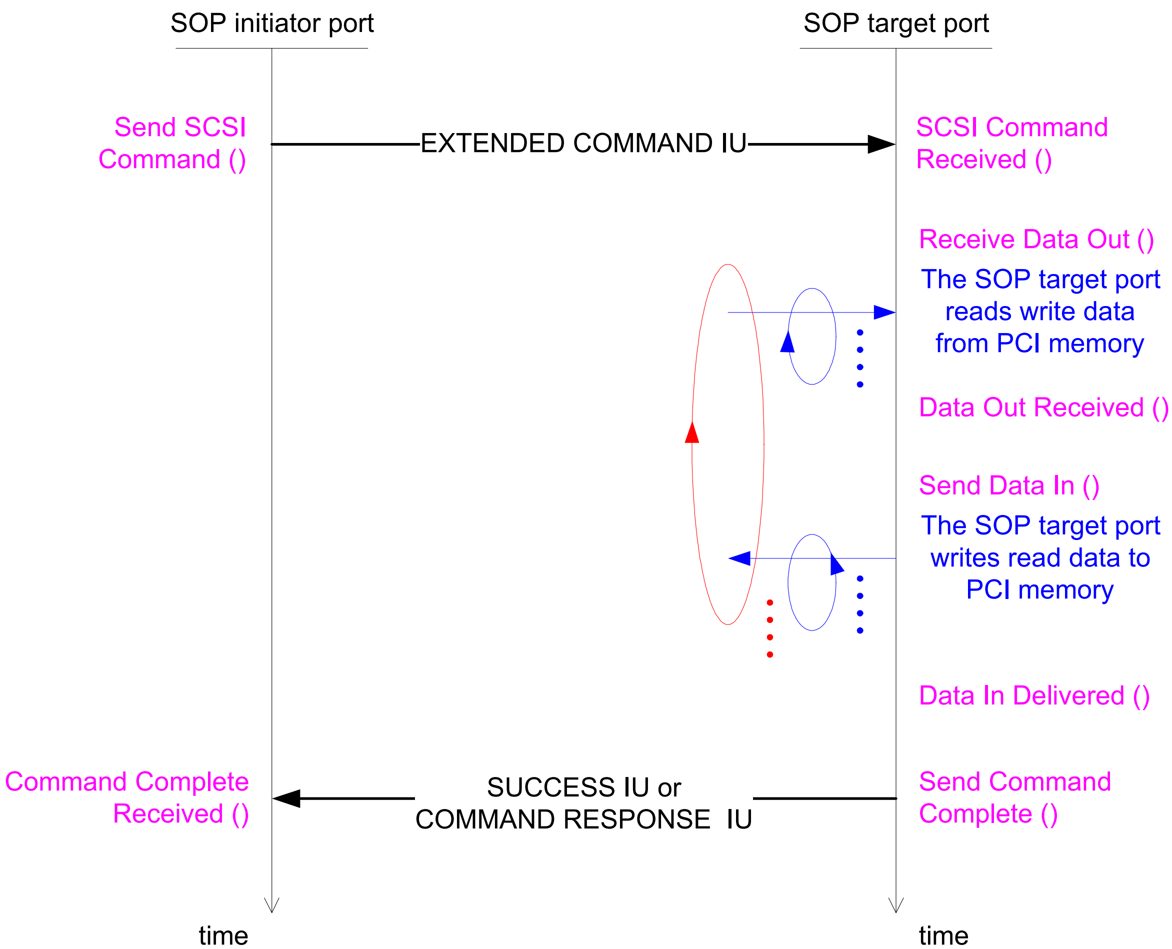


Figure 28 — Bidirectional command sequence

6 Application layer

6.1 Application layer overview

The application layer defines SCSI specific features.

6.2 SCSI transport protocol services

6.2.1 SCSI transport protocol services for SCSI commands

6.2.1.1 SCSI transport protocol services for SCSI commands overview

An application client requests the processing of a SCSI command by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following procedure call (see SAM-5):

Service response = **Execute Command** (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Priority]), OUT ([Data-In Buffer], [Sense Data], [Sense Data Length], Status, [Status Qualifier]))

This standard defines the transport protocol services required by SAM-5 in support of this procedure call. Table 82 describes the mapping of the Execute Command procedure call to transport protocol services and the SOP implementation of each transport protocol service.

Table 82 — Execute Command procedure call transport protocol services

Transport protocol service	I/T [a]	SOP implementation	Reference
Request/Confirmation			
Send SCSI Command request	I	SCSI command request IU	6.2.1.2
SCSI Command Received indication	T	Receipt of the SCSI command request IU	6.2.1.3
Send Command Complete response	T	SCSI command response IU	6.2.1.4
Command Complete Received confirmation	I	Receipt of the SCSI command response IU or problem sending the SCSI command request IU	6.2.1.5
Data-In Transfer [b]			
Send Data-In request	T	Memory write transactions	6.2.1.6
Data-In Delivered confirmation	T	Completion of memory write transactions	6.2.1.7
Data-Out Transfer [b]			
Receive Data-Out request	T	Memory read request transactions	6.2.1.8
Data-Out Received confirmation	T	Memory read completion transactions	6.2.1.9
Terminate Data Transfer [b]			
Terminate Data Transfer request	T		6.2.1.10
Data Transfer Terminated confirmation	T		6.2.1.11

[a] I/T indicates whether the SOP initiator port (I) or the SOP target port (T) implements the transport protocol service.
[b] Data transfer transport protocol services for SCSI initiator ports are not specified by SAM-5.

6.2.1.2 Send SCSI Command transport protocol service request

An application client invokes the Send SCSI Command operation in an SOP initiator port to request that the SOP initiator port send a SCSI command request IU.

Send SCSI Command (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Priority], [CRN], [First Burst Enabled], [Request Fence]))

To process this operation, the SOP initiator port:

a) constructs a SCSI command request IU; and
b) invokes the Add To Queue operation in the Host Queuing Layer object to add the SCSI command request IU to an inbound queue.

Table 83 shows how the arguments to the Send SCSI Command operation are used to construct the SCSI command request IU.

Table 83 — Send SCSI Command operation argument usage (part 1 of 2)

Argument	Usage
I_T_L_Q Nexus	I is ignored. T specifies: a) the SOP target port to which the SCSI command request IU is to be sent; and b) the NEXUS IDENTIFIER field, if any, in the SCSI command request IU [b]. L specifies the LOGICAL UNIT NUMBER field, if any, in the SCSI command request IU [b]. Q specifies the REQUEST IDENTIFIER field in the SCSI command request IU.
CDB	Specifies the CDB field in the SCSI command request IU.
Task Attribute	Specifies the TASK ATTRIBUTE field, if any, in the SCSI command request IU [b].
[Data-In Buffer Size] [a]	For a LIMITED COMMAND IU or a COMMAND IU, specifies the DATA BUFFER SIZE field. For an EXTENDED COMMAND IU, specifies the DATA-IN BUFFER SIZE field. Maximum of 2^{32} bytes.
[Data-Out Buffer]	For a LIMITED COMMAND IU or a COMMAND IU, the Data Buffer descriptor area describes the Data-Out Buffer. For an EXTENDED COMMAND IU, the Data-Out Buffer descriptor area describes the Data-Out Buffer.
[Data-Out Buffer Size]	For a LIMITED COMMAND IU or a COMMAND IU, specifies the DATA BUFFER SIZE field. For an EXTENDED COMMAND IU, specifies the DATA-OUT BUFFER SIZE field. Maximum of 2^{32} bytes.

[a] For a LIMITED COMMAND IU or a COMMAND IU, the Data-In Buffer descriptor area points to the Data-In Buffer. For an EXTENDED COMMAND IU, the Data-In Buffer descriptor area points to the Data-In Buffer.
[b] The LIMITED COMMAND IU does not have this field, but has an implied value for this field.

Table 83 — Send SCSI Command operation argument usage (part 2 of 2)

Argument	Usage
[CRN]	Specifies the PROTOCOL SPECIFIC field's COMMAND REFERENCE NUMBER field, if any, in the SCSI command request IU.
[Command Priority]	Specifies the COMMAND PRIORITY field, if any, in the SCSI command request IU [b].
[First Burst Enabled]	Ignored
[Request Fence]	Ignored

[a] For a LIMITED COMMAND IU or a COMMAND IU, the Data-In Buffer descriptor area points to the Data-In Buffer. For an EXTENDED COMMAND IU, the Data-In Buffer descriptor area points to the Data-In Buffer.
[b] The LIMITED COMMAND IU does not have this field, but has an implied value for this field.

6.2.1.3 SCSI Command Received transport protocol service indication

An SOP target port invokes the SCSI Command Received operation to notify a task manager that it has received a SCSI command request IU.

Table 84 shows how the arguments to the SCSI Command Received operation are set by the SOP target port.

Table 84 — SCSI Command Received operation argument sources

Argument	Usage
I_T_L_Q Nexus	I indicates: a) the SOP initiator port from which the SCSI command request IU was received [a]; and b) the value of the NEXUS IDENTIFIER field, if any, in the SCSI command request IU [c]. T indicates the SOP target port that invoked the operation (i.e., that received the SCSI command request IU). L specifies the LOGICAL UNIT NUMBER field, if any, in the SCSI command request IU [c]. Q specifies the REQUEST IDENTIFIER field in the SCSI command request IU [b].
CDB	Indicates the CDB field in the SCSI command request IU.
Task Attribute	Indicates the TASK ATTRIBUTE field, if any, in the SCSI command request IU [c].
[CRN]	Indicates the PROTOCOL SPECIFIC field's COMMAND REFERENCE NUMBER field, if any, in the SCSI command request IU.
[Command Priority]	Indicates the COMMAND PRIORITY field, if any, in the SCSI command request IU [c].
[First Burst Enabled]	Not included

[a] Each Device Queuing Layer object is assumed to be associated with one Host Queuing Layer object, and each SOP target port is assumed to be associated with one SOP initiator port through that pair of Queuing Layer objects.
[b] The request identifier is shared by all logical units behind an SOP target port (see 4.1.2.8.5).
[c] The LIMITED COMMAND IU does not have this field, but has an implied value for this field.

The SOP target port stores the DATA BUFFER SIZE field(s) and the contents of the Data Buffer descriptor area(s) received in the SCSI command request IU for use while processing the Send Data-In operation (see 6.2.1.6) and the Receive Data-Out operation (see 6.2.1.8).

6.2.1.4 Send Command Complete transport protocol service response

A device server invokes the Send Command Complete operation in an SOP target port to request that the SOP target port transmit a SCSI command response IU.

Send Command Complete (IN (I_T_L_Q Nexus, [Sense Data], [Sense Data Length], Status,
[Status Qualifier], Service Response, [Response Fence]))

To process this operation, the SOP target port:

1) constructs a SCSI command response IU; and
2) invokes the Add To Queue operation in the Host Queuing Layer object to add the SCSI command response IU to an outbound queue.

A SCSI device server shall only call Send Command Complete () after receiving SCSI Command Received ().

