



ISO/IEC 14776-415

Edition 1.0 2019-12

# INTERNATIONAL STANDARD



**Information technology – Small Computer System Interface (SCSI) –  
Part 415: SCSI Architecture Model - 5 (SAM-5)**



**Documents**

Document Sharing Hub



**THIS PUBLICATION IS COPYRIGHT PROTECTED**  
**Copyright © 2019 ISO/IEC, Geneva, Switzerland**

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about ISO/IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland

Tel.: +41 22 919 02 11  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)

**About the IEC**

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

**About IEC publications**

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

**IEC publications search - [webstore.iec.ch/advsearchform](http://webstore.iec.ch/advsearchform)**

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

**IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)**

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

**IEC Customer Service Centre - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)**

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: [sales@iec.ch](mailto:sales@iec.ch).

**Electropedia - [www.electropedia.org](http://www.electropedia.org)**

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

**IEC Glossary - [std.iec.ch/glossary](http://std.iec.ch/glossary)**

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.



ISO/IEC 14776-415

Edition 1.0 2019-12

# INTERNATIONAL STANDARD



---

**Information technology – Small Computer System Interface (SCSI) –  
Part 415: SCSI Architecture Model - 5 (SAM-5)**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

ICS 35.200

ISBN 978-2-8322-7662-4

**Warning! Make sure that you obtained this publication from an authorized distributor.**



**ISO/IEC  
14776-415**

---

**Information technology -  
Small Computer System Interface (SCSI) -  
Part 415: SCSI Architecture Model - 5 (SAM-5)**

---

Reference number  
ISO/IEC 14776-415:2019 © ISO/IEC 2019

Printed Wednesday, October 30, 2019 4:56 pm

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| 1 Scope .....                                                                                  | 15 |
| 2 Normative references .....                                                                   | 15 |
| 3 Terms, definitions, symbols, abbreviations, and conventions .....                            | 15 |
| 3.1 Terms and definitions .....                                                                | 15 |
| 3.2 Symbols and Abbreviations .....                                                            | 30 |
| 3.2.1 Abbreviations .....                                                                      | 30 |
| 3.2.2 Units .....                                                                              | 30 |
| 3.3 Keywords .....                                                                             | 30 |
| 3.4 Editorial conventions .....                                                                | 32 |
| 3.5 Numeric and character conventions .....                                                    | 33 |
| 3.5.1 Numeric conventions .....                                                                | 33 |
| 3.5.2 Byte encoded character strings conventions .....                                         | 34 |
| 3.6 UML notation conventions .....                                                             | 34 |
| 3.6.1 Notation conventions overview .....                                                      | 34 |
| 3.6.2 Constraint and note conventions .....                                                    | 34 |
| 3.6.3 Class diagram conventions .....                                                          | 35 |
| 3.6.4 Object diagram conventions .....                                                         | 40 |
| 3.7 State machine conventions .....                                                            | 42 |
| 3.7.1 State machine conventions overview .....                                                 | 42 |
| 3.7.2 Transitions .....                                                                        | 43 |
| 3.7.3 Messages, requests, indications, confirmations, responses, and event notifications ..... | 43 |
| 3.7.4 State machine counters, timers, and variables .....                                      | 44 |
| 3.8 Bit and byte ordering .....                                                                | 44 |
| 3.9 Notation for procedure calls .....                                                         | 46 |
| 4 SCSI architecture model .....                                                                | 47 |
| 4.1 Overview .....                                                                             | 47 |
| 4.2 Compliance requirements .....                                                              | 47 |
| 4.3 The SCSI distributed service model .....                                                   | 49 |
| 4.4 The SCSI client-server model .....                                                         | 50 |
| 4.4.1 SCSI client-server model overview .....                                                  | 50 |
| 4.4.2 Synchronizing client and server changes .....                                            | 51 |
| 4.4.3 Server request/response ordering .....                                                   | 51 |
| 4.5 The SCSI structural model .....                                                            | 52 |
| 4.6 SCSI classes .....                                                                         | 53 |
| 4.6.1 SCSI classes overview .....                                                              | 53 |
| 4.6.2 SCSI Domain class .....                                                                  | 54 |
| 4.6.3 Service Delivery Subsystem class .....                                                   | 55 |
| 4.6.4 SCSI Device class .....                                                                  | 55 |
| 4.6.4.1 SCSI Device class overview .....                                                       | 55 |
| 4.6.4.2 SCSI Device Name attribute .....                                                       | 56 |
| 4.6.5 SCSI Port class .....                                                                    | 57 |
| 4.6.5.1 SCSI Port class overview .....                                                         | 57 |
| 4.6.5.2 Relative Port Identifier attribute .....                                               | 58 |
| 4.6.6 SCSI Target Port class .....                                                             | 58 |
| 4.6.6.1 SCSI Target Port class overview .....                                                  | 58 |
| 4.6.6.2 Target Port Identifier attribute .....                                                 | 58 |
| 4.6.6.3 Target Port Name attribute .....                                                       | 59 |
| 4.6.6.4 Send Data-In operation .....                                                           | 59 |
| 4.6.6.5 Receive Data-Out operation .....                                                       | 59 |
| 4.6.6.6 Terminate Data Transfer operation .....                                                | 59 |
| 4.6.6.7 Send Command Complete operation .....                                                  | 59 |
| 4.6.6.8 Task Management Function Executed operation .....                                      | 59 |
| 4.6.7 Task Router class .....                                                                  | 59 |
| 4.6.7.1 Task Router class overview .....                                                       | 59 |
| 4.6.7.2 Route Command operation .....                                                          | 60 |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 4.6.7.3 Route Task Management Function operation .....                        | 60 |
| 4.6.7.4 Reroute Conglomerate Command operation .....                          | 61 |
| 4.6.7.5 Reroute Conglomerate Task Management Functions operation.....         | 61 |
| 4.6.7.6 Stop Conglomerate Task Management Functions Rerouting operation ..... | 62 |
| 4.6.8 SCSI Initiator Port class .....                                         | 62 |
| 4.6.8.1 SCSI Initiator Port class overview .....                              | 62 |
| 4.6.8.2 Initiator Port Identifier attribute .....                             | 63 |
| 4.6.8.3 Initiator Port Name attribute .....                                   | 63 |
| 4.6.8.4 Send SCSI Command operation .....                                     | 63 |
| 4.6.8.5 Send Task Management Request operation.....                           | 63 |
| 4.6.8.6 Get Initiator Port Identifier operation .....                         | 63 |
| 4.6.8.7 Get Initiator Port Name operation.....                                | 63 |
| 4.6.9 SCSI Target Device class .....                                          | 64 |
| 4.6.10 Logical Unit Conglomerate class .....                                  | 65 |
| 4.6.11 Level 1 Hierarchical Logical Unit class .....                          | 66 |
| 4.6.12 Level 2 Hierarchical Logical Unit class .....                          | 67 |
| 4.6.13 Level 3 Hierarchical Logical Unit class .....                          | 68 |
| 4.6.14 Level 4 Hierarchical Logical Unit class .....                          | 68 |
| 4.6.15 Hierarchical Logical Unit class .....                                  | 68 |
| 4.6.16 Management Logical Unit class .....                                    | 68 |
| 4.6.17 Well Known Logical Unit class .....                                    | 69 |
| 4.6.18 Logical Unit class .....                                               | 69 |
| 4.6.18.1 Logical Unit class overview .....                                    | 69 |
| 4.6.18.2 LUN attribute .....                                                  | 71 |
| 4.6.18.3 Logical Unit Name attribute .....                                    | 71 |
| 4.6.18.4 Dependent Logical Unit attribute .....                               | 72 |
| 4.6.18.5 Device Type attribute .....                                          | 72 |
| 4.6.19 Device Server class .....                                              | 72 |
| 4.6.19.1 Device Server class overview .....                                   | 72 |
| 4.6.19.2 Data-In Delivered operation .....                                    | 72 |
| 4.6.19.3 Data-Out Received operation.....                                     | 72 |
| 4.6.19.4 Data Transfer Terminated operation .....                             | 73 |
| 4.6.20 Copy Manager class .....                                               | 73 |
| 4.6.20.1 Copy Manager class overview .....                                    | 73 |
| 4.6.20.2 Create ROD Token operation.....                                      | 73 |
| 4.6.20.3 Process ROD Token operation .....                                    | 73 |
| 4.6.21 Task Manager class.....                                                | 73 |
| 4.6.21.1 Task Manager class overview .....                                    | 73 |
| 4.6.21.2 SCSI Command Received operation .....                                | 73 |
| 4.6.21.3 Task Management Request Received operation .....                     | 74 |
| 4.6.21.4 Data Transfer Terminated operation .....                             | 74 |
| 4.6.21.5 Nexus Loss operation.....                                            | 74 |
| 4.6.21.6 Transport Reset operation.....                                       | 74 |
| 4.6.21.7 Power Loss Expected operation.....                                   | 74 |
| 4.6.22 Task Set class.....                                                    | 74 |
| 4.6.23 Command class .....                                                    | 74 |
| 4.6.23.1 Command class overview .....                                         | 74 |
| 4.6.23.2 I_T_L Nexus attribute .....                                          | 74 |
| 4.6.23.3 Command Identifier attribute .....                                   | 75 |
| 4.6.23.4 Task Attribute attribute .....                                       | 75 |
| 4.6.23.5 CDB attribute.....                                                   | 75 |
| 4.6.23.6 CRN attribute.....                                                   | 75 |
| 4.6.23.7 Command Priority attribute.....                                      | 75 |
| 4.6.23.8 Status attribute .....                                               | 75 |
| 4.6.23.9 Sense Data attribute.....                                            | 75 |
| 4.6.23.10 Sense Data Length attribute.....                                    | 75 |
| 4.6.23.11 Service Response attribute .....                                    | 75 |
| 4.6.23.12 Status Qualifier attribute.....                                     | 75 |