A SCSI device server shall not call Send Command Complete () for a given I_T_L_Q nexus until:

a) all outstanding Receive Data-Out () calls for that I_T_L_Q nexus have been responded to with Data-Out Received (); and
b) all outstanding Send Data-In () calls for that I_T_L_Q nexus have been responded to with Data-In Delivered ().

Table 85 shows how the arguments to the Send Command Complete operation are used to construct the SCSI command response IU.

Table 85 — Send Command Complete operation argument usage

Argument	Usage
I_T_L_Q Nexus	I Indicates: a) the SOP initiator port to which the SCSI command response IU is to be sent [a]; and b) the value of the NEXUS IDENTIFIER field, if any, in the SCSI command request IU [c]. T is implied by the SOP target port selected to process the operation. L is ignored [b]. Q specifies the REQUEST IDENTIFIER field in the SCSI command request IU.
[Sense Data]	Specifies the SENSE DATA field, if any, in the SCSI command response IU [c]. If present, causes the SOP target port to use a COMMAND RESPONSE IU rather than a SUCCESS IU.
[Sense Data Length]	Specifies the SENSE DATA LENGTH field, if any, in the SCSI command response IU [c].
Status	Specifies the STATUS field, if any, in the SCSI command response IU [c]. If not GOOD, causes the SOP target port to use a COMMAND RESPONSE IU rather than a SUCCESS IU.
[Status Qualifier]	Specifies the STATUS QUALIFIER field, if any, in the SCSI command response IU [c]. If present, causes the SOP target port to use a COMMAND RESPONSE IU rather than a SUCCESS IU.
Service Response	A value of COMMAND COMPLETE specifies that the SOP target port sends a SUCCESS IU or a COMMAND RESPONSE IU based on the other arguments.
[Response Fence]	Ignored

[a] Each Device Queuing Layer object is assumed to be associated with one Host Queuing Layer object, and each SOP target port is assumed to be associated with one SOP initiator port through that pair of Queuing Layer objects.
[b] Since the request identifier is shared by all logical units behind a SOP target port (see 4.1.2.8.5), the logical unit is implied by the request identifier.
[c] The SUCCESS IU does not contain this field, but has an implicit value for this field.

If:

a) the Service Response argument is set to COMMAND COMPLETE;
b) the Status argument is set to GOOD;
c) the Status Qualifier argument is not present;
d) the Sense Data argument is not present;
e) there was no Data-In Buffer underflow; and
f) there was no Data-Out Buffer underflow,

then the SOP target port shall send a SUCCESS IU. Otherwise, the SOP target port shall send a COMMAND RESPONSE IU.

In the COMMAND RESPONSE IU, if any, the SOP target port shall:

a) set the RESPONSE DATA LENGTH field to 0000h;
b) if there was an error processing the Data-In Buffer (e.g., in the SCSI command request IU, the PARTIAL IN bit is set to zero and the Data-In Buffer descriptor area does not completely describe the Data-In Buffer), then:
    A) set the DATA-IN TRANSFER RESULT field to DATA-IN BUFFER ERROR; and
    B) set the DATA-IN TRANSFERRED field to 00000000h;
c) if there was a Data-In Buffer underflow, then:
    A) set the DATA-IN TRANSFER RESULT field to DATA-IN BUFFER UNDERFLOW; and
    B) set the DATA-IN TRANSFERRED field to the number of data-in bytes transferred;
d) if there was neither a Data-In Buffer error nor a Data-In Buffer underflow, then set the DATA-IN TRANSFER RESULT field to DATA-IN BUFFER OK;
e) if there was an error processing the Data-Out Buffer (e.g., in the SCSI command request IU, the PARTIAL OUT bit is set to zero and the Data-Out Buffer descriptor area does not completely describe the Data-Out Buffer), then:
    A) set the DATA-OUT TRANSFER RESULT field to DATA-OUT BUFFER ERROR; and
    B) set the DATA-OUT TRANSFERRED field to 00000000h;
f) if there was a Data-Out Buffer underflow, then:
    A) set the DATA-OUT TRANSFER RESULT field to DATA-OUT BUFFER UNDERFLOW; and
    B) set the DATA-OUT TRANSFERRED field to the number of data-out bytes transferred; and
g) if there was neither a Data-Out Buffer error nor a Data-Out Buffer underflow, then set the DATA-OUT TRANSFER RESULT field to DATA-OUT BUFFER OK.

6.2.1.5 Command Complete Received transport protocol service confirmation

A SOP initiator port invokes the SCSI Command Received operation to notify an application client that it has received a SCSI command response IU or terminated a command because of an error.

Table 86 shows how the arguments to the Command Complete Received operation are set by the SOP initiator port.

Table 86 — Command Complete Received operation argument sources

Argument	Usage
I_T_L_Q Nexus	I indicates the SOP initiator port that invoked the operation. T indicates: a) the SOP target port from which the SCSI command response IU was received [a]; and b) the value of the NEXUS IDENTIFIER field in the SCSI command response IU. L indicates the logical unit, derived from the REQUEST IDENTIFIER field in the SCSI command response IU [b]. Q indicates the value of the REQUEST IDENTIFIER field in the SCSI command response IU.
[Data-In Buffer]	Represents the Data-In buffer described by the Data Buffer descriptor area.
[Sense Data]	Indicates the value of the SENSE DATA field, if any, in the SCSI command response IU [c].
[Sense Data Length]	The smaller of the value of the SENSE DATA LENGTH field, if any, in the SCSI command response IU [c] and the actual number of sense data bytes received by the SOP initiator port.
Status	Indicates the value of the STATUS field, if any, in the SCSI command response IU [c].
[Status Qualifier]	Indicates the value of the STATUS QUALIFIER field, if any, in the SCSI command response IU [c].
Service Response	A value of COMMAND COMPLETE indicates that: a) the DATA-IN TRANSFER RESULT field in the SCSI command response IU, if any, is set to a value indicating a Service Response of COMMAND COMPLETE (see table 75); b) the DATA-OUT TRANSFER RESULT field in the SCSI command response IU, if any, is set to a value indicating a Service Response of COMMAND COMPLETE (see table 76); and c) the RESPONSE DATA LENGTH field in the SCSI command response IU, if any, is set to 0000h in the SCSI command response IU [c]. A value of SERVICE DELIVERY OR TARGET FAILURE indicates that: a) the DATA-IN TRANSFER RESULT field in the SCSI command response IU is set to a value indicating a Service Response of SERVICE DELIVERY OR TARGET FAILURE; b) the DATA-OUT TRANSFER RESULT field in the SCSI command response IU is set to a value indicating a Service Response of SERVICE DELIVERY OR TARGET FAILURE; or c) the RESPONSE DATA field is present in the SCSI command response IU.

[a] Each Device Queuing Layer object is assumed to be associated with one Host Queuing Layer object, and each SOP target port is assumed to be associated with one SOP initiator port through that pair of Queuing Layer objects.
[b] Since the request identifier is shared by all logical units behind a SOP target port (see 4.1.2.8.5), the logical unit is implied by the request identifier.
[c] The SUCCESS IU does not contain this field, but has an implicit value for this field.

6.2.1.6 Send Data-In transport protocol service request

A SCSI device server uses the Send Data-In transport protocol service request to request that an SOP target port transmit data using memory write transactions.

Send Data-In (IN (I_T_L_Q Nexus, Device Server Buffer, Application Client Buffer Offset, Request Byte Count))

A SCSI device server shall only invoke Send Data-In () during a read command or bidirectional command.

A SCSI device server shall not invoke Send Data-In () for a given I_T_L_Q nexus after the SCSI device server has called Send Command Complete () for that I_T_L_Q nexus (e.g., a COMMAND RESPONSE IU for that I_T_L_Q nexus has been sent) or invoked Task Management Function Executed () for a task management function that terminates that command (e.g., an ABORT TASK).

Table 87 shows how the arguments to the Send Data-In transport protocol service are used.

Table 87 — Send Data-In operation argument usage

Argument	Usage
I_T_L_Q Nexus	Specifies the command whose Data Buffer descriptor area is to be used
Device Server Buffer	Specifies the source of the data, internal to the SCSI device server
Application Client Buffer Offset	Specifies the destination offset within the Data-In Buffer described by the Data Buffer descriptor area
Request Byte Count	Specifies the amount of data to transfer

If the Send Data-In operation requests a data transfer beyond the size specified by the DATA BUFFER SIZE field, then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE - DATA-IN BUFFER OVERFLOW - DATA BUFFER SIZE.

If the Send Data-In operation requests a data transfer:

a) within the length specified by the DATA BUFFER SIZE field; and
b) beyond the length of the Data-In Buffer described by the Data Buffer descriptor area (e.g., in PQI, implied by the LENGTH fields in the SGL),

then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE - DATA-IN BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA.

If the Send Data-In operation requests a data transfer that results in a PCIe fabric error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE FABRIC ERROR.

If the Send Data-In operation requests a data transfer that results in a PCIe completion timeout error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE COMPLETION TIMEOUT.

If the Send Data-In operation requests a data transfer that results in a PCIe completer abort error, then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE COMPLETER ABORT.

If the Send Data-In operation requests a data transfer that results in a PCIe poisoned TLP received error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE POISONED TLP RECEIVED.

If the Send Data-In operation requests a data transfer that results in a PCIe ECRC check failed error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE ECRC CHECK FAILED.

If the Send Data-In operation requests a data transfer that results in a PCIe unsupported request error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE UNSUPPORTED REQUEST.

If the Send Data-In operation requests a data transfer that results in a PCIe ACS violation error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE ACS VIOLATION.

If the Send Data-In operation requests a data transfer that results in a PCIe TLP prefix blocked error then the SOP target port shall invoke the Data-In Delivered operation with the Delivery Result set to DELIVERY FAILURE – PCIE TLP PREFIX BLOCKED.

6.2.1.7 Data-In Delivered transport protocol service confirmation

A SOP target port uses the Data-In Delivered transport protocol service indication to notify a SCSI device server of the results of transmitting data using memory write transactions.

Data-In Delivered (IN (I_T_L_Q Nexus, Delivery Result))

Table 88 shows how the arguments to the Data-In Delivered transport protocol service are determined.

Table 88 — Data-In Delivered operation argument sources

Argument	Usage
I_T_L_Q Nexus	Indicates the command whose Data Buffer descriptor area was used
Delivery Result	Set to either: a) DELIVERY SUCCESSFUL if there were no problems during the data transfer(s); b) DELIVERY FAILURE if the Data-In Buffer was too short; or c) an unrecovered PCI Express fabric error has occurred.

6.2.1.8 Receive Data-Out transport protocol service request

A SCSI device server uses the Receive Data-Out transport protocol service request to request that an SOP target port transmit data using memory read transactions.

Receive Data-Out (IN (I_T_L_Q Nexus, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))

A SCSI device server shall only call Receive Data-Out () during a write command or bidirectional command.

A SCSI device server shall not call Receive Data-Out () for a given I_T_L_Q nexus after a Send Command Complete () has been called for that I_T_L_Q nexus or after a Task Management Function Executed () has been called for a task management function that terminates that command (e.g., an ABORT TASK).

Table 89 shows how the arguments to the Receive Data-Out transport protocol service are used.

Table 89 — Receive Data-Out operation argument usage

Argument	Usage
I_T_L_Q Nexus	Specifies the command whose Data Buffer descriptor area is to be used
Device Server Buffer	Specifies the destination of the data, internal to the SCSI device server
Application Client Buffer Offset	Specifies the source offset within the Data-Out Buffer described by the Data Buffer descriptor area
Request Byte Count	Specifies the amount of data to transfer

If the Receive Data-Out operation requests a data transfer beyond the size specified by the DATA BUFFER SIZE field, then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE - DATA-OUT BUFFER OVERFLOW - DATA BUFFER SIZE.

If the Receive Data-Out operation requests a data transfer:

a) within the length specified by the DATA BUFFER SIZE field; and
b) beyond the length of the Data-Out Buffer described by the Data Buffer descriptor area (e.g.,in PQI, implied by the LENGTH fields in the SGL),

then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE - DATA-OUT BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA.

If the Receive Data-Out operation requests a data transfer that results in a PCIe fabric error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE FABRIC ERROR.

If the Receive Data-Out operation requests a data transfer that results in a PCIe completion timeout error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE COMPLETION TIMEOUT.

If the Receive Data-Out operation requests a data transfer that results in a PCIe completer abort error, then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE COMPLETER ABORT.

If the Receive Data-Out operation requests a data transfer that results in a PCIe poisoned TLP received error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE POISONED TLP RECEIVED.

If the Receive Data-Out operation requests a data transfer that results in a PCIe ECRC check failed error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE ECRC CHECK FAILED.

If the Receive Data-Out operation requests a data transfer that results in a PCIe unsupported request error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE UNSUPPORTED REQUEST.

If the Receive Data-Out operation requests a data transfer that results in a PCIe ACS violation error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE ACS VIOLATION.

If the Receive Data-Out operation requests a data transfer that results in a PCIe TLP prefix blocked error then the SOP target port shall invoke the Data-Out Received operation with the Delivery Result set to DELIVERY FAILURE – PCIE TLP PREFIX BLOCKED.

6.2.1.9 Data-Out Received transport protocol service confirmation

An SOP target port uses the Data-Out Received transport protocol service indication to notify a SCSI device server of the result of transmitting data with memory read transactions.

Data-Out Received (IN (I_T_L_Q Nexus, Delivery Result))

Table 90 shows how the arguments to the Data-Out Received transport protocol service are determined.

Table 90 — Data-Out Received operation argument sources

Argument	Usage
I_T_L_Q Nexus	Indicates the command whose SGL was used
Delivery Result	Set to either: a) DELIVERY SUCCESSFUL if there were no problems during the data transfer(s); or b) DELIVERY FAILURE if: A) the Data-Out buffer described by the SGL was too short; or B) an unrecovered PCI Express fabric error has occurred.

6.2.1.10 Terminate Data Transfer transport protocol service request

A SCSI device server uses the Terminate Data Transfer transport protocol service request to request that an SOP target port terminate any Send Data-In () or Receive Data-Out () transport protocol services, if any, being processed using the specified nexus.

Terminate Data Transfer (IN (Nexus))

Table 91 shows how the arguments to the Terminate Data Transfer transport protocol service are used.

Table 91 — Terminate Data Transfer operation argument usage

Argument	Usage
Nexus	Specifies the scope of the data transfer(s) to terminate

6.2.1.11 Data Transfer Terminated transport protocol service confirmation

An SOP target port uses the Data Transfer Terminated transport protocol service indication to notify a SCSI device server that all data transfers for the indicated nexus have been terminated.

Data Transfer Terminated (IN (Nexus))

Table 92 shows how the arguments to the Data Transfer Terminated transport protocol service are determined.

Table 92 — Data Transfer Terminated operation argument sources

Argument	Usage
Nexus	Indicates the scope of the data transfer(s) terminated

6.2.2 SCSI transport protocol services for task management functions

6.2.2.1 SCSI transport protocol services for task management functions overview

An application client requests the processing of a SCSI task management function by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following procedure calls (see SAM-5):

a) Service Response = ABORT TASK (IN (Nexus));
b) Service Response = ABORT TASK SET (IN (Nexus));
c) Service Response = CLEAR ACA (IN (Nexus));
d) Service Response = CLEAR TASK SET (IN (Nexus));
e) Service Response = I_T NEXUS RESET (IN (Nexus));
f) Service Response = LOGICAL UNIT RESET (IN (Nexus));
g) Service Response = QUERY TASK (IN (Nexus));
h) Service Response = QUERY TASK SET (IN (Nexus), OUT ([Additional Response Information])); and
i) Service Response = QUERY ASYNCHRONOUS EVENT (IN (Nexus), OUT ([Additional Response Information])).

This standard defines the transport protocol services required by SAM-5 in support of these procedure calls.

Table 93 describes the mapping of these procedure calls to transport protocol services and the SOP implementation of each transport protocol service.

Table 93 — Task management function procedure call transport protocol services

Transport protocol service	I/T [a]	SOP implementation	Reference
Request/Confirmation			
Send Task Management Request request	I	TASK MANAGEMENT IU	6.2.2.2
Task Management Request Received indication	T	Receipt of the TASK MANAGEMENT IU	6.2.2.3
Task Management Function Executed response	T	TASK MANAGEMENT RESPONSE IU	6.2.2.4
Received Task Management Function Executed confirmation	I	Receipt of the TASK MANAGEMENT RESPONSE IU or problem sending the TASK MANAGEMENT IU	6.2.2.5

[a] I/T indicates whether the SOP initiator port (I) or the SOP target port (T) implements the transport protocol service.

6.2.2.2 Send Task Management Request transport protocol service request

A SCSI application client uses the Send Task Management Request transport protocol service request to request that an SOP initiator port transmit a TASK MANAGEMENT IU.

Send Task Management Request (IN (Nexus, Function Identifier, Task Management Tag, [Request Fence]))

Table 94 shows how the arguments to the Send Task Management Request transport protocol service are used.

Table 94 — Send Task Management Request operation argument usage

Argument	Usage
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier) where: a) I is ignored; b) T specifies: A) the SOP target port to which the TASK MANAGEMENT IU is to be sent; and B) the NEXUS IDENTIFIER field in the TASK MANAGEMENT IU; c) L specifies the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU; and d) Q specifies the REQUEST IDENTIFIER field in the TASK MANAGEMENT IU
Function Identifier	Specifies the TASK MANAGEMENT FUNCTION field in the TASK MANAGEMENT IU.
Task Management Tag	Specifies the REQUEST IDENTIFIER field in the TASK MANAGEMENT IU.
[Request Fence]	Ignored

6.2.2.3 Task Management Request Received SCSI transport protocol service indication

A SOP target port uses the Task Management Request Received transport protocol service indication to notify a task manager that the SOP target port has received a TASK MANAGEMENT IU.

Task Management Request Received (IN (Nexus, Function Identifier, Task Management Tag))

Table 95 shows how the arguments to the Task Management Request Received transport protocol service are determined.

Table 95 — Task Management Request Received operation argument sources

Argument	Usage
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier) where: a) I indicates the SOP initiator port from which the TASK MANAGEMENT IU was received [a]; b) T indicates: A) the SOP target port that invoked the operation (i.e., that received the TASK MANAGEMENT IU); and B) the value of the NEXUS IDENTIFIER field in the TASK MANAGEMENT IU; c) L indicates the value of the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU; and d) Q indicates the value of the REQUEST IDENTIFIER field in the TASK MANAGEMENT IU
Function Identifier	Indicates the TASK MANAGEMENT FUNCTION field in the TASK MANAGEMENT IU.
Task Management Tag	Indicates the REQUEST IDENTIFIER field in the TASK MANAGEMENT IU.
[a] Each Device Queuing Layer object is assumed to be associated with one Host Queuing Layer object, and each SOP target port is assumed to be associated with one SOP initiator port through that pair of Queuing Layer objects.	

6.2.2.4 Task Management Function Executed SCSI transport protocol service response

A task manager uses the Task Management Function Executed transport protocol service response to request that an SOP target port transmit a TASK MANAGMENT RESPONSE IU.

Task Management Function Executed (IN (Nexus, Service Response, [Additional Response Information], Task Management Tag, [Response Fence]))

A task manager shall only call Task Management Function Executed () after receiving Task Management Request Received ().

Table 96 shows how the arguments to the Task Management Function Executed transport protocol service are used.

Table 96 — Task Management Function Executed operation argument usage

Argument	Usage
Nexus	I specifies: a) the SOP initiator port to which the TASK MANAGEMENT RESPONSE IU is to be sent [a]; and b) the NEXUS IDENTIFIER field in the TASK MANAGEMENT RESPONSE IU. T is implied by the SOP target port selected to process the operation. L is ignored [b]. Q specifies the REQUEST IDENTIFIER field in the TASK MANAGEMENT RESPONSE IU.
Service Response	Specifies the RESPONSE CODE field in the TASK MANAGEMENT RESPONSE IU.
[Additional Response Information]	Specifies the ADDITIONAL RESPONSE INFORMATION field in the TASK MANAGEMENT RESPONSE IU.
Task Management Tag	Specifies the REQUEST IDENTIFIER field in the TASK MANAGEMENT RESPONSE IU.
[Response Fence]	Ignored

[a] Each Device Queuing Layer object is assumed to be associated with one Host Queuing Layer object, and each SOP target port is assumed to be associated with one SOP initiator port through that pair of Queuing Layer objects.

[b] Since the request identifier is shared by all logical units behind an SOP target port (see 4.1.2.8.5), the logical unit is implied by the request identifier.

6.2.2.5 Received Task Management Function Executed transport protocol service confirmation

An SOP initiator port uses the Received Task Management Function Executed transport protocol service confirmation to notify a SCSI application client that the SOP initiator port has received a TASK MANAGEMENT RESPONSE IU.

Received Task Management Function Executed (IN (Nexus, Service Response, [Additional Response Information], Task Management Tag))

Table 97 shows how the arguments to the Received Task Management Function Executed transport protocol service are determined.