|                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------|----|
| 4.6.23.13 First Burst Enabled attribute .....                                                             | 75 |
| 4.6.23.14 Device Server Buffer attribute .....                                                            | 75 |
| 4.6.23.15 Application Client Buffer Offset attribute .....                                                | 76 |
| 4.6.23.16 Request Byte Count attribute .....                                                              | 76 |
| 4.6.23.17 Delivery Result attribute .....                                                                 | 76 |
| 4.6.24 Task Management Function class .....                                                               | 76 |
| 4.6.24.1 Task Management Function class overview .....                                                    | 76 |
| 4.6.24.2 Function Identifier attribute .....                                                              | 76 |
| 4.6.24.3 I_T Nexus attribute .....                                                                        | 76 |
| 4.6.24.4 I_T_L Nexus attribute .....                                                                      | 76 |
| 4.6.24.5 Command Identifier attribute .....                                                               | 76 |
| 4.6.24.6 Service Response attribute .....                                                                 | 76 |
| 4.6.24.7 Additional Response Information attribute .....                                                  | 77 |
| 4.6.25 Administrative Logical Unit class .....                                                            | 77 |
| 4.6.25.1 Administrative Logical Unit class overview .....                                                 | 77 |
| 4.6.25.2 Rerouted commands and task management functions .....                                            | 77 |
| 4.6.25.2.1 Overview .....                                                                                 | 77 |
| 4.6.25.2.2 Commands rerouted due to incorrect logical unit selection .....                                | 77 |
| 4.6.25.2.3 Command rerouting in response to an administrative logical unit request .....                  | 77 |
| 4.6.25.2.4 Task management function rerouting in response to an administrative logical unit request ..... | 78 |
| 4.6.26 Subsidiary Logical Unit class .....                                                                | 79 |
| 4.6.26.1 Subsidiary Logical Unit class overview .....                                                     | 79 |
| 4.6.26.2 Binding attribute .....                                                                          | 79 |
| 4.6.26.3 Host Bind Identifier attribute .....                                                             | 79 |
| 4.6.27 SCSI Initiator Device class .....                                                                  | 80 |
| 4.6.28 Application Client class .....                                                                     | 81 |
| 4.6.28.1 Application Client class overview .....                                                          | 81 |
| 4.6.28.2 Command Complete Received operation .....                                                        | 81 |
| 4.6.28.3 Received Task Management Function Executed operation .....                                       | 81 |
| 4.6.28.4 Nexus Loss operation .....                                                                       | 82 |
| 4.6.28.5 Transport Reset operation .....                                                                  | 82 |
| 4.6.29 Application Client Task Management Function class .....                                            | 82 |
| 4.6.29.1 Application Client Task Management Function class overview .....                                 | 82 |
| 4.6.29.2 Function Identifier attribute .....                                                              | 82 |
| 4.6.29.3 I_T Nexus attribute .....                                                                        | 82 |
| 4.6.29.4 I_T_L Nexus attribute .....                                                                      | 82 |
| 4.6.29.5 Command Identifier attribute .....                                                               | 82 |
| 4.6.29.6 Service Response attribute .....                                                                 | 82 |
| 4.6.29.7 Additional Response Information attribute .....                                                  | 82 |
| 4.6.30 Application Client Task Set class .....                                                            | 83 |
| 4.6.31 Application Client Command class .....                                                             | 83 |
| 4.6.31.1 Application Client Command class overview .....                                                  | 83 |
| 4.6.31.2 I_T_L Nexus attribute .....                                                                      | 83 |
| 4.6.31.3 Command Identifier attribute .....                                                               | 83 |
| 4.6.31.4 CDB attribute .....                                                                              | 84 |
| 4.6.31.5 Task Attribute attribute .....                                                                   | 84 |
| 4.6.31.6 Status attribute .....                                                                           | 84 |
| 4.6.31.7 Service Response attribute .....                                                                 | 84 |
| 4.6.31.8 Data-In Buffer attribute .....                                                                   | 84 |
| 4.6.31.9 Data-In Buffer Size attribute .....                                                              | 84 |
| 4.6.31.10 Data-Out Buffer attribute .....                                                                 | 85 |
| 4.6.31.11 Data-Out Buffer Size attribute .....                                                            | 85 |
| 4.6.31.12 CRN attribute .....                                                                             | 85 |
| 4.6.31.13 Command Priority attribute .....                                                                | 85 |
| 4.6.31.14 First Burst Enabled attribute .....                                                             | 85 |
| 4.6.31.15 Sense Data attribute .....                                                                      | 85 |
| 4.6.31.16 Sense Data Length attribute .....                                                               | 85 |
| 4.6.31.17 Status Qualifier attribute .....                                                                | 85 |

|                                                                                                  |     |
|--------------------------------------------------------------------------------------------------|-----|
| 4.6.32 Discovery class .....                                                                     | 85  |
| 4.6.32.1 Discovery class overview .....                                                          | 85  |
| 4.6.32.2 I_T Nexus attribute .....                                                               | 86  |
| 4.6.32.3 I_T_L Nexus attribute .....                                                             | 86  |
| 4.6.32.4 Return I_T Nexus operation .....                                                        | 86  |
| 4.6.32.5 Return I_T_L Nexus operation .....                                                      | 86  |
| 4.7 Logical unit number (LUN).....                                                               | 86  |
| 4.7.1 Overview .....                                                                             | 86  |
| 4.7.2 Logical unit representation format.....                                                    | 87  |
| 4.7.3 LUNs overview.....                                                                         | 87  |
| 4.7.4 Minimum LUN addressing requirements.....                                                   | 87  |
| 4.7.5 Single level LUN structure .....                                                           | 88  |
| 4.7.6 Complex LUN structures.....                                                                | 91  |
| 4.7.6.1 Complex LUN structures overview .....                                                    | 91  |
| 4.7.6.2 Logical unit conglomerate LUN structure .....                                            | 91  |
| 4.7.6.3 Hierarchical LUN structure .....                                                         | 92  |
| 4.7.7 Addressing methods .....                                                                   | 95  |
| 4.7.7.1 Simple logical unit addressing method.....                                               | 95  |
| 4.7.7.2 Peripheral device addressing method.....                                                 | 95  |
| 4.7.7.3 Flat space addressing method .....                                                       | 97  |
| 4.7.7.4 Logical unit addressing method.....                                                      | 98  |
| 4.7.7.5 Extended logical unit addressing.....                                                    | 99  |
| 4.7.7.5.1 Extended logical unit addressing formats .....                                         | 99  |
| 4.7.7.5.2 Well known logical unit addressing.....                                                | 102 |
| 4.7.7.5.3 Extended flat space addressing method.....                                             | 103 |
| 4.7.7.5.4 Long extended flat space addressing method .....                                       | 103 |
| 4.7.7.5.5 Logical unit not specified addressing.....                                             | 104 |
| 4.8 SCSI ports .....                                                                             | 104 |
| 4.8.1 SCSI port configurations .....                                                             | 104 |
| 4.8.2 SCSI devices with multiple SCSI ports .....                                                | 105 |
| 4.8.3 SCSI target device with multiple SCSI ports structure.....                                 | 106 |
| 4.8.4 SCSI initiator device with multiple SCSI initiator ports structure.....                    | 107 |
| 4.8.5 SCSI device with multiple SCSI ports structure .....                                       | 107 |
| 4.8.6 SCSI initiator device view of SCSI target device with multiple SCSI target ports .....     | 108 |
| 4.8.7 SCSI target device view of a SCSI initiator device with multiple SCSI initiator ports..... | 111 |
| 4.9 The SCSI model for distributed communications .....                                          | 112 |
| 5 SCSI command model .....                                                                       | 117 |
| 5.1 The Execute Command procedure call .....                                                     | 117 |
| 5.2 Command descriptor block (CDB).....                                                          | 118 |
| 5.3 Status .....                                                                                 | 120 |
| 5.3.1 Status codes .....                                                                         | 120 |
| 5.3.2 Status qualifier .....                                                                     | 121 |
| 5.3.3 Status precedence .....                                                                    | 124 |
| 5.4 SCSI transport protocol services for SCSI commands.....                                      | 124 |
| 5.4.1 SCSI transport protocol services for SCSI commands overview .....                          | 124 |
| 5.4.2 Command and status SCSI transport protocol services .....                                  | 125 |
| 5.4.2.1 Command and status SCSI transport protocol services overview .....                       | 125 |
| 5.4.2.2 Send SCSI Command SCSI transport protocol service request .....                          | 125 |
| 5.4.2.3 SCSI Command Received SCSI transport protocol service indication .....                   | 125 |
| 5.4.2.4 Send Command Complete SCSI transport protocol service response .....                     | 126 |
| 5.4.2.5 Command Complete Received SCSI transport protocol service confirmation.....              | 126 |
| 5.4.3 Data transfer SCSI transport protocol services.....                                        | 127 |
| 5.4.3.1 Overview .....                                                                           | 127 |
| 5.4.3.2 Data-In delivery service.....                                                            | 129 |
| 5.4.3.2.1 Send Data-In SCSI transport protocol service request.....                              | 129 |
| 5.4.3.2.2 Data-In Delivered SCSI transport protocol service confirmation .....                   | 129 |
| 5.4.3.3 Data-Out delivery service .....                                                          | 129 |