Table 97 — Received Task Management Function Executed operation argument sources

Argument	Usage
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the function), where: a) I indicates the SOP initiator port that invoked the operation; b) T indicates: A) the SOP target port from which the TASK MANAGEMENT RESPONSE IU was received; and B) the value of the NEXUS IDENTIFIER field, if any, in the TASK MANAGEMENT RESPONSE IU; c) L indicates the logical unit, derived from the REQUEST IDENTIFIER field in the TASK MANAGEMENT RESPONSE IU [a]; and d) Q indicates the value of the REQUEST IDENTIFIER field in the TASK MANAGEMENT RESPONSE IU.
Service Response	Set to: a) FUNCTION COMPLETE if the TASK MANAGEMENT RESPONSE IU contains response data with a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION COMPLETE (see table 79); b) FUNCTION SUCCEEDED if the TASK MANAGEMENT RESPONSE IU contains response data with a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION SUCCEEDED; c) FUNCTION REJECTED if the TASK MANAGEMENT RESPONSE IU contains response data with a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION REJECTED; d) INCORRECT LOGICAL UNIT NUMBER if the TASK MANAGEMENT RESPONSE IU contains response data with a RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER; or e) SERVICE DELIVERY OR TARGET FAILURE if the TASK MANAGEMENT RESPONSE IU contains response data with a RESPONSE CODE field set to: A) TASK MANAGEMENT FUNCTION FAILED; B) OVERLAPPED REQUEST IDENTIFIER ATTEMPTED; C) INVALID FIELD IN INFORMATION UNIT; or D) a reserved or unsupported value.
[Additional Response Information]	Indicates the ADDITIONAL RESPONSE INFORMATION field in the TASK MANAGEMENT RESPONSE IU.
Task Management Tag	Indicates the REQUEST IDENTIFIER field in the TASK MANAGEMENT RESPONSE IU.

[a] Since the request identifier is shared by all logical units behind an SOP target port (see 4.1.2.8.5), the logical unit is implied by the request identifier.

6.2.3 SCSI transport protocol services for event notification

6.2.3.1 Transport Reset transport protocol service indication

To process a reset event (see 4.7.1 and SAM-5), a SCSI target port invokes the following operation in each device server and task manager, and a SCSI initiator port invokes the following operation in each application client:

Transport Reset (IN (SCSI Port))

Table 98 shows how an SOP port that is a SCSI port sets the arguments to the Transport Reset operation.

Table 98 — Transport Reset operation arguments

Argument	Source
SCSI Port	Indicates the SCSI port that detected the transport reset.

6.2.3.2 Nexus Loss transport protocol service indication

To process an I_T nexus loss event (see 4.7.2 and SAM-5), a SCSI target port invokes the following operation in each device server and task manager, and a SCSI initiator port invokes the following operation in each application client:

Nexus Loss (IN (I_T Nexus))

The Nexus Loss operation is not invoked by SOP ports.

6.2.3.3 Power Loss Expected transport protocol service indication

To process a power loss expected event (see 4.7.3 and SAM-5), a SCSI target port invokes the following operation in each device server and task manager:

Power Loss Expected (IN (SCSI Port))

The Power Loss Expected operation is not invoked by SOP target ports.

6.3 Configuration parameters

6.3.1 Summary of configuration parameter page codes

Table 99 defines the page code assignments for configuration parameters.

Table 99 — PAGE CODE field and SUBPAGE CODE field

Code	Subpage	Description	Support	Reference
00h to 0Fh	00h to FFh	Reserved		
10h to BFh	00h to FFh	Reserved		
C0h to FFh	00h to FFh	Vendor Specific		

6.3.2 Configuration parameters overview

This subclause describes the configuration parameter header and configuration pages used with a SET CONFIGURATION command or a REPORT CONFIGURATION command that are applicable to a SOP device.

Subpages are identical to configuration pages except they include a SUBPAGE CODE field that further differentiates the configuration page contents.

6.3.3 Configuration parameter list format

Table 100 defines the configuration parameter list.

Table 100 — Configuration parameter list format

Byte \ Bit	7	6	5	4	3	2	1	0
1 ••• 7	Configuration parameter header (see table 101)							
8 ••• n	Configuration parameters							

6.3.4 Configuration parameter header format

The configuration parameter header that is used by the REPORT CONFIGURATION IU and SET CONFIGURATION IU is defined in table 101.

Table 101 — Configuration parameter header

Byte \ Bit	7	6	5	4	3	2	1	0
0	PS	Reserved			PAGE CONTROL			
1	PAGE CODE							
2	SUBPAGE CODE							
3	Reserved							
4 ••• 7	PAGE LENGTH							

When using the REPORT CONFIGURATION IU request, a page saveable (PS) bit set to one indicates that the configuration page may be saved by the SOP device in a nonvolatile, vendor specific location. A PS bit set to zero indicates that the SOP device is not able to save the supported configuration page. When using a SET CONFIGURATION IU request, the PS bit is reserved.

The PAGE CODE field and the SUBPAGE CODE field are defined in table 99.

The PAGE LENGTH field specifies the length in bytes of the configuration parameters that follow. If the SOP management application client does not set this value to the value that is returned for the configuration page by the REPORT CONFIGURATION IU, the SOP management device server shall return a MANAGEMENT RESPONSE IU with the RESULT field set to INVALID FIELD IN REQUEST IU. The SOP management device server may implement a configuration page that is less than the full configuration page, provided no field is truncated and the page length correctly specifies the actual length implemented.

6.4 Task router and task manager error handling

6.4.1 Request identifier overlap handling summary

Table 102 summarizes the responses of a SCSI target device that performs request identifier checking and receives an IU with a request identifier that overlaps one already being processed.

Table 102 — Request identifier overlap handling summary

Received	Overlaps with	Response	Reference
SCSI command	SCSI command	CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to OVERLAPPED COMMANDS ATTEMPTED	6.4.2
	Task management function or SOP management function	CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to OVERLAPPED COMMANDS ATTEMPTED	6.4.3
Task management function	SCSI command, task management function, or SOP management function	TASK MANAGEMENT RESPONSE IU with the RESPONSE CODE field set to OVERLAPPED REQUEST IDENTIFIER ATTEMPTED	6.4.4
SOP management function	SCSI command, task management function, or SOP management function	MANAGEMENT RESPONSE IU with the RESULT field set to INVALID FIELD IN REQUEST IU	6.4.5

6.4.2 SCSI command overlap with SCSI command

If a SCSI target device performs request identifier checking and a task router or task manager processes SCSI Command Received () with a request identifier already in use by another command (i.e., an overlapped command (see SAM-5)) in any logical unit, then:

a) the task router or task manager(s) shall abort all task management functions received on that I_T nexus;
b) the SOP management device server shall abort all SOP management functions received on that I_T nexus; and
c) the task router or task manager(s) shall respond to the overlapped command as defined in SAM-5 (i.e., abort all commands received on that I_T nexus and return CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to OVERLAPPED COMMANDS ATTEMPTED).

6.4.3 SCSI command overlap with task management function or SOP management function

If a SCSI target device performs request identifier checking and a task router or task manager processes SCSI Command Received () with a request identifier already in use by:

a) a task management function in any logical unit; or
b) an SOP management function,

then:

a) the task router or task manager(s) shall abort all commands received on that I_T nexus;
b) the task router or task manager(s) shall abort all task management functions received on that I_T nexus;
c) the SOP management device server shall abort all SOP management functions received on that I_T nexus; and
d) the task router or task manager(s) shall call Send Command Complete () with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to OVERLAPPED COMMANDS ATTEMPTED.

6.4.4 Task management function overlap with SCSI command, task management function or SOP management function

If a SCSI target device performs request identifier checking and a task router or task manager processes Task Management Request Received () with a request identifier already in use by:

a) a SCSI command in any logical unit;
b) a task management function in any logical unit; or
c) an SOP management function,

then:

a) the task router or task manager(s) shall abort all commands received on that I_T nexus;
b) the task router or task manager(s) shall abort all task management functions received on that I_T nexus;
c) the SOP management device server shall abort all SOP management functions received on that I_T nexus; and
d) the task router or task manager(s) shall call Task Management Function Executed () with the Service Response set to SERVICE DELIVERY OR TARGET FAILURE - Overlapped Request Identifier Attempted (i.e., requesting that the SOP target port send a TASK MANAGEMENT RESPONSE IU with the RESPONSE CODE field set to OVERLAPPED REQUEST IDENTIFIER ATTEMPTED).

6.4.5 SOP management function overlap with SCSI command, task management function or SOP management function

If a SCSI target device performs request identifier checking and the SOP management device server processes an SOP management function with a request identifier already in use by:

a) a SCSI command in any logical unit;
b) a task management function in any logical unit; or
c) an SOP management function,

then:

a) the task router or task manager(s) shall abort all commands received on that I_T nexus;
b) the task router or task manager(s) shall abort all task management functions received on that I_T nexus;
c) the SOP management device server shall abort all SOP management functions received on that I_T nexus; and
d) the SOP management device server shall return a MANAGEMENT RESPONSE IU with the RESULT field set to INVALID FIELD IN REQUEST IU.

6.5 Device server error handling

6.5.1 Device server Data-In Buffer error handling

If the device server is not a bridge device server and processes a Data-In Delivered operation with a Delivery Result set to DELIVERY FAILURE for one of the reasons in table 103, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set as specified in table 103.

Table 103 — Data-In Delivered Delivery Result of DELIVERY FAILURE handling

Delivery Result	Additional sense code	DATA-IN TRANSFER RESULT field
DELIVERY FAILURE - DATA-IN BUFFER ERROR	DATA-IN BUFFER ERROR	DATA-IN BUFFER ERROR
DELIVERY FAILURE - DATA-IN BUFFER OVERFLOW - DATA BUFFER SIZE	DATA-IN BUFFER OVERFLOW - DATA BUFFER SIZE	DATA-IN BUFFER OVERFLOW - DATA BUFFER SIZE
DELIVERY FAILURE - DATA-IN BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA	DATA-IN BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA	DATA-IN BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA
DELIVERY FAILURE – PCIE FABRIC ERROR	PCIE FABRIC ERROR	PCIE FABRIC ERROR
DELIVERY FAILURE – PCIE COMPLETION TIMEOUT	PCIE COMPLETION TIMEOUT	PCIE COMPLETION TIMEOUT
DELIVERY FAILURE – PCIE COMPLETER ABORT	PCIE COMPLETER ABORT	PCIE COMPLETER ABORT
DELIVERY FAILURE – PCIE POISONED TLP RECEIVED	PCIE POISONED TLP RECEIVED	PCIE POISONED TLP RECEIVED
DELIVERY FAILURE – PCIE ECRC CHECK FAILED	PCIE ECRC CHECK FAILED	PCIE ECRC CHECK FAILED
DELIVERY FAILURE – PCIE UNSUPPORTED REQUEST	PCIE UNSUPPORTED REQUEST	PCIE UNSUPPORTED REQUEST
DELIVERY FAILURE – PCIE ACS VIOLATION	PCIE ACS VIOLATION	PCIE ACS VIOLATION
DELIVERY FAILURE – PCIE TLP PREFIX BLOCKED	PCIE TLP PREFIX BLOCKED	PCIE TLP PREFIX BLOCKED

If the device server is a bridge device server and processes a Data-In Delivered operation with a Delivery Result set to DELIVERY FAILURE for one of the reasons in table 103, then the device server shall set the DATA-IN TRANSFER RESULT field to the value specified in table 103 when processing the Send Command Complete operation for the command.

If a bridge application client processes a Command Complete Received operation with a SCSI command response IU indicating a Data-In Buffer overflow (see B.4), then the device server shall set the DATA-IN TRANSFER RESULT field to DATA-IN OVERFLOW - BRIDGE LOCAL PORT when processing the Send Command Complete operation for the command.

6.5.2 Device server Data-Out Buffer error handling

If the device server is not a bridge device server and processes a Data-Out Received operation with a Delivery Result set to DELIVERY FAILURE for one of the reasons in table 104, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set as indicated in table 104.

Table 104 — Data-Out Received Delivery Result of DELIVERY FAILURE handling

Delivery Result	Additional sense code	DATA-OUT TRANSFER RESULT field
DELIVERY FAILURE - DATA-OUT BUFFER ERROR	DATA-OUT BUFFER ERROR	DATA-OUT BUFFER ERROR
DELIVERY FAILURE - DATA-OUT BUFFER OVERFLOW - DATA BUFFER SIZE	DATA-OUT BUFFER OVERFLOW - DATA BUFFER SIZE	DATA-OUT BUFFER OVERFLOW - DATA BUFFER SIZE
DELIVERY FAILURE - DATA-OUT BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA	DATA-OUT BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA	DATA-OUT BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA
DELIVERY FAILURE – PCIE FABRIC ERROR	PCIE FABRIC ERROR	PCIE FABRIC ERROR
DELIVERY FAILURE – PCIE COMPLETION TIMEOUT	PCIE COMPLETION TIMEOUT	PCIE COMPLETION TIMEOUT
DELIVERY FAILURE – PCIE COMPLETER ABORT	PCIE COMPLETER ABORT	PCIE COMPLETER ABORT
DELIVERY FAILURE – PCIE POISONED TLP RECEIVED	PCIE POISONED TLP RECEIVED	PCIE POISONED TLP RECEIVED
DELIVERY FAILURE – PCIE ECRC CHECK FAILED	PCIE ECRC CHECK FAILED	PCIE ECRC CHECK FAILED
DELIVERY FAILURE – PCIE UNSUPPORTED REQUEST	PCIE UNSUPPORTED REQUEST	PCIE UNSUPPORTED REQUEST
DELIVERY FAILURE – PCIE ACS VIOLATION	PCIE ACS VIOLATION	PCIE ACS VIOLATION
DELIVERY FAILURE – PCIE TLP PREFIX BLOCKED	PCIE TLP PREFIX BLOCKED	PCIE TLP PREFIX BLOCKED

If the device server is a bridge device server and processes a Data-Out Received operation with a Delivery Result set to DELIVERY FAILURE for one of the reasons in table 104, then the device server shall set the DATA-OUT TRANSFER RESULT field to the value specified in table 104 when processing the Send Command Complete operation for the command.

If a bridge application client processes a Command Complete Received operation with a SCSI command response IU indicating a Data-Out Buffer overflow (see B.4), then the device server shall set the DATA-OUT TRANSFER RESULT field to DATA-OUT OVERFLOW - BRIDGE LOCAL PORT when processing the Send Command Complete operation for the command

6.6 SCSI commands

6.6.1 INQUIRY command

SOP-specific vital product data accessed with the INQUIRY command (see SPC-4) is described in 6.8.

6.6.2 MODE SELECT and MODE SENSE commands

SOP-specific mode pages accessed with the MODE SELECT command and MODE SENSE command (see SPC-4) are described in 6.7.

6.7 SCSI mode parameters

6.7.1 SCSI mode parameter overview

Table 105 defines mode pages supported by logical units in SCSI target devices in SOP domains (i.e., with SOP target ports) that support the MODE SELECT command or MODE SENSE command.

Table 105 — SOP target port mode pages

Mode page code	Subpage code	Description	Reference
02h	00h	Disconnect-Reconnect mode page	6.7.2
18h	00h	Protocol-Specific Logical Unit mode page	6.7.3
	01h to DFh	Reserved	
	E0h to FEh	Vendor specific	
	FFh	Return all subpages for this mode page code	SPC-4
19h	00h	Protocol-Specific Logical Unit mode page	6.7.4
	01h to DFh	Reserved	
	E0h to FEh	Vendor specific	
	FFh	Return all subpages for this mode page code	SPC-4

If any field in an implemented mode page is not implemented, then the value of the field shall be assumed to be zero (i.e., as if the field is set to zero) (see SPC-4).

If a mode page defined by this standard is not implemented, then the value of each field in that mode page that is:

a) allowed by this standard to be changeable (e.g., not defined as a read only field); and
b) not used solely to define the mode page structure or coordinate access to the mode page,

shall be assumed to be zero (i.e., as if the mode page is implemented and the field is set to zero).

6.7.2 Disconnect-Reconnect mode page

The Disconnect-Reconnect mode page (see SPC-4) provides the application client the means to tune the performance of a service delivery subsystem. None of the parameters are applicable to SOP, as depicted in table 106.

Table 106 — Disconnect-Reconnect mode page for SOP

Byte \ Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (02h)					
1	PAGE LENGTH (0Eh)							
2 ••• 15	Reserved							

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SPF bit, the PAGE CODE field, and the PAGE LENGTH field are defined in SPC-4 and are set to the values shown in table 106.

If any of the reserved bytes are set to a non-zero value, then the device server shall terminate the MODE SELECT command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

6.7.3 Protocol-Specific Logical Unit mode page

The Protocol-Specific Logical Unit mode page (see SPC-4) contains parameters that affect SOP target port operation on behalf of the logical unit.

The mode page policy (see SPC-4) for this mode page shall be either shared or per target port. If the SOP target device has multiple SOP target ports, then the mode page policy should be per target port.

Parameters in this mode page:

a) shall affect only the SOP target port if the mode page policy is per target port; or
b) shall affect all SOP target ports in the SOP target device if the mode page policy is shared.

Table 107 defines the format of the page for SOP.

Table 107 — Protocol-Specific Logical Unit mode page for SOP

Byte \ Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (18h)					
1	PAGE LENGTH (02h)							
2	Reserved				PROTOCOL IDENTIFIER (Ah)			
3	Reserved							

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SPF bit, the PAGE CODE field, and the PAGE LENGTH field are defined in SPC-4 and are set to the values shown in table 107.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 107 indicating that this is a SOP specific mode page.

6.7.4 Protocol-Specific Port mode page

The Protocol-Specific Port mode page (see SPC-4) contains parameters that affect SOP target port operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT or MODE SENSE commands.

The mode page policy (see SPC-4) for this mode page shall be either shared or per target port. If a SOP target device has multiple SOP target ports, then the mode page policy should be per target port.

Parameters in this mode page:

a) shall affect only the SOP target port if the mode page policy is per target port; or
b) shall affect all SOP target ports in the SOP target device if the mode page policy is shared.

Table 108 defines the format of this mode page for SOP.

Table 108 — Protocol-Specific Port mode page for SOP

Byte \ Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (02h)							
2	Reserved				PROTOCOL IDENTIFIER (Ah)			
3	Reserved							

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SPF bit, the PAGE CODE field, and the PAGE LENGTH field are defined in SPC-4 and are set to the values shown in table 108.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 108 indicating that this is a SOP specific mode page.

6.8 SCSI vital product data (VPD)

6.8.1 SCSI vital product data (VPD) overview

Table 109 lists VPD pages for which this standard defines special requirements.

Table 109 — Device Identification VPD page designation descriptors for an SOP target port

Page code	VPD page name	Reference	Support
83h	Device Identification VPD page	6.8.2 and SPC-4	Mandatory

6.8.2 Device Identification VPD page

In the Device Identification VPD page (83h) returned in response to an INQUIRY command (see SPC-4), each logical unit in a SOP target device shall include the designation descriptors for the target port name (see 4.2.6), the target port identifier (see 4.2.7), and the relative target port identifier (see SAM-5 and SPC-4) listed in table 110.

Table 110 — Device Identification VPD page designation descriptors for an SOP target port

Field in designation descriptor	Designation descriptor		
	Target port name	**Target port identifier**	**Relative target port identifier**
DESIGNATOR TYPE	3h (i.e., NAA)	9h (i.e., protocol specific port identifier)	4h (i.e., relative target port identifier)
ASSOCIATION	01b (i.e., SCSI target port)		
CODE SET	1h (i.e., binary)		
DESIGNATOR LENGTH	8	8	4
PIV (protocol identifier valid)	1		
PROTOCOL IDENTIFIER	Ah (i.e., SOP)		
DESIGNATOR	NAA IEEE Registered format identifier (see 4.2.3) or NAA Locally Assigned format identifier (see 4.2.4)	Routing ID [a] (see 4.2.8)	Relative port identifier [b] as described in SAM-4 and SPC-4

[a] The DESIGNATOR field contains the routing ID of the SOP target port through which the INQUIRY command was received.
[b] The DESIGNATOR field contains the relative port identifier of the SOP target port through which the INQUIRY command was received.

NOTE 30 - If the PCI Express Routing ID changes (e.g., because the bus number changes), then each device server generates a unit attention condition with an additional sense code set to INQUIRY DATA HAS CHANGED (see SPC-4).

In the Device Identification VPD page (83h) returned by the INQUIRY command (see SPC-4), each logical unit in an SOP target device shall include a designation descriptor for the SOP target device name (see 4.2.5) using NAA format and may include a designation descriptor for the SOP target device name using the SCSI name string format as listed in table 111.

Table 111 — Device Identification VPD page designation descriptors for an SOP target device

Field in designation descriptor	Designation descriptor	
	NAA format (required)	SCSI name string format (optional)
DESIGNATOR TYPE	3h (i.e., NAA)	8h (i.e., SCSI name string)
ASSOCIATION	10b (i.e., SCSI target device)	
CODE SET	1h (i.e., binary)	3h (i.e., UTF-8)
DESIGNATOR LENGTH	8	24
PIV (protocol identifier valid)	1	0
PROTOCOL IDENTIFIER	Ah (i.e., SOP)	0h [a]
DESIGNATOR	Device name of the SOP target device in NAA format (see 4.2.5)	Device name of the SOP target device in SCSI name string format (e.g., “naa.” followed by 16 hexadecimal digits followed by 4 ASCII null characters)

[a] The PROTOCOL IDENTIFIER field is reserved if the PIV bit is set to zero.

Logical units may include designation descriptors in addition to those required by this standard (e.g., SCSI target devices with SCSI target ports using other SCSI transport protocols may return additional SCSI target device names for those other SCSI transport protocols).

Annex A
(Normative)

Mapping SOP over PQI

A.1 Inbound queue and outbound queue terminology

Table A.1 lists the mapping of SOP terminology to PQI terminology related to queues.

Table A.1 — Queue terminology

SOP term	PQI term
Inbound queue	Operational inbound queue (IQ)
Outbound queue	Operational outbound queue (OQ)

A.2 SOP classes for PQI

A.2.1 Host Queuing Layer class for PQI

A.2.1.1 Add To Queue operation

The SOP port implements the Add To Queue (IUs, Number Of IUs) operation (see 4.1.2.8.3.3) by splitting the IUs into one or more elements and invoking the PQI host IQ's Enqueue (Elements, Number Of Elements) operation (see PQI).

A.2.1.2 Remove From Queue operation

The SOP port implements the IUs = Remove From Queue (Number Of IUs) operation (see 4.1.2.8.3.4) by invoking the PQI host IQ's Dequeue (Elements, Number Of Elements) operation (see PQI) and joining elements into IUs based on the IU LENGTH field in the IU header.