|                                                                                                                  |     |
|------------------------------------------------------------------------------------------------------------------|-----|
| 5.4.3.3.1 Receive Data-Out SCSI transport protocol service request .....                                         | 129 |
| 5.4.3.3.2 Data-Out Received SCSI transport protocol service confirmation .....                                   | 130 |
| 5.4.3.4 Terminate Data Transfer service .....                                                                    | 130 |
| 5.4.3.4.1 Terminate Data Transfer SCSI transport protocol service request .....                                  | 130 |
| 5.4.3.4.2 Data Transfer Terminated SCSI transport protocol service confirmation .....                            | 131 |
| 5.5 Command lifetime .....                                                                                       | 131 |
| 5.6 Aborting commands .....                                                                                      | 132 |
| 5.7 Command processing example .....                                                                             | 139 |
| 5.8 Commands that complete with CHECK CONDITION status .....                                                     | 140 |
| 5.8.1 Overview .....                                                                                             | 140 |
| 5.8.2 Handling commands when ACA is not in effect .....                                                          | 140 |
| 5.8.3 Aborting commands terminated with a CHECK CONDITION status without establishing an ACA .....               | 140 |
| 5.9 Auto contingent allegiance (ACA) .....                                                                       | 141 |
| 5.9.1 ACA overview .....                                                                                         | 141 |
| 5.9.2 Establishing an ACA .....                                                                                  | 141 |
| 5.9.3 Handling new commands received on the faulted I_T nexus when ACA is in effect .....                        | 142 |
| 5.9.4 Handling new commands received on non-faulted I_T nexuses when ACA is in effect .....                      | 142 |
| 5.9.4.1 Command processing that is permitted for commands received on a non-faulted I_T nexuses during ACA ..... | 142 |
| 5.9.4.2 Handling new commands received on non-faulted I_T nexuses when ACA is in effect .....                    | 142 |
| 5.9.5 Clearing an ACA condition .....                                                                            | 143 |
| 5.10 Overlapped commands .....                                                                                   | 144 |
| 5.11 Incorrect logical unit numbers for commands .....                                                           | 145 |
| 5.12 Task attribute exception conditions .....                                                                   | 145 |
| 5.13 Sense data .....                                                                                            | 146 |
| 5.13.1 Command terminated sense data or polled sense data .....                                                  | 146 |
| 5.13.2 Command completed sense data .....                                                                        | 146 |
| 5.14 Unit attention conditions .....                                                                             | 146 |
| 5.14.1 Unit attention conditions that are not coalesced .....                                                    | 146 |
| 5.14.2 Coalescing unit attention conditions .....                                                                | 149 |
| 6 SCSI events and event notification model .....                                                                 | 151 |
| 6.1 SCSI events overview .....                                                                                   | 151 |
| 6.2 Establishing a unit attention condition subsequent to detection of an event .....                            | 153 |
| 6.3 Conditions resulting from SCSI events .....                                                                  | 154 |
| 6.3.1 Power on .....                                                                                             | 154 |
| 6.3.2 Hard reset .....                                                                                           | 155 |
| 6.3.3 Logical unit reset .....                                                                                   | 155 |
| 6.3.4 I_T nexus loss .....                                                                                       | 156 |
| 6.3.5 Power loss expected .....                                                                                  | 156 |
| 6.4 SCSI transport protocol services for event notification .....                                                | 157 |
| 6.4.1 SCSI transport protocol service for event notification overview .....                                      | 157 |
| 6.4.2 Nexus Loss SCSI transport protocol service indication .....                                                | 157 |
| 6.4.3 Transport Reset SCSI transport protocol service indication .....                                           | 157 |
| 6.4.4 Power Loss Expected SCSI transport protocol service indication .....                                       | 157 |
| 7 Task management functions .....                                                                                | 159 |
| 7.1 Task management function procedure calls .....                                                               | 159 |
| 7.2 ABORT TASK .....                                                                                             | 160 |
| 7.3 ABORT TASK SET .....                                                                                         | 161 |
| 7.4 CLEAR ACA .....                                                                                              | 161 |
| 7.5 CLEAR TASK SET .....                                                                                         | 162 |
| 7.6 I_T NEXUS RESET .....                                                                                        | 162 |
| 7.7 LOGICAL UNIT RESET .....                                                                                     | 163 |
| 7.8 QUERY TASK .....                                                                                             | 163 |
| 7.9 QUERY TASK SET .....                                                                                         | 164 |
| 7.10 QUERY ASYNCHRONOUS EVENT .....                                                                              | 164 |
| 7.11 Task management function lifetime .....                                                                     | 166 |

|                                                                                                   |     |
|---------------------------------------------------------------------------------------------------|-----|
| 7.12 SCSI transport protocol services for task management functions .....                         | 166 |
| 7.12.1 SCSI transport protocol services for task management functions overview .....              | 166 |
| 7.12.2 Send Task Management Request SCSI transport protocol service request .....                 | 167 |
| 7.12.3 Task Management Request Received SCSI transport protocol service indication .....          | 167 |
| 7.12.4 Task Management Function Executed SCSI transport protocol service response .....           | 167 |
| 7.12.5 Received Task Management Function Executed SCSI transport protocol service confirmation .. | 168 |
| 7.13 Task management function example .....                                                       | 169 |
| 8 Task set management .....                                                                       | 171 |
| 8.1 Task set management overview .....                                                            | 171 |
| 8.2 Implicit head of queue .....                                                                  | 171 |
| 8.3 Command management model .....                                                                | 171 |
| 8.4 Task attributes .....                                                                         | 172 |
| 8.4.1 Overview .....                                                                              | 172 |
| 8.4.2 Commands having the SIMPLE task attribute .....                                             | 172 |
| 8.4.3 Commands having the ORDERED task attribute .....                                            | 172 |
| 8.4.4 Commands having the HEAD OF QUEUE task attribute .....                                      | 172 |
| 8.4.5 Commands having the ACA task attribute .....                                                | 172 |
| 8.5 Command priority .....                                                                        | 173 |
| 8.6 Command duration limit .....                                                                  | 173 |
| 8.6.1 Command duration limit overview .....                                                       | 173 |
| 8.6.2 Command duration scheduling .....                                                           | 174 |
| 8.7 LU (logical unit) state machines .....                                                        | 174 |
| 8.7.1 LU state machine overview .....                                                             | 174 |
| 8.7.2 LU_TM (task manager) state machine .....                                                    | 176 |
| 8.7.2.1 LU_TM state machine overview .....                                                        | 176 |
| 8.7.2.2 LU_TM command processing .....                                                            | 177 |
| 8.7.2.3 LU_TM task management function processing .....                                           | 179 |
| 8.7.2.4 LU_TM event processing .....                                                              | 180 |
| 8.7.2.5 LU_TM terminated command processing .....                                                 | 182 |
| 8.7.2.5.1 LU_TM ACA not established .....                                                         | 182 |
| 8.7.2.5.2 LU_TM ACA established .....                                                             | 183 |
| 8.7.3 LU_DS (device server) state machine .....                                                   | 185 |
| 8.7.3.1 LU_DS state machine overview .....                                                        | 185 |
| 8.7.3.2 LU_DS command processing .....                                                            | 185 |
| 8.7.3.3 LU_DS background processing .....                                                         | 188 |
| 8.7.4 LU_CS (command state) state machine .....                                                   | 189 |
| 8.7.4.1 LU_CS state machine overview .....                                                        | 189 |
| 8.7.4.2 LU_CS1:Idle state .....                                                                   | 189 |
| 8.7.4.2.1 LU_CS1:Idle state description .....                                                     | 189 |
| 8.7.4.2.2 Transition LU_CS1:Idle to LU_CS2:Dormant .....                                          | 189 |
| 8.7.4.2.3 Transition LU_CS1:Idle to LU_CS3:Enabled .....                                          | 189 |
| 8.7.4.3 LU_CS2:Dormant state .....                                                                | 190 |
| 8.7.4.3.1 LU_CS2:Dormant state description .....                                                  | 190 |
| 8.7.4.3.2 Transition LU_CS2:Dormant to LU_CS3:Enabled .....                                       | 190 |
| 8.7.4.3.3 Transition LU_CS2:Dormant to LU_CS1:Idle .....                                          | 190 |
| 8.7.4.4 LU_CS3:Enabled state .....                                                                | 190 |
| 8.7.4.4.1 LU_CS1:Enabled state description .....                                                  | 190 |
| 8.7.4.4.2 Transition LU_CS3:Enabled to LU_CS1:Idle .....                                          | 191 |
| 8.7.4.4.3 Transition LU_CS3:Enabled to LU_CS4:Blocked .....                                       | 191 |
| 8.7.4.4.4 Transition LU_CS3:Enabled to LU_CS5:Completed .....                                     | 191 |
| 8.7.4.5 LU_CS4:Blocked state .....                                                                | 191 |
| 8.7.4.5.1 LU_CS4:Blocked state description .....                                                  | 191 |
| 8.7.4.5.2 Transition LU_CS4:Blocked to LU_CS3:Enabled .....                                       | 191 |
| 8.7.4.5.3 Transition LU_CS4:Blocked to LU_CS5:Completed .....                                     | 192 |
| 8.7.4.6 LU_CS5:Completed state .....                                                              | 192 |
| 8.7.4.6.1 LU_CS5:Completed state description .....                                                | 192 |
| 8.7.4.6.2 Transition LU_CS5:Completed to LU_CS1:Idle .....                                        | 192 |