A.2.2 Device Queuing Layer class for PQI

A.2.2.1 Add To Queue operation

The SOP port implements the Add To Queue (IUs, Number Of IUs) operation (see 4.1.2.8.4.3) by splitting the IUs into one or more elements and invoking the PQI device OQ's Enqueue (Elements, Number Of Elements) operation (see PQI).

After adding to an OQ, the PQI device is capable of asserting an interrupt as defined in PQI.

A.2.2.2 Remove From Queue operation

The SOP port implements the IUs = Remove From Queue (Number Of IUs) operation (see 4.1.2.8.4.4) by invoking the PQI device OQ's Dequeue (Elements, Number Of Elements) operation (see PQI) and joining elements into IUs based on the IU LENGTH field in the IU header.

A.3 Support requirements for the PQI SGL types for SOP IUs

Table A.2 defines the support requirement for the PQI SGL types for SOP IUs.

Table A.2 — Support requirements for the PQI SGL types for SOP IUs

SGL Type	Description	Support
0h	SGL Data Block descriptor	M
1h	SGL Bit Bucket descriptor	M
2h	SGL Standard Segment descriptor	M
3h	SGL Standard Last Segment descriptor	M
4h	SGL Alternative Last Segment descriptor	O
5h to Eh	Reserved	
Fh	Vendor specific	O
Key: M = Support is mandatory. O = Support is optional.		

REPORT GENERAL parameter data (see 5.2.4.1.2) may be used to return the supported optional SGL descriptor types.

A.4 PQI specific SOP IU fields

A.4.1 QUEUING INTERFACE SPECIFIC EVENT QUEUE field

Table A.3 defines the QUEUING INTERFACE SPECIFIC EVENT QUEUE field for the REPORT EVENT CONFIGURATION IU and the SET EVENT CONFIGURATION IU.

Table A.3 — QUEUING INTERFACE SPECIFIC EVENT QUEUE field

Byte \ Bit	7	6	5	4	3	2	1	0
0	RESPONSE QUEUE ID							(LSB)
1	(MSB)							

The RESPONSE QUEUE ID field specifies the OQ ID of the operational OQ in which the EVENT IU is to be delivered.

A.4.2 QUEUING INTERFACE SPECIFIC field

A.4.2.1 Queuing interface specific for request IUs

Table A.4 shows the QUEUING INTERFACE SPECIFIC field definition for a request IU (see table 31).

Table A.4 — QUEUING INTERFACE SPECIFIC field definition for a request IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	RESPONSE QUEUE ID							(LSB)
1	(MSB)							
2	WORK AREA							
3								

The RESPONSE QUEUE ID field specifies the OQ ID of the OQ where the response IU is to be delivered.

The WORK AREA field shall be ignored by the recipient.

A.4.2.2 Queuing interface specific for response IUs

Table A.5 shows the QUEUING INTERFACE SPECIFIC field definition for a response IU (see table 31).

Table A.5 — QUEUING INTERFACE SPECIFIC field definition for a response IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	WORK AREA							
3								

The WORK AREA field shall be ignored by the recipient.

A.4.3 Data-In Buffer descriptor area

The Data-In Buffer descriptor area in a request IU contains the first standard SGL segment of an SGL (see PQI) describing the Data-In Buffer as a destination data buffer.

If the request IU contains a PARTIAL IN bit and the PARTIAL IN bit is set to zero, then the standard SGL segment in the request IU is the last SGL segment.

A.4.4 Data-Out Buffer descriptor area

The Data-Out Buffer descriptor area in a request IU contains the first standard SGL segment of an SGL (see PQI) describing the Data-Out Buffer as a source data buffer.

If the request IU contains a PARTIAL OUT bit and the PARTIAL OUT bit is set to zero, then the standard SGL segment in the request IU is the last SGL segment.

A.4.5 Data Buffer descriptor area

The Data Buffer descriptor area in a request IU contains the first standard SGL segment of an SGL (see PQI) describing:

a) for a read command, the Data-In Buffer as a destination data buffer; or
b) for a write command, the Data-Out Buffer as a source data buffer.

If the request IU contains a PARTIAL bit and the PARTIAL bit is set to zero, then the standard SGL segment in the request IU is the last SGL segment.

A.4.6 DATA-IN BUFFER DESCRIPTOR AREA SIZE field

The DATA-IN BUFFER DESCRIPTOR AREA SIZE field in the EXTENDED COMMAND IU (see 5.2.5.3) specifies the number of SGL descriptors in the Data-In Buffer descriptor area contained within the IU.

If $((k \times 16) + 56 + m) > (4 + c)$, where:

k = the value of the DATA-IN BUFFER DESCRIPTOR AREA SIZE field;
m = the value of the ADDITIONAL CDB LENGTH field; and
c = the value of the IU LENGTH field,

then the SOP device shall transfer no data and return a COMMAND RESPONSE IU with the RESPONSE CODE field set to INVALID FIELD IN INFORMATION UNIT.

A.5 Queuing layer reset events

A PQI firm reset and a PQI hard reset (see PQI) cause a reset event (see SAM-5).

A.6 Queuing layer I_T nexus loss events

The following queuing layer event is considered to be an I_T nexus loss event (see SAM-5):

a) A PQI soft reset (see PQI); and
b) deletion of all outbound queues.

A.7 REPORT GENERAL queuing layer specific parameter data for PQI

For PQI, the QUEUING LAYER SPECIFIC DATA LENGTH field shall be set to 0004h in the REPORT GENERAL parameter data (see 5.2.4.1.2).

Table A.6 defines the format of the queuing layer specific data in the REPORT GENERAL parameter data (see 5.2.4.1.2).

Table A.6 — PQI QUEUING INTERFACE SPECIFIC field definition for a response IU

Byte \ Bit	7	6	5	4	3	2	1	0
0	INCOMING SGL TYPE SUPPORT BITMASK							
1								
2	Reserved							
3								

The INCOMING SGL DESCRIPTOR TYPE SUPPORT BITMASK field indicates the SGL types supported by the SOP target device. A bit set to one indicates that the corresponding SGL descriptor type is supported. A bit set to zero indicates that the corresponding SGL descriptor type is not supported. The first bit (i.e., byte 0 bit 0) corresponds to an SGL type of 0h (i.e, the SGL Data Block descriptor); the last bit (i.e., byte 1 bit 7) corresponds to an SGL type of Fh. The application client should only use mandatory SGL descriptor types until it retrieves the REPORT GENERAL parameter data (e.g., to send the REPORT GENERAL function itself).

A.8 Queuing-layer specific inbound queue error information for PQI

Table A.7 defines the queuing-layer-specific inbound queue error information for PQI.

Table A.7 — Queuing-layer-specific inbound queue error information for PQI

Byte \ Bit	7	6	5	4	3	2	1	0
0	BAD IU IQ ID							(LSB)
1	(MSB)							
2	Reserved							
3								
4	BAD IU IQ ELEMENT INDEX							(LSB)
•••								
6	(MSB)							
7	Reserved							

The BAD IU IQ ID field indicates the operational IQ in which the bad IU was received.

The BAD IU IQ ELEMENT INDEX field indicates the element index in the operational IQ in which the bad IU was received.

NOTE 31 - The SOP target port may have consumed the elements containing the bad IU and the SOP initiator port may have produced a new IU over those elements by the time that the event is returned.

Annex B
(Normative)

Protocol Specific requirements for bridges

B.1 PROTOCOL SPECIFIC field

B.1.1 Overview

The PROTOCOL SPECIFIC field is processed in certain request IUs by a bridge device server and specifies protocol specific behaviors. If the IU is not processed by a bridge device server then this field is reserved. The PROTOCOL SPECIFIC field is dependent upon the protocol of the local port as defined in table B.1.

Table B.1 — PROTOCOL SPECIFIC field

Protocol of the local port	Description
FCP	B.1.2
SAS SSP	B.1.3
All others	Reserved

B.1.2 FCP

Table B.2 defines the PROTOCOL SPECIFIC field for FCP.

Table B.2 — PROTOCOL SPECIFIC field for FCP

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="8">COMMAND REFERENCE NUMBER</td></tr>
<tr><td>1</td><td colspan="6">Reserved</td><td>ERROR RECOVERY</td><td>CLASS2</td></tr>
</table>

The COMMAND REFERENCE NUMBER field is defined in FCP-4.

NOTE 32 - The FCP-4 command reference number is used as the Send SCSI Command() CRN argument.

A CLASS2 bit set to zero specifies that the command shall be processed as an unacknowledged class (e.g., Class 3). A CLASS2 bit set to one specifies that the command shall be processed as an acknowledged class (e.g., Class 2).

An ERROR RECOVERY bit set to zero specifies that the FCP negotiated error recovery level shall be used. An ERROR RECOVERY bit set to one specifies that exchange level error recovery mechanism shall be used.

B.1.3 SAS SSP

Table B.3 defines the PROTOCOL SPECIFIC field for SAS SSP.

Table B.3 — PROTOCOL SPECIFIC field for SAS SSP

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved			TLR CONTROL		Reserved		
1	Reserved							

The TLR CONTROL field is defined in SPL-2.

B.2 Request identifiers

B.2.1 Request identifier mapping

The bridge device server, bridge task manager, and bridge application client shall map request identifiers used by the SOP port to command identifiers and task management associations used by the local port as defined in table B.4.

Table B.4 — Request identifier mapping

Local port protocol	field defined by the local port's SCSI transport protocol	Size	Scope	Reference
FCP	OX_ID field in the frame headers	16 bits	I_T nexus between the local port and the remote port	FCP-4
SRP	TAG field in the IUs	64 bits	I_T nexus between the local port and the remote port	SRP
iSCSI	Initiator Task Tag in the iSCSI PDUs	32 bits	I_T nexus between the local port and the remote port	iSCSI
SAS SSP	INITIATOR PORT TRANSFER TAG field in the SSP frame header	16 bits	I_T nexus between the local port and the remote port	SPL-2
SAS STP	If the local port is a SAS port that supports STP, then the bridge shall include a SCSI/ATA Translation Layer (see SAT-3) and translate SCSI commands into ATA commands, and send the ATA commands in STP connections			SPL-2 and SAT-3
SATA	If the local port is a SAS port that supports SATA or STP, then the bridge shall include a SCSI/ATA Translation Layer (see SAT-3) and translate SCSI commands into ATA commands, and send the ATA commands in SATA			SATA-3 and SAT-3
UAS	TAG field in the IUs	16 bits	I_T nexus between the local port and the remote port	UAS-2
SOP	REQUEST IDENTIFIER field in SCSI command and task management IUs	16 bits	I_T nexus between the local port and the remote port	5.2.5 and 5.2.5.4

B.2.2 Protocol specific bridge forwarding examples

Table B.5 shows an example for a bridge forwarding to a local port that is an FCP port.