|                                                                                                                                    |     |
|------------------------------------------------------------------------------------------------------------------------------------|-----|
| 8.8 Task set management examples.....                                                                                              | 192 |
| 8.8.1 Overview .....                                                                                                               | 192 |
| 8.8.2 Commands having the HEAD OF QUEUE task attribute .....                                                                       | 193 |
| 8.8.3 Commands having the ORDERED task attribute .....                                                                             | 195 |
| 8.8.4 Commands having the ACA task attribute .....                                                                                 | 196 |
| Annex A (informative) Identifiers and names for objects.....                                                                       | 197 |
| A.1 Identifiers and names overview.....                                                                                            | 197 |
| A.2 Identifiers and names.....                                                                                                     | 198 |
| Annex B (informative) SCSI Initiator Port attributes and SCSI Target Port attributes supported by<br>SCSI transport protocols..... | 203 |
| Annex C (informative) Terminology mapping.....                                                                                     | 207 |
| C.1 Terminology mapping to SAM-3 .....                                                                                             | 207 |
| C.2 Terminology mapping to SAM-4 .....                                                                                             | 207 |
| Annex D (informative) SCSI transport protocol acronyms .....                                                                       | 208 |
| Bibliography .....                                                                                                                 | 209 |

|                                                                                                  |     |
|--------------------------------------------------------------------------------------------------|-----|
| Table 1 — Numbering conventions.....                                                             | 33  |
| Table 2 — Constraint and note notation .....                                                     | 34  |
| Table 3 — Class diagram notation for classes.....                                                | 35  |
| Table 4 — Multiplicity notation .....                                                            | 36  |
| Table 5 — Class diagram notation for associations.....                                           | 36  |
| Table 6 — Class diagram notation for aggregations.....                                           | 37  |
| Table 7 — Class diagram notation for generalizations .....                                       | 38  |
| Table 8 — Class diagram notation for dependency .....                                            | 39  |
| Table 9 — Object diagram notation for objects.....                                               | 40  |
| Table 10 — Object diagram notation for link.....                                                 | 40  |
| Table 11 — Example of ordering of bits and bytes within a data dword .....                       | 45  |
| Table 12 — Example of ordering of bits and bytes within a data dword element .....               | 45  |
| Table 13 — Command identifier scopes .....                                                       | 84  |
| Table 14 — Single level LUN structure using peripheral device addressing method .....            | 88  |
| Table 15 — Single level LUN structure using flat space addressing method .....                   | 88  |
| Table 16 — Single level LUN structure using extended flat space addressing method.....           | 89  |
| Table 17 — Single level LUN structure using long extended flat space addressing method.....      | 89  |
| Table 18 — Complex LUN structures .....                                                          | 91  |
| Table 19 — Logical unit conglomerate LUN structure .....                                         | 91  |
| Table 20 — Format of addressing fields in the logical unit conglomerate LUN structure .....      | 92  |
| Table 21 — ADDRESS METHOD field in the logical unit conglomerate LUN structure.....              | 92  |
| Table 22 — Hierarchical LUN structure adjustments .....                                          | 93  |
| Table 23 — Hierarchical LUN structure .....                                                      | 94  |
| Table 24 — Format of addressing fields in the hierarchical LUN structure.....                    | 94  |
| Table 25 — ADDRESS METHOD field in the hierarchical LUN structure .....                          | 95  |
| Table 26 — Simple logical unit addressing format.....                                            | 95  |
| Table 27 — Peripheral device addressing format.....                                              | 96  |
| Table 28 — Flat space addressing format.....                                                     | 97  |
| Table 29 — Logical unit addressing format.....                                                   | 98  |
| Table 30 — Extended logical unit addressing format.....                                          | 100 |
| Table 31 — LENGTH field and related sizes .....                                                  | 100 |
| Table 32 — Two byte extended logical unit addressing format .....                                | 101 |
| Table 33 — Four byte extended logical unit addressing format.....                                | 101 |
| Table 34 — Six byte extended logical unit addressing format.....                                 | 101 |
| Table 35 — Eight byte extended logical unit addressing format.....                               | 101 |
| Table 36 — Logical unit extended addressing .....                                                | 102 |
| Table 37 — Well known logical unit extended addressing format.....                               | 103 |
| Table 38 — Extended flat space addressing format .....                                           | 103 |
| Table 39 — Long extended flat space addressing format.....                                       | 104 |
| Table 40 — Logical unit not specified extended addressing format.....                            | 104 |
| Table 41 — CONTROL byte.....                                                                     | 119 |
| Table 42 — Status codes.....                                                                     | 120 |
| Table 43 — Status qualifier format.....                                                          | 121 |
| Table 44 — SCOPE field.....                                                                      | 122 |
| Table 45 — QUALIFIER field.....                                                                  | 123 |
| Table 46 — SCSI device conditions that abort commands in a SCSI initiator device.....            | 133 |
| Table 47 — SCSI device conditions that abort commands in a SCSI target device .....              | 134 |
| Table 48 — Task management functions that abort commands.....                                    | 135 |
| Table 49 — Command related conditions that abort commands .....                                  | 137 |
| Table 50 — Command handling when ACA is not in effect .....                                      | 140 |
| Table 51 — Handling for new commands received on a faulted I_T nexus during ACA .....            | 142 |
| Table 52 — Handling for new commands received on non-faulted I_T nexuses during ACA .....        | 143 |
| Table 53 — Unit attention condition precedence level.....                                        | 147 |
| Table 54 — Unit attention additional sense codes for events detected by SCSI target devices..... | 154 |
| Table 55 — Task Management Functions .....                                                       | 159 |
| Table 56 — Additional Response Information argument for QUERY TASK .....                         | 164 |
| Table 57 — Additional Response Information argument for QUERY ASYNCHRONOUS EVENT.....            | 165 |
| Table 58 — UADE DEPTH field .....                                                                | 165 |

|                                                                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Table 59 — Task attributes .....                                                                                                                             | 172 |
| Table 60 — Command priority .....                                                                                                                            | 173 |
| Table 61 — Aborting commands already in a task set if an ACA is not established.....                                                                         | 182 |
| Table 62 — Handling of commands already in a task set if an ACA is established.....                                                                          | 183 |
| Table 63 — Task attribute and state indications in examples.....                                                                                             | 193 |
| Table 64 — Dormant command blocking boundary requirements.....                                                                                               | 195 |
| Table A.1 — Identifier attribute size and support requirements .....                                                                                         | 198 |
| Table A.2 — Name attribute size and support requirements .....                                                                                               | 198 |
| Table A.3 — Identifier attribute size for each SCSI transport protocol.....                                                                                  | 199 |
| Table A.4 — Identifier attribute format for each SCSI transport protocol.....                                                                                | 200 |
| Table A.5 — Name attribute size for each SCSI transport protocol.....                                                                                        | 201 |
| Table A.6 — Name attribute format for each SCSI transport protocol .....                                                                                     | 202 |
| Table B.1 — SCSI Initiator Port attributes and SCSI Target Port attributes that are supported by ADT-2,<br>FCP-4, and iSCSI SCSI transport protocols .....   | 204 |
| Table B.2 — SCSI Initiator Port attributes and SCSI Target Port attributes that are supported by SOP,<br>SAS SSP, SRP, and UAS SCSI transport protocols..... | 206 |
| Table C.1 — Terminology mapping to SAM-3 .....                                                                                                               | 207 |
| Table C.2 — Terminology mapping to SAM-4 .....                                                                                                               | 207 |

|                                                                                                     |     |
|-----------------------------------------------------------------------------------------------------|-----|
| Figure 1 — SCSI document structure .....                                                            | 14  |
| Figure 2 — Examples of association relationships in class diagrams.....                             | 37  |
| Figure 3 — Examples of aggregation relationships in class diagrams.....                             | 38  |
| Figure 4 — Example of generalization relationships in class diagrams.....                           | 39  |
| Figure 5 — Example of a dependency relationship in class diagrams.....                              | 39  |
| Figure 6 — Examples of link relationships for object diagrams .....                                 | 41  |
| Figure 7 — State machine conventions .....                                                          | 42  |
| Figure 8 — Example description for one state machine .....                                          | 43  |
| Figure 9 — Requirements precedence .....                                                            | 48  |
| Figure 10 — Client-server model .....                                                               | 49  |
| Figure 11 — SCSI client-server model.....                                                           | 50  |
| Figure 12 — SCSI I/O system and SCSI domain model.....                                              | 52  |
| Figure 13 — SCSI Domain class diagram overview .....                                                | 53  |
| Figure 14 — SCSI Domain class diagram .....                                                         | 54  |
| Figure 15 — SCSI Domain object diagram .....                                                        | 54  |
| Figure 16 — SCSI Device class diagram.....                                                          | 55  |
| Figure 17 — SCSI Port class diagram .....                                                           | 57  |
| Figure 18 — SCSI Target Device class diagram .....                                                  | 64  |
| Figure 19 — Level 1 Hierarchical Logical Unit class.....                                            | 66  |
| Figure 20 — Logical Unit class diagram .....                                                        | 70  |
| Figure 21 — SCSI Initiator Device class diagram.....                                                | 80  |
| Figure 22 — Hierarchical LUN structure adjustments.....                                             | 93  |
| Figure 23 — Logical unit selection using the peripheral device addressing format .....              | 97  |
| Figure 24 — Logical unit selection using the logical unit addressing format.....                    | 99  |
| Figure 25 — SCSI device functional models.....                                                      | 105 |
| Figure 26 — SCSI device with multiple SCSI target ports structure model .....                       | 106 |
| Figure 27 — SCSI initiator device with multiple SCSI initiator ports structure model .....          | 107 |
| Figure 28 — SCSI device with multiple SCSI ports structure model.....                               | 108 |
| Figure 29 — SCSI target device configured in a single SCSI domain .....                             | 109 |
| Figure 30 — SCSI target device configured in multiple SCSI domains .....                            | 110 |
| Figure 31 — SCSI target device and SCSI initiator device configured in a single SCSI domain.....    | 111 |
| Figure 32 — Protocol service reference model.....                                                   | 112 |
| Figure 33 — SCSI transport protocol service mode.....                                               | 113 |
| Figure 34 — Request-Response SAL transaction and related STPL services .....                        | 114 |
| Figure 35 — SCSI transport protocol service model for data transfers.....                           | 114 |
| Figure 36 — Device server data transfer transaction and related STPL services .....                 | 115 |
| Figure 37 — SCSI transport protocol service model for Terminate Data Transfer .....                 | 115 |
| Figure 38 — Device server Terminate Data Transfer transaction and related STPL services .....       | 116 |
| Figure 39 — Model for Data-In and Data-Out data transfers .....                                     | 127 |
| Figure 40 — Command processing events.....                                                          | 139 |
| Figure 41 — Events and event notifications for SCSI target devices.....                             | 152 |
| Figure 42 — Events and event notifications for SCSI initiator devices .....                         | 153 |
| Figure 43 — Task management processing events.....                                                  | 169 |
| Figure 44 — LU (logical unit) state machines (part 1).....                                          | 175 |
| Figure 45 — LU (logical unit) state machines (part 2).....                                          | 176 |
| Figure 46 — Commands having the HEAD OF QUEUE task attribute and blocking boundaries (example 1)... | 193 |
| Figure 47 — Commands having the HEAD OF QUEUE task attribute and blocking boundaries (example 2)... | 194 |
| Figure 48 — Commands having ORDERED task attributes and blocking boundaries.....                    | 195 |
| Figure 49 — Commands having ACA task attributes example .....                                       | 196 |

## INFORMATION TECHNOLOGY – SMALL COMPUTER SYSTEM INTERFACE (SCSI) –

### Part 415: SCSI Architecture Model - 5 (SAM-5)

#### FOREWORD

- 1) ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.
- 2) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees and ISO member bodies.
- 3) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC National Committees and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO, IEC or ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.
- 5) ISO and IEC do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. ISO or IEC are not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC National Committees or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.
- 8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this ISO/IEC publication may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14776-415 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 14776 series, under the general title *Information technology – Small computer system interface (SCSI)*, can be found on the ISO and IEC web sites.

The text of this document is based on the following documents:

|                    |                    |
|--------------------|--------------------|
| CDV                | Report on voting   |
| JTC1-SC25/2828/CDV | JTC1-SC25/2862/RVC |

Full information on the voting for the approval of this document can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2, except as described in 3.4 and 3.5.

A bilingual version of this publication may be issued at a later date.

**IMPORTANT - The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.**

## INTRODUCTION

The purpose of this document is to provide a basis for the coordination of SCSI standards development and to define requirements, common to all SCSI technologies and implementations, that are essential for compatibility with host SCSI application software and device-resident firmware across all SCSI transport protocols. These requirements are defined through a reference model that specifies the behavior and abstract structure that is generic to all SCSI I/O system implementations.

Figure 1 shows the relationship of this document to the other standards and related projects in ISO/IEC 14776 (all parts) as of the publication of this document.

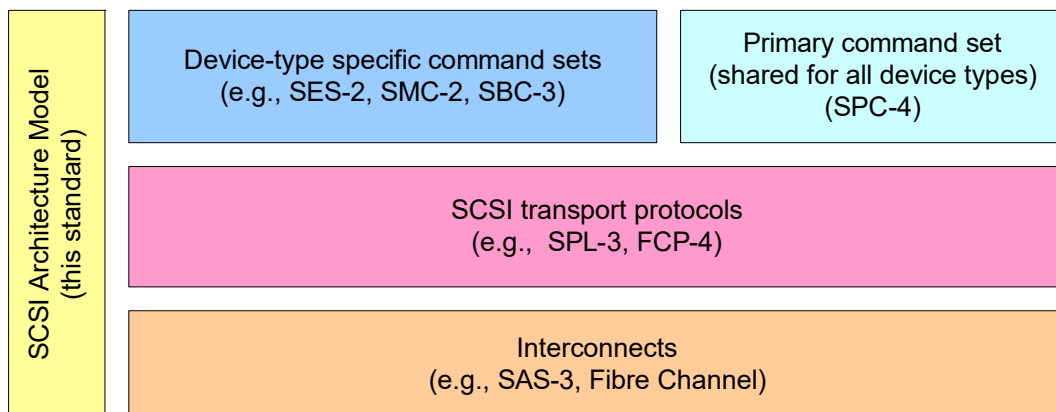


**Figure 1 — SCSI document structure**

The SCSI document structure in figure 1 is intended to show the general applicability of the documents to one another. Figure 1 is not intended to imply any hierarchy, protocol stack, or system architecture relationship.

The functional areas identified in figure 1 characterize the scope of standards within a group as follows:

**SCSI Architecture Model:** Defines the SCSI systems model, the functional partitioning of the SCSI standard set and requirements applicable to all SCSI implementations and implementation standards.

**Device-Type Specific Command Sets:** Implementation standards that define specific device types including a device model for each device type. These standards specify the required commands and behaviors that are specific to a given device type and prescribe the requirements to be followed by a SCSI initiator device when sending commands to a SCSI target device having the specific device type. The commands and behaviors for a specific device type may include by reference commands and behaviors that are defined by other command sets.

**Shared Command Set:** An implementation standard that defines a model for all SCSI device types. This document specifies the required commands and behavior that is common to all SCSI devices, regardless of device type, and prescribes the requirements to be followed by a SCSI initiator device when sending commands to any SCSI target device.

**SCSI Transport Protocols:** Implementation standards that define the requirements for exchanging information so that different SCSI devices are capable of communicating.

**Interconnects:** Implementation standards that define the communications mechanism employed by the SCSI transport protocols. These standards may describe the electrical and signaling requirements essential for SCSI devices to interoperate over a given interconnect. Interconnect standards may allow the interconnection of devices other than SCSI devices in ways that are outside the scope of this document.

The term SCSI is used to refer to ISO/IEC 14776 (all parts).

# INFORMATION TECHNOLOGY - SMALL COMPUTER SYSTEM INTERFACE (SCSI) -

## Part 415: SCSI Architecture Model - 5 (SAM-5)

### 1 Scope

The set of Small Computer System Interface (SCSI) standards consists of this document and the SCSI implementation standards described in 4.2. This document defines a reference model that specifies common behaviors for SCSI devices, and an abstract structure that is generic to all SCSI I/O system implementations.

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

The following concepts from previous versions of the SAM standard are made obsolete by this version:

- a) support for the SPI-5 SCSI transport protocol;
- b) Contingent Allegiance;
- c) the TARGET RESET task management function;
- d) basic task management model;
- e) untagged tasks; and
- f) linked command function.

### 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14776-454, *Information technology - Small computer system interface (SCSI) - Part 454: SCSI Primary Commands - 4 (SPC-4)*

INCITS 497-2012, *Information Technology Automation/Drive Interface Commands - 3 (ADC-3)*

### 3 Terms, definitions, symbols, abbreviations, and conventions

#### 3.1 Terms and definitions

##### 3.1.1

##### **additional sense code**

combination of the ADDITIONAL SENSE CODE field and the ADDITIONAL SENSE CODE QUALIFIER field in the sense data (see 3.1.104 and SPC-4)

##### 3.1.2

##### **aggregation**

form of association that defines a whole-part relationship between the whole (i.e., aggregate) and its parts

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: See 3.6.