Table B.5 — Forwarding by a bridge to a local port that is an FCP port

SOP target port	Dir	Local FCP port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit FCP_CMND IU
Receive COMMAND IU	⇒	
Receive EXTENDED COMMAND IU	⇒	
Transmit SUCCESS IU	⇐	Receive FCP_RSP IU with GOOD status
Transmit COMMAND RESPONSE IU	⇐	Receive FCP_RSP IU without GOOD status
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive FCP_XFER_RDY IU(s)
Receive PCI memory read completion(s)	⇒	Transmit FCP_DATA IU(s)
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Receive FCP_DATA IU(s)
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit FCP_CMND IU with the TASK MANAGEMENT FLAGS field set to a a value other than 00h (i.e., none)
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive FCP_RSP IU for a task management function

Table B.6 shows an example for a bridge forwarding to a local port that is an SRP port.

Table B.6 — Forwarding by a bridge to a local port that is an SRP port

SOP target port	Dir	Local SRP port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit SRP_CMND IU
Receive COMMAND IU	⇒	
Receive EXTENDED COMMAND IU	⇒	
Transmit SUCCESS IU	⇐	Receive SRP_RSP IU with GOOD status
Transmit COMMAND RESPONSE IU	⇐	Receive SRP_RSP IU without GOOD status
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive RDMA Read request(s)
Receive PCI memory read completion(s)	⇒	Transmit RDMA Read completion(s)
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Receive RDMA Write request(s)
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit SRP_TSK_MGMT IU
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive SRP_RSP IU for a task management function

Table B.7 shows an example for a bridge forwarding to a local port that is an iSCSI port.

Table B.7 — Forwarding by a bridge to a local port that is an iSCSI port

SOP target port	Dir	Local iSCSI port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit SCSI Command PDU
Receive COMMAND IU	⇒	
Receive EXTENDED COMMAND IU	⇒	
Transmit SUCCESS IU	⇐	Receive SCSI Response PDU with GOOD status
Transmit COMMAND RESPONSE IU	⇐	Receive SCSI Response PDU without GOOD status
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive Ready To Transfer PDU(s)
Receive PCI memory read completion(s)	⇒	Transmit SCSI Data-Out PDU(s)
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Receive SCSI Data-In PDU(s)
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit Task Management Function Request PDU
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive Task Management Function Response PDU
Note - PDU = protocol data unit		

Table B.8 shows an example for a bridge forwarding to a local port that is a SAS SSP port.

Table B.8 — Forwarding by a bridge to a local port that is an SAS SSP port

SOP target port	Dir	Local SAS SSP port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit COMMAND frame
Receive COMMAND IU	⇒	
Receive EXTENDED COMMAND IU	⇒	
Transmit SUCCESS IU	⇐	Receive RESPONSE frame with GOOD status
Transmit COMMAND RESPONSE IU	⇐	Receive RESPONSE frame without GOOD status
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive XFER_RDY frame(s)
Receive PCI memory read completion(s)	⇒	Transmit write DATA frame(s)
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Receive read DATA frame(s)
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit TASK frame
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive RESPONSE frame for a task management function

Table B.9 shows an example for a bridge forwarding to a local port that is a UAS USB-2 port.

Table B.9 — Forwarding by a bridge to a local port that is a UAS USB-2 port

SOP target port	Dir	Local UAS USB-2 port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit COMMAND IU
Receive COMMAND IU	⇒	
Receive EXTENDED COMMAND IU	⇒	
Transmit SUCCESS IU	⇐	Receive RESPONSE IU with GOOD status
Transmit COMMAND RESPONSE IU	⇐	Receive RESPONSE IU without GOOD status
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive WRITE READY IU(s)
Receive PCI memory read completion(s)	⇒	Transmit data on the Data-out pipe
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Receive READ READY IU(s) followed by data on the Data-in pipe
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit TASK MANAGEMENT IU
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive RESPONSE IU for a task management function

Table B.10 shows an example for a bridge forwarding to a local port that is a UAS USB-2 port.

Table B.10 — Forwarding by a bridge to a local port that is a UAS USB-3 port

SOP target port	Dir	Local UAS USB-3 port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit COMMAND IU
Receive COMMAND IU	⇒	
Receive EXTENDED COMMAND IU	⇒	
Transmit SUCCESS IU	⇐	Receive RESPONSE IU with GOOD status
Transmit COMMAND RESPONSE IU	⇐	Receive RESPONSE IU without GOOD status
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive ERDY on Data-out pipe
Receive PCI memory read completion(s)	⇒	Transmit data on the Data-out pipe
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Transmit ERDY on the Data-in pipe followed by Receive data on the Data-in pipe
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit TASK MANAGEMENT IU
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive RESPONSE IU for a task management function

Table B.11 shows an example for a bridge forwarding to a local port that is an SOP port.

Table B.11 — Forwarding by a bridge to a local port that is an SOP port

SOP target port	Dir	Local SOP port
Command and status		
Receive LIMITED COMMAND IU	⇒	Transmit LIMITED COMMAND IU
Receive COMMAND IU	⇒	Transmit COMMAND IU
Receive EXTENDED COMMAND IU	⇒	Transmit EXTENDED COMMAND IU
Transmit SUCCESS IU	⇐	Receive SUCCESS IU
Transmit COMMAND RESPONSE IU	⇐	Receive COMMAND RESPONSE IU
Data-out transfer		
Transmit PCI memory read request(s)	⇐	Receive PCI memory read request(s)
Receive PCI memory read completion(s)	⇒	Transmit PCI memory read completion(s)
Data-in transfer		
Transmit PCI memory write request(s)	⇐	Receive PCI memory write request(s)
Task management		
Receive TASK MANAGEMENT IU	⇒	Transmit TASK MANAGEMENT IU
Transmit TASK MANAGEMENT RESPONSE IU	⇐	Receive TASK MANAGEMENT RESPONSE IU

B.3 SCSI transport protocol specific data

B.3.1 Bridge port descriptors

B.3.1.1 Bridge port descriptors overview

Bridge port descriptor are used to identify:

a) local port(s) (see 5.2.6.2.1); and
b) remote port(s) accessed through the local port(s) (see 5.2.6.3.1).

Table B.12 defines the bridge port descriptors.

Table B.12 — Bridge port descriptors

Protocol identifier	Description
0h (i.e., Fibre Channel Protocol for SCSI (FCP))	B.3.1.2
1h to 3h (See SPC-4)	Reserved
4h (i.e., SCSI RDMA Protocol (SRP))	B.3.1.3
5h (i.e., Internet SCSI (iSCSI))	B.3.1.4
6h (i.e., SAS Serial SCSI Protocol (SSP))	B.3.1.5
7h to 8h	Reserved
9h (i.e., USB Attached SCSI (UAS))	B.3.1.6
Ah (i.e., SCSI over PCIe architecture (SOP))	B.3.1.7
Bh to Fh	Reserved

B.3.1.2 FCP bridge port descriptor

Table B.13 defines the bridge port descriptor for FCP.

Table B.13 — Bridge port descriptor for FCP

Byte \ Bit	7	6	5	4	3	2	1	0
0 ••• 7	N_PORT_NAME							
8	Reserved							
9 ••• 11	N_PORT_ID							

The N_PORT_NAME field indicates the N_Port_Name defined by the port login (PLOGI) extended link service (see FC-FS-3).

The N_PORT_ID field indicates the port D_ID (see FC-FS-3) to be used to transport frames including PLOGI and FCP-4 related frames.

B.3.1.3 SRP bridge port identifier

Table B.14 defines the bridge port identifier for SRP.

Table B.14 — Bridge port descriptor for SRP

Bit / Byte	7	6	5	4	3	2	1	0
0	TARGET PORT IDENTIFIER							
•••								
15								

The TARGET PORT IDENTIFIER field indicates the SRP port identifier (see SRP).

B.3.1.4 iSCSI bridge port identifier

Table B.15 defines the bridge port identifier for iSCSI.

Table B.15 — Bridge port identifier for iSCSI (part 1 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved						IPV6 VALID	IPV4 VALID
1	Reserved							
•••								
3								
4	(MSB)							
•••	IPV4 ADDRESS							
7								(LSB)
8	(MSB)							
•••	IPV6 ADDRESS							
23								(LSB)
24	(MSB)							
25	PORT NUMBER							(LSB)
26	Reserved							
27	INTERNET PROTOCOL NUMBER							
28	ISCSI NAME LENGTH (m-31)							
29	ISCSI INITIATOR SESSION ID LENGTH (x-m-5)							
30	ISCSI TARGET PORTAL GROUP LENGTH (n-x-4)							
31	Reserved							
32	(MSB)							
•••	ISCSI NAME							
m-1								(LSB)
m	(MSB)							
•••	ISCSI INITIATOR PORT SEPARATOR (2C 692C 3078h)							
m+4								(LSB)
m+5	(MSB)							
•••	ISCSI INITIATOR SESSION ID							
x-1								(LSB)

Table B.15 — Bridge port identifier for iSCSI (part 2 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
x	(MSB)							
•••	ISCSI TARGET PORT SEPARATOR (2C 742C 3078h)							
x+4								(LSB)
x+5	(MSB)							
•••	ISCSI TARGET PORTAL GROUP							
n								(LSB)

If the IPV4 VALID bit is set to one, then the IPV4 ADDRESS field indicates the IPv4 address of the SCSI port (see iSCSI and RFC 791), if any. If the IPV4 VALID bit is set to zero, then the IPV4 ADDRESS field shall be ignored by the recipient.

If the IPV6 VALID bit is set to one, then the IPV6 ADDRESS field indicates the IPv6 address of the SCSI port (see iSCSI), if any. If the IPV6 VALID bit is set to zero, then the IPV6 ADDRESS field shall be ignored by the recipient. At least one of following bits shall be set to one:

a) the IPV4 VALID bit; or
b) the IPV6 VALID bit.

The PORT NUMBER field indicates the TCP port number of the SCSI port (see iSCSI and RFC 3720).

The INTERNET PROTOCOL NUMBER field indicates the Internet protocol number of the SCSI port (see iSCSI and RFC 3720).

The ISCSI NAME LENGTH field indicates the length of the ISCSI NAME field, including the byte(s) containing the ASCII null character.

The ISCSI INITIATOR SESSION ID LENGTH field indicates the length of the ISCSI INITIATOR SESSION ID field, including the byte(s) containing the ASCII null character. An ISCSI INITIATOR SESSION ID LENGTH field set to 00h indicates that the iscsi initiator session id field is not present.

The ISCSI TARGET PORTAL GROUP LENGTH field indicates the length of the ISCSI TARGET PORTAL GROUP field, including the byte(s) containing the ASCII null character. An ISCSI TARGET PORTAL GROUP LENGTH field set to 00h indicates that the target portal group field is not present.

The null-terminated, null-padded (see 4.8) ISCSI NAME field indicates the iSCSI name of an iSCSI initiator node (see RFC 3720). The first ISCSI NAME field byte containing an ASCII null character terminates the ISCSI NAME field.

NOTE 33 - The iSCSI name length does not exceed 223 bytes.