### 3.1.3

#### **application client**

class whose objects are, or an object that is, the source of commands and task management function requests

Note 1 to entry: See 4.6.28.

### 3.1.4

#### **argument**

information provided as input to or output from a procedure call (see 3.1.76)

### 3.1.5

#### **association**

relationship between two or more classes that specifies connections among their objects

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: An association is a relationship that specifies that objects of one class are connected to objects of another class.

Note 3 to entry: See 3.6.

### 3.1.6

#### **attribute**

<class (see 3.1.11)> named property of a class that describes a range of values that the class or its objects may hold

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: When referring to objects, an attribute is a named property of the object.

Note 3 to entry: See 3.6.

### 3.1.7

#### **auto contingent allegiance**

##### **ACA**

task set condition established following the completion of a command with CHECK CONDITION status when the NACA bit is set to one in the CONTROL byte

Note 1 to entry: See 5.9.

### 3.1.8

#### **background operation**

operation started by a command that continues processing after the command is no longer in the task set

Note 1 to entry: See 5.5.

### 3.1.9

#### **blocking boundary**

task set boundary denoting a set of conditions that inhibit commands outside the boundary from requesting to be processed by the device server

Note 1 to entry: See 8.8.

### 3.1.10

#### **byte**

8-bit construct

**3.1.11****class**

description of a set of objects that share the same attributes, operations, relationships and semantics

Note 1 to entry: Classes may have attributes and may support operations.

Note 2 to entry: Examples of class relationships are aggregation, association, generalization, and dependency.

Note 3 to entry: See 3.6.

**3.1.12****class diagram**

visual representation that shows a set of classes and their relationships

Note 1 to entry: Class diagrams are used to illustrate the static design view of a system.

Note 2 to entry: See 3.6.3.

**3.1.13****client**

entity that requests a service from a server

Note 1 to entry: This document defines one client, the application client.

**3.1.14****client-server**

relationship established between a pair of entities where one (the client) requests the other (the server) to perform some operation or unit of work on the client's behalf

Note 1 to entry: See 4.4.

**3.1.15****command  
task**

request describing a unit of work to be performed by a device server

Note 1 to entry: Task was the term used by previous versions of the SAM standard (see Annex C).

**3.1.16****command completion**

completion of work associated with command processing

Note 1 to entry: To indicate command completion, a device server or a task manager sends a Send Command Complete SCSI transport protocol service response (see 5.4.2.4) for the command

**3.1.17****command descriptor block  
CDB**

structure used to communicate a command from an application client to a device server having a fixed length of 6 bytes, 10 bytes, 12 bytes, or 16 bytes, or a variable length of between 12 bytes and 260 bytes

Note 1 to entry: See 5.2 and SPC-4.

**3.1.18****command duration limit**

maximum duration of command processing specified by an application client

Note 1 to entry: See 8.6.

### 3.1.19

#### **command identifier**

##### **task tag**

numerical identifier of a command (see 3.1.15)

Note 1 to entry: See 4.6.31.3.

Note 2 to entry: Task tag was the term used by previous versions of the SAM standard (see Annex C).

### 3.1.20

#### **command priority**

relative scheduling importance of a command having the SIMPLE task attribute among the set of commands having the SIMPLE task attribute already in the task set

Note 1 to entry: See 8.5.

### 3.1.21

#### **command standard**

SCSI standard that defines the model, commands, and parameter data for a device type (e.g., SPC-4, SBC-3)

Note 1 to entry: See clause 1.

### 3.1.22

#### **completed command**

command that has completed with a service response of COMMAND COMPLETE

### 3.1.23

#### **confirmation**

acknowledge returned to an application client or device server that signals the completion of a service request

### 3.1.24

#### **constraint**

<class (see 3.1.11), object> mechanism for specifying semantics or conditions that are maintained as true between entities

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: An example of a constraint is a required condition between associations.

Note 3 to entry: See 3.6.

### 3.1.25

#### **copy manager**

class whose objects are each, or an object that is, an application client that processes third-party copy commands and manages copy operations (see SPC-4) from within a logical unit

Note 1 to entry: See 4.6.20.

### 3.1.26

#### **current command**

command that has a data transfer SCSI transport protocol service request in progress (see 5.4.3) or is in the process of sending command status

Note 1 to entry: Each SCSI transport protocol standard may define the SCSI transport protocol specific conditions under which a command is considered a current command.

### 3.1.27

#### **deferred error**

error generated by a background operation (see SPC-4)

**3.1.28****dependency**

relationship between two elements in which a change to one element (e.g., the server) may affect or supply information needed by the other element (e.g., the client)

**3.1.29****dependent logical unit**

logical unit that is addressed via some other logical unit(s) in a hierarchical LUN structure

Note 1 to entry: See 4.6.18.4.

**3.1.30****device model**

description of a type of SCSI target device

Note 1 to entry: Examples of SCSI target devices are a block device and a stream device.

**3.1.31****device server**

class whose objects process, or an object that processes, commands according to the requirements for command management described in clause 8

Note 1 to entry: See 4.6.19.

**3.1.32****device service request**

request submitted by an application client conveying a command to a device server

**3.1.33****device service response**

response returned to an application client by a device server on completion of a command

**3.1.34****extended logical unit addressing**

logical unit addressing method in which the address method field is set to 11b

Note 1 to entry: See 4.7.7.5.

**3.1.35****faulted I\_T nexus**

I\_T nexus on which a command was terminated with CHECK CONDITION status that resulted in the establishment of an ACA

Note 1 to entry: The faulted I\_T nexus condition is cleared when the ACA condition is cleared.

**3.1.36****field**

group of one or more contiguous bits, part of a larger structure (e.g., a CDB (see 3.1.17) or sense data (see 3.1.104))

**3.1.37****generalization**

<class (see 3.1.11)> relationship among classes where one class (i.e., superclass) shares the attributes and operations of one or more classes (i.e., subclasses)

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: See 3.6.

### **3.1.38**

#### **hard reset**

condition resulting from a power on condition or a reset event in which the SCSI device performs the hard reset operations described in 6.3.2, SPC-4, and the appropriate command standards

### **3.1.39**

#### **hierarchical LUN structure**

inverted tree structure for forming and parsing LUNs (see 3.1.58) containing up to four addressable levels (see 4.6.11)

Note 1 to entry: See 4.7.6.3.

### **3.1.40**

#### **hierarchical logical unit**

logical unit that is addressed by a LUN in a hierarchical LUN structure

Note 1 to entry: See 4.6.15.

### **3.1.41**

#### **I\_T nexus**

nexus between a SCSI initiator port and a SCSI target port

Note 1 to entry: See 4.6.32.2.

### **3.1.42**

#### **I\_T nexus loss**

condition resulting from a hard reset condition or an I\_T nexus loss event in which the SCSI device performs the I\_T nexus loss operations described in 6.3.4, SPC-4, and the appropriate command standards

### **3.1.43**

#### **I\_T nexus loss event**

SCSI transport protocol specific event that results in an I\_T nexus loss condition as described in 6.3.4

### **3.1.44**

#### **I\_T nexus transaction**

information transferred between SCSI ports in a single data structure with defined boundaries (e.g., an information unit)

### **3.1.45**

#### **I\_T\_L nexus**

nexus between a SCSI initiator port, a SCSI target port, and a logical unit

Note 1 to entry: See 4.6.32.3.

### **3.1.46**

#### **identifier**

label of an object that is unique within a specified context and that may change

### **3.1.47**

#### **implicit head of queue**

optional processing model for specific commands wherein a command may be treated as if it had been received with a HEAD OF QUEUE task attribute

Note 1 to entry: See 8.2.

**3.1.48****incorrect logical unit number**

logical unit number of a logical unit that does not exist in the SCSI target device when addressed through a given I\_T nexus

Note 1 to entry: See 4.7.1.

**3.1.49****initiator port identifier**

value by which a SCSI initiator port is referenced within a SCSI domain

Note 1 to entry: See 4.6.8.2.

**3.1.50****initiator port name**

name (see 3.1.64) of a SCSI initiator port that is world wide unique within the SCSI transport protocol of the SCSI domain of that SCSI initiator port

Note 1 to entry: The name may be made available to other SCSI devices or SCSI ports in that SCSI domain in SCSI transport protocol specific ways.

Note 2 to entry: See 4.6.8.3.

**3.1.51****instance**

concrete manifestation of an abstraction to which a set of operations may be applied and which may have a state that stores the effects of the operation

Note 1 to entry: An object is an instance of a class.

**3.1.52****in transit**

delivered to a service delivery subsystem for transmission, but not yet arrived at the intended recipient

Note 1 to entry: See 4.6.3.

**3.1.53****layer**

subdivision of the architecture constituted by SCSI initiator device and SCSI target device elements at the same level relative to the interconnect

**3.1.54****link**

individual connection between two objects in an object diagram representing an instance of a relationship

**3.1.55****logical unit**

class whose objects implement, or an object that implements, a device model that manages and processes commands sent by an application client

Note 1 to entry: See 4.6.18.

**3.1.56****logical unit inventory**

list of LUNs reported by a REPORT LUNS command (see SPC-4)

### 3.1.57

#### **logical unit name**

name (see 3.1.64) of a logical unit that is world wide unique within the SCSI transport protocol of a SCSI domain in which the SCSI device containing the logical unit has SCSI ports (see 4.6.4.2)

Note 1 to entry: The logical unit name may be made available to other SCSI devices or SCSI ports in SCSI transport protocol specific ways.

### 3.1.58

#### **logical unit number**

##### **LUN**

<transport protocol standards> 64-bit or 16-bit identifier for a logical unit

Note 1 to entry: See 4.7.

### 3.1.59

#### **logical unit number**

##### **LUN**

<command standards> 64-bit identifier for a logical unit

Note 1 to entry: See 4.7.

### 3.1.60

#### **logical unit reset**

condition resulting from a hard reset condition or a logical unit reset event in which the logical unit performs the logical unit reset operations described in 6.3.3, SPC-4, and the appropriate command standards

### 3.1.61

#### **logical unit reset event**

event that results in a logical unit reset condition as described in 6.3.3

### 3.1.62

#### **management logical unit**

class whose objects are each, or an object that is, a logical unit that only performs management functions

Note 1 to entry: Management logical units allow an application client to issue requests to receive specific information and manage specific information relating to a SCSI target device.

Note 2 to entry: See 4.6.16.

### 3.1.63

#### **multiplicity**

indication of the range in number of allowable instances that a class or an attribute may have

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: See 3.6.

### 3.1.64

#### **name**

label of an object that is unique within a defined context and should never change

### 3.1.65

#### **nexus**

I\_T nexus or I\_T\_L nexus

### 3.1.66

#### **non-faulted I\_T nexus**

I\_T nexus that is not a faulted I\_T nexus (see 3.1.35)

**3.1.67****object**

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

Note 1 to entry: All objects are instances of classes (see 3.1.51).

**3.1.68****object diagram**

visual representation that shows a set of objects and their relationships at a point in time

Note 1 to entry: Object diagrams are used to illustrate static snapshots of instances of the things found in class diagrams.

Note 2 to entry: See 3.6.4.

**3.1.69****operation**

service that may be requested from any object of the class in order to affect behavior

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: Operations describe what a class is allowed to do and may be a request or a query.

Note 3 to entry: An operation that is a request may change the state of the object but an operation that is a query should not.

Note 4 to entry: See 3.6.

**3.1.70****peer entities**

entities within the same layer (see 3.1.53)

**3.1.71****power loss expected**

condition resulting from a power loss expected event in which the logical unit performs the power loss expected operations described in 6.3.5, SPC-4, and the appropriate transport protocol and command standards

**3.1.72****power loss expected event**

event that results in a power loss expected condition (see 3.1.71) as described in 6.3.5

**3.1.73****power on**

condition resulting from a power on event in which the SCSI device performs the power on operations described in 6.3.1, SPC-4, and the appropriate command standards

**3.1.74****power on event**

power being applied to a SCSI device, resulting in a power on condition as described in 6.3.1

**3.1.75****procedure**

operation that is invoked through an external calling interface

**3.1.76**

**procedure call**

model used by this document for the interfaces involving both the SAL (see 3.1.85) and STPL (see 3.1.97), having the appearance of a programming language function call

Note 1 to entry: See 3.9.

**3.1.77**

**protocol**

specification and/or implementation of the requirements governing the content and exchange of information passed between distributed entities through a service delivery subsystem

**3.1.78**

**queue**

class whose objects maintain, or an object that maintains, an arrangement of commands within a task set (see 3.1.122)

**3.1.79**

**receiver**

client or server that is the recipient of a service delivery transaction

**3.1.80**

**reference model**

standard model used to specify system requirements in an implementation independent manner

**3.1.81**

**relative port identifier**

identifier for a SCSI port that is unique within a SCSI device

Note 1 to entry: See 4.6.5.2.

**3.1.82**

**request-response transaction**

interaction between a pair of distributed, cooperating entities, consisting of a request for service submitted to an entity followed by a response conveying the result

**3.1.83**

**reset event**

SCSI transport protocol specific event that results in a hard reset condition as described in 6.3.2

**3.1.84**

**role**

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.

Note 1 to entry: This definition only applies when used in relation to UML.

Note 2 to entry: See 3.6.

**3.1.85**

**SCSI application layer**

**SAL**

set of protocols and procedures that implement or issue commands and task management functions by using services provided by a STPL (see 3.1.97)

**3.1.86****SCSI device**

class whose objects are, or an object that is, connected to a service delivery subsystem and supports a SCSI application protocol

Note 1 to entry: See 4.6.4.

**3.1.87****SCSI device name**

name (see 3.1.64) of a SCSI device that is world wide unique within the SCSI transport protocol of a SCSI domain in which the SCSI device has SCSI ports (see 4.6.4.2)

Note 1 to entry: The SCSI device name may be made available to other SCSI devices or SCSI ports in SCSI transport protocol specific ways.

**3.1.88****SCSI domain**

I/O system consisting of a set of SCSI devices and a service delivery subsystem, where the SCSI devices interact with one another by means of the service delivery subsystem

**3.1.89****SCSI event**

condition defined by this document that is detected by a SCSI device and that requires notification of its occurrence within the SCSI device

Note 1 to entry: See clause 6.

Note 2 to entry: An example of a SCSI event is a logical unit reset.

**3.1.90****SCSI I/O operation**

operation defined by a command or a task management function

**3.1.91****SCSI I/O system**

I/O system, consisting of two or more SCSI devices, a SCSI interconnect, and a SCSI transport protocol that collectively interact to perform SCSI I/O operations

**3.1.92****SCSI initiator device**

class whose objects originate, or an object that originates, device service and task management requests to be processed by a SCSI target device and receives device service and task management responses from SCSI target devices

**3.1.93****SCSI initiator port**

class whose objects each act, or an object that acts, as the connection between application clients and a service delivery subsystem through which server requests and server responses are routed

Note 1 to entry: See 4.6.8.

### **3.1.94**

#### **SCSI port**

class whose objects connect, or an object that connects, the application client, device server or task manager to a service delivery subsystem through which server requests and server responses are routed

Note 1 to entry: A SCSI port is either a SCSI initiator port (see 3.1.93), a SCSI target port (see 3.1.96), or both.

Note 2 to entry: See 4.6.5.

### **3.1.95**

#### **SCSI target device**

class whose objects receive, or an object that receives, device service requests and task management requests for processing and sends device service responses and task management responses to SCSI initiator devices

Note 1 to entry: See 4.6.9.

### **3.1.96**

#### **SCSI target port**

class whose objects each act, or an object that acts, as the connection between device servers and task managers and a service delivery subsystem through which server requests and server responses are routed

Note 1 to entry: See 4.6.6.

### **3.1.97**

#### **SCSI transport protocol layer**

##### **STPL**

set of protocols and services used by a SAL (see 3.1.85) to transport data representing a SCSI application protocol transaction

### **3.1.98**

#### **SCSI transport protocol service confirmation**

procedure call from the STPL notifying the SAL that a SCSI transport protocol service request has completed

### **3.1.99**

#### **SCSI transport protocol service indication**

procedure call from the STPL notifying the SAL that a SCSI transport protocol transaction has occurred

### **3.1.100**

#### **SCSI transport protocol service request**

procedure call to the STPL to begin a SCSI transport protocol service transaction

### **3.1.101**

#### **SCSI transport protocol service response**

procedure call to the STPL containing a reply from the SAL in response to a SCSI transport protocol service indication

### **3.1.102**

#### **SCSI transport protocol specific**

<referenced item> having the property of being defined by a SCSI transport protocol standard (see clause 2 and Bibliography)

### **3.1.103**

#### **sender**

client or server that originates a service delivery transaction

**3.1.104****sense data**

data describing command completion information that a device server delivers to an application client in the same I\_T nexus transaction as the status or as parameter data in response to a REQUEST SENSE command

Note 1 to entry: See 5.13 and SPC-4.

**3.1.105****sense key**

SENSE KEY field in the sense data (see 3.1.104 and SPC-4)

**3.1.106****server**

entity that performs a service on behalf of a client

**3.1.107****server request**

transaction from a client to a server invoking a service

**3.1.108****server response**

transaction from a server to a client conveying the result of a request

**3.1.109****service**

operation or function performed by a SCSI object that is invoked by other SCSI objects

**3.1.110****service delivery failure**

non-recoverable error causing the corruption or loss of one or more service delivery transactions while in transit

**3.1.111****service delivery subsystem**

class whose objects are, or an object that is, part of a SCSI I/O system that transmits service requests to a logical unit or SCSI target device and returns logical unit or SCSI target device responses to a SCSI initiator device

Note 1 to entry: See 4.6.3.

**3.1.112****service delivery transaction**

request or response sent through a service delivery subsystem

**3.1.113****standard INQUIRY data**

data returned to an application client as a result of an INQUIRY command (see SPC-4) with the EVPD bit set to zero

Note 1 to entry: Fields in the standard INQUIRY data are referenced by name in this document.

**3.1.114****target port identifier**

value by which a SCSI target port is referenced within a SCSI domain

Note 1 to entry: See 4.6.6.2.

### **3.1.115**

#### **target port name**

name (see 3.1.64) of a SCSI target port that is world wide unique within the SCSI transport protocol of the SCSI domain of that SCSI target port

Note 1 to entry: The name may be made available to other SCSI devices or SCSI ports in that SCSI domain in SCSI transport protocol specific ways.