The ISCSI INITIATOR PORT SEPARATOR field is set to the value shown in table B.15.

The null-terminated, null-padded ISCSI INITIATOR SESSION ID field indicates the iSCSI initiator session identifier (see RFC 3720) in the form of ASCII characters that are the hexadecimal digits converted from the binary iSCSI initiator session identifier value. The first ISCSI INITIATOR SESSION ID field byte containing an ASCII null character terminates the ISCSI INITIATOR SESSION ID field. If the SCSI port is not a SCSI initiator port, then this field is not present.

The ISCSI TARGET PORT GROUP SEPARATOR field is set to the value shown in table B.15.

The null-terminated, null-padded ISCSI TARGET PORTAL GROUP field indicates the iSCSI target portal group (see RFC 3720) in the form of ASCII characters that are the hexadecimal digits converted from the binary iSCSI target portal group value. The first ISCSI TARGET PORTAL GROUP field byte containing an ASCII null character terminates the ISCSI TARGET PORTAL GROUP field. If the SCSI port is not a SCSI target port, then this field is not present.

B.3.1.5 SAS SSP bridge port identifier

Table B.16 defines the bridge port identifier for SAS SSP.

Table B.16 — Bridge port descriptor for SAS SSP

Byte \ Bit	7	6	5	4	3	2	1	0
0 ••• 7	SAS ADDRESS							

The SAS ADDRESS field indicates the identifier assigned to the SAS port (see SPL-2).

B.3.1.6 UAS bridge port identifier

Table B.17 defines the bridge port identifier for UAS.

Table B.17 — Bridge port descriptor for UAS

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved	DEVICE ADDRESS						
1 ••• 3	Reserved							

The DEVICE ADDRESS field indicates the USB device address of the SCSI port (see UAS-2).

B.3.1.7 SOP bridge port identifier

Table B.18 defines the bridge port identifier for SOP.

Table B.18 — Bridge port descriptor for SOP

Byte \ Bit	7	6	5	4	3	2	1	0
0 – 1	ROUTING ID							
2 ••• 7	Reserved							

The ROUTING ID field indicates the routing ID of the SCSI port (see 4.2.8).

B.4 Data Buffer overflow mapping

Table B.19 lists the method that a local port uses to detect a Data-In Buffer overflow and/or a Data-Out Buffer overflow.

Table B.19 — Data-In Buffer overflow and Data-Out Buffer overflow detection

Protocol of the local port	Data-In Buffer overflow	Data-Out Buffer overflow
FCP	Receiving: a) FCP_CMND IU with RDDATA bit set to one and WRDATA bit set to zero; and b) FCP_RSP IU with FCP_RESID_OVER bit set to one; or: a) FCP_CMND IU with RDDATA bit set to one and WRDATA bit set to one; and b) FCP_RSP IU with FCP_BIDI_READ_RESID_OVER bit set to one.	Receiving: a) FCP_CMND IU with RDDATA bit set to zero and WRDATA bit set to one; and b) FCP_RSP IU with FCP_RESID_OVER bit set to one; or: a) FCP_CMND IU with RDDATA bit set to one and WRDATA bit set to one; and a) FCP_RSP IU with FCP_BIDI_READ_RESID_OVER bit set to one.
SRP	SRP_RSP with the DIOVER bit set to one	SRP_RSP with DOOVER bit set to one
iSCSI	Receiving: a) SCSI Command PDU with the Read (R) bit set to one and the Write (W) bit set to zero; and b) SCSI Response PDU with the Residual Overflow (O) bit set to one; a) or: b) SCSI Command PDU with the Read (R) bit set to one and the Write (W) bit set to one; and c) SCSI Response PDU with the Bidirectional Read Residual Overflow (o) bit set to one.	Receiving: a) SCSI Command PDU with the Read (R) bit set to zero and the Write (W) bit set to one; and b) SCSI Response PDU with the Residual Overflow (O) bit set to one; or: a) SCSI Command PDU with the Read (R) bit set to one and the Write (W) bit set to one; and b) SCSI Response PDU with the Residual Overflow (O) bit set to one.
SAS SSP	The local port detects overflows by counting bytes in incoming read DATA frames, which are contiguous.	The local port detects overflows by counting bytes requested in incoming XFER_RDY frames, which are contiguous.
UAS	The local port detects overflows by counting bytes in incoming Data-in pipe transfers, which are contiguous.	The local port detects overflows by counting bytes requested in incoming WRITE READY IUs, which are contiguous.
SOP	Receiving a COMMAND RESPONSE IU with the DATA-IN TRANSFER RESULT field set to: a) DATA-IN BUFFER OVERFLOW - DATA BUFFER SIZE; b) DATA-IN BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA; or c) DATA-IN BUFFER OVERFLOW - BRIDGE LOCAL PORT.	Receiving a COMMAND RESPONSE IU with the DATA-OUT TRANSFER RESULT field set to: a) DATA-OUT BUFFER OVERFLOW - DATA BUFFER SIZE; b) DATA-OUT BUFFER OVERFLOW - DATA BUFFER DESCRIPTOR AREA; or c) DATA-OUT BUFFER OVERFLOW - BRIDGE LOCAL PORT.

B.5 Additional response information

B.5.1 Additional response information overview

Table B.20 lists the additional response information (see 5.2.8.2.1) returned by a bridge in the COMMAND RESPONSE IU (see 5.2.8.2) and the TASK MANAGEMENT RESPONSE IU (see 5.2.8.3), which is based on the SCSI transport protocol of the local port.

Table B.20 — Additional response information returned by a bridge

Protocol of the local port	Description
SAS SSP	B.5.2
All others	Reserved

B.5.2 Additional response information for a SAS SSP local port

Table C.2 defines the ADDITIONAL RESPONSE INFORMATION field for a RESPONSE CODE field set to 80h (i.e., BRIDGE LOCAL PORT DETECTED ERROR RESPONSE) for a local port that is SAS SSP.

Table B.21 — bridge local port detected error responses for SAS SSP

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	SAS ERROR RESPONSE							

Table C.3 defines the SAS ERROR RESPONSE field.

Table B.22 — SAS ERROR RESPONSE field (part 1 of 2)

Code	Description
Generic error responses	
00h	Unspecified error response
01h	Unspecified SAS error response
02h	Unspecified SAS SSP error response
03h to 0Fh	Reserved
Error responses while sending a COMMAND frame or in response to a command	
10h	Unspecified error response to a COMMAND frame

[a] Reported after a vendor-specific number of retries have failed.

Table B.22 — SAS ERROR RESPONSE field (part 2 of 2)

Code	Description
11h	Received a NAK [a]
12h	Received a RESPONSE frame with the RESPONSE CODE field set to INVALID FRAME
13h	Received a RESPONSE frame with the RESPONSE CODE field set to OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED
14h	Received a RESPONSE frame with a reserved or invalid RESPONSE CODE field
15h	Received a RESPONSE frame with an invalid LENGTH field
16h to 1Fh	Reserved
Error responses while sending a TASK frame or in response to a task management function	
20h	Unspecified error response to a TASK frame
21h	Received a NAK [a]
22h	Received a RESPONSE frame with the RESPONSE CODE field set to INVALID FRAME
23h	Received a RESPONSE frame with the RESPONSE CODE field set to OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED
24h	Received a RESPONSE frame with a reserved or invalid RESPONSE CODE field
25h	Received a RESPONSE frame with an invalid LENGTH field
26h	Received a RESPONSE frame with the RESPONSE CODE field set to TASK MANAGEMENT FUNCTION FAILED
27h to 2Fh	Reserved
Other error responses	
30h to BFh	Reserved
C0h to FFh	Vendor specific error

[a] Reported after a vendor-specific number of retries have failed.

B.6 Event information

B.6.1 Event information overview

Table B.23 lists the event information (see 5.2.8.2.1) returned by a bridge in the EVENT IU (see 5.2.7.2) based on the SCSI transport protocol of the local port.

Table B.23 — Event information returned by a bridge

Protocol of the local port	Description
SAS SSP	B.5.2
All others	Reserved

B.6.2 Event information for a SAS SSP local port

Table B.24 defines the EVENT IDENTIFIER field for the EVENT TYPE field set to 80h (i.e., bridge local port detected error event) for a local port that is SAS SSP.

Table B.24 — Bridge local port detected error events for SAS SSP (part 1 of 2)

Code	Description
Generic bridge error events	
0000h	Unspecified error event
0001h	Unspecified SAS error event
0002h	Unspecified SAS SSP error event
0003h to 000Fh	Reserved
Bridge error events while sending a COMMAND frame or in response to a command	
0010h	Unspecified error event while sending a COMMAND frame
0011h	ACK/NAK timeout [a]
0012h	Connection failed [a]
0013h to 001Fh	Reserved
Bridge error events while sending a TASK frame	
0020h	Unspecified error event while sending a TASK frame
0021h	ACK/NAK timeout [a]
0022h	Connection failed [a]
0023h to 002Fh	Reserved
Bridge error events while receiving and processing an XFER_RDY frame	
0030h	Unspecified error event while receiving and processing an XFER_RDY frame
0031h	Incorrect write data length

[a] Reported after a vendor-specific number of retries have failed.
[b] Reported after a vendor-specific number of retries have failed, if transport layer retries are enabled.

Table B.24 — Bridge local port detected error events for SAS SSP (part 2 of 2)

Code	Description
0032h	Requested offset error
0033h	XFER_RDY frame was not expected
0034h to 003Fh	Reserved
Bridge error events while sending a write DATA frame	
0040h	Unspecified error event while sending a write DATA frame
0041h	ACK/NAK timeout [b]
0042h	Connection failed [b]
43h	Received a NAK [b]
0044h to 004Fh	Reserved
Bridge error events while receiving and processing a read DATA frame	
0050h	Unspecified error event while receiving and processing a read DATA frame
0051	Data offset error
0052	Too much read data
0053	Incorrect data length
0054	Read DATA frame was not expected
0055h to 005fh	Reserved
Other bridge error events	
00E0h to BFFFh	Reserved
C000h to FFFFh	Vendor specific

[a] Reported after a vendor-specific number of retries have failed.
[b] Reported after a vendor-specific number of retries have failed, if transport layer retries are enabled.

Table B.25 defines the ADDITIONAL EVENT IDENTIFIER field for the EVENT TYPE field set to 80h (i.e., bridge local port detected error event) for a local port that is SAS SSP.

Table B.25 — Bridge local port detected error events for SAS SSP

Byte \ Bit	7	6	5	4	3	2	1	0
0	REQUEST IDENTIFIER							
1								
2	Reserved							
3								

The REQUEST IDENTIFIER field indicates the request identifier of the SCSI command or task management function corresponding to the bridge error event. A REQUEST IDENTIFIER field set to 0000h indicates that the request identifier is not known.

Table C.7 defines event data for the EVENT TYPE field set to 80h (i.e., bridge local port detected error event) for a local port that is SAS SSP

Table B.26 — Event data for bridge local port detected error events for SAS SSP

Byte \ Bit	7	6	5	4	3	2	1	0
1	Vendor specific							
n								