Note 2 to entry: See 4.6.6.3.

### **3.1.116**

#### **task attribute**

attribute of a command (see 3.1.15) that specifies the processing relationship of a command with regard to other commands in the task set (see 3.1.122)

Note 1 to entry: See 8.4.

### **3.1.117**

#### **task management function**

task manager service capable of being requested by an application client to affect the processing of one or more commands

Note 1 to entry: See clause 7.

### **3.1.118**

#### **task management request**

request submitted by an application client, invoking a task management function to be processed by a task manager

### **3.1.119**

#### **task management response**

response returned to an application client by a task manager in reply to a task management request

### **3.1.120**

#### **task manager**

class whose objects control, or an object that controls, the sequencing of commands and processes task management functions

Note 1 to entry: See 4.6.21.

### **3.1.121**

#### **task router**

class whose objects route, or an object that routes, commands and task management functions between a service delivery subsystem (see 3.1.111) and the appropriate task manager(s)

Note 1 to entry: See 4.6.7.

### **3.1.122**

#### **task set**

class whose objects are, or an object that is, a group of commands within a logical unit, whose interaction is dependent on the task management (e.g., queuing) and ACA requirements

Note 1 to entry: See 4.6.22.

### **3.1.123**

#### **token**

representation of a collection of data

**3.1.124****transaction**

cooperative interaction between two entities, involving the exchange of information or the processing of some request by one entity on behalf of the other

**3.1.125****transport protocol standard**

SCSI standard that defines a transport protocol (e.g., SPL-3)

**3.1.126****well known logical unit**

class whose objects are each, or an object that is, a logical unit that only performs specific functions

Note 1 to entry: Well known logical units allow an application client to issue requests to receive and manage specific information relating to a SCSI target device.

Note 2 to entry: See 4.6.17.

**3.1.127****well known logical unit number****W-LUN**

LUN that identifies a well known logical unit

Note 1 to entry: See 4.7.7.5.2.

## 3.2 Symbols and Abbreviations

### 3.2.1 Abbreviations

| Abbreviation | Meaning                                                                           |
|--------------|-----------------------------------------------------------------------------------|
| ACA          | Auto Contingent Allegiance (see 3.1.7)                                            |
| ADC-3        | Automation/Drive Interface - Commands - 3 (see clause 2)                          |
| ADT-2        | Automation/Drive Interface Transport Protocol - 2 (see Bibliography)              |
| CDB          | Command Descriptor Block (see 3.1.17)                                             |
| CRN          | Command Reference Number                                                          |
| EUI-64       | Extended Unique Identifier (see Annex D)                                          |
| FCP-4        | Fibre Channel Protocol - 4 (see Bibliography)                                     |
| FC-SB-5      | Fibre Channel - Single-Byte Command Code Sets - 5 Mapping Protocol (see clause 2) |
| iSCSI        | Internet SCSI (see Bibliography)                                                  |
| ISO          | International Organization for Standardization                                    |
| LUN          | Logical Unit Number (see 3.1.58)                                                  |
| NAA          | Name Address Authority (see Annex D)                                              |
| n/a          | Not Applicable                                                                    |
| RAID         | Redundant Array of Independent Disks                                              |
| ROD          | Representation Of Data (see SPC-4)                                                |
| SAL          | SCSI application layer (see 3.1.85)                                               |
| SAM-2        | SCSI Architecture Model - 2 (see Bibliography)                                    |
| SAM-3        | SCSI Architecture Model - 3 (see Bibliography)                                    |
| SAM-4        | SCSI Architecture Model - 4 (see Bibliography)                                    |
| SAS-3        | Serial Attached SCSI - 3 (see Bibliography)                                       |
| SBC-3        | SCSI Block Commands - 3 (see clause 2)                                            |
| SBP-3        | Serial Bus Protocol - 3 (see Bibliography)                                        |
| SCSI         | The architecture defined by ISO/IEC 14776 (all parts)                             |
| SPC-4        | SCSI Primary Commands - 4 (see clause 2)                                          |
| SRP          | SCSI RDMA Protocol (see Bibliography)                                             |
| STPL         | SCSI transport protocol layer (see 3.1.97)                                        |
| UML          | Unified Modeling Language (see Bibliography)                                      |
| UTF          | Unicode Transformation Format (see Annex D)                                       |
| VPD          | Vital Product Data (see SPC - 4)                                                  |
| W-LUN        | Well known logical unit number (see 3.1.127)                                      |

### 3.2.2 Units

| Unit | Meaning                               |
|------|---------------------------------------|
| ms   | millisecond (i.e., $10^{-3}$ seconds) |

## 3.3 Keywords

### 3.3.1 invalid

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

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

### 3.3.2

#### **mandatory**

keyword indicating an item that is required to be implemented as defined in this document

### 3.3.3

#### **may**

keyword that indicates flexibility of choice with no implied preference

Note 1 to entry: May is synonymous with the phrase “may or may not”.

### 3.3.4

#### **may not**

keyword that indicates flexibility of choice with no implied preference

Note 1 to entry: May not is synonymous with the phrase “may or may not”.

### 3.3.5

#### **obsolete**

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

### 3.3.6

#### **option**

#### **optional**

keywords that describe features that are not required to be implemented by this document

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

### 3.3.7

#### **prohibited**

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

### 3.3.8

#### **reserved**

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

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

Note 2 to entry: Recipients are not required to check reserved bits, bytes, words, or fields for zero values. Receipt of reserved code values in defined fields shall be reported as an error.

### 3.3.9

#### **restricted**

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

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

### 3.3.10

#### **shall**

keyword indicating a mandatory requirement

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

### 3.3.11

#### **should**

keyword indicating flexibility of choice with a strongly preferred alternative

Note 1 to entry: Equivalent to the phrase “it is strongly recommended”.

### 3.3.12

#### **vendor specific**

keyword indicating that specification of the referenced item is determined by the SCSI device vendor

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

## 3.4 Editorial conventions

Certain words and terms used in this document have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.1 or in the text where they first appear.

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, such as **Execute Command** (see clause 5). Names of arguments are denoted by capitalizing each word in the name. For instance, Sense Data is the name of an argument in the **Execute Command** procedure call.

Quantities having a defined numeric value are identified by large capital letters. CHECK CONDITION, for example, refers to the numeric quantity defined in table 42 (see 5.3.1). Quantities having a discrete but unspecified value are identified using small capital letters. As an example, COMMAND COMPLETE, indicates a quantity returned by the **Execute Command** procedure call (see clause 5). 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 1 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, then the order of precedence to resolve the conflicts is:

- 1) text;

- 2) tables; and
- 3) figures.

Not all tables or figures are fully described in the text. Tables show data format and values.

Notes and examples do not constitute any requirements.

Notes are numbered consecutively throughout this document.

## 3.5 Numeric and character conventions

### 3.5.1 Numeric conventions

A binary number is represented in this document by any sequence of digits consisting of only the Western-Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate organizational boundaries (e.g., 00010101 11001110b, 00010101\_11001110b, 0 0101 1010b or 0\_0101\_1010b).

A hexadecimal number is represented in this document by any sequence of digits consisting of only the Western-Arabic numerals 0 to 9 and/or the upper-case English letters A to F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate organizational boundaries (e.g., 3456FDCA 84BD5E7Ah, 3456FDCA\_84BD5E7Ah, B FD8C FA23h, or B\_FD8C\_FA23h).

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

This document 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 document |
|--------------|--------------|---------------|
| 0,6          | 0.6          | 0.6           |
| 3,141 592 65 | 3.14159265   | 3.141 592 65  |
| 1 000        | 1,000        | 1 000         |
| 1 323 462,95 | 1,323,462.95 | 1 323 462.95  |

A decimal number represented in this document with an overline over one or more digits following the decimal point is a number where the overlined digits are infinitely repeating (e.g., 666.6 means 666.666 666... or 666 2/3, and 12.142 857 means 12.142 857 142 857... or 12 1/7).

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

### 3.5.2 Byte encoded character strings conventions

When this document 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' represent encoded characters is the same as writing out the following sequence of byte values: 53h 43h 53h 49h 20h 64h 65h 76h 69h 63h 65h.

## 3.6 UML notation conventions

### 3.6.1 Notation conventions overview

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

See 3.6.3 for the conventions used for class diagrams.

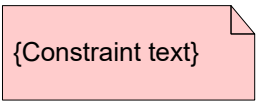
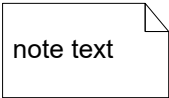
See 3.6.4 for the conventions used for object diagrams.

### 3.6.2 Constraint and note conventions

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

Table 2 shows the notation used for constraints and notes.

**Table 2 — Constraint and note notation**

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

### 3.6.3 Class diagram conventions

Table 3 shows the notation used for classes in class diagrams.

**Table 3 — Class diagram notation for classes**

| Notation                                              |                                                                                            |                   | Description                                                                              |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------|------------------------------------------------------------------------------------------|
| <b>Class Name</b>                                     | <b>Class Name</b>                                                                          | <b>Class Name</b> | A class with no attributes or operations.                                                |
| <b>Class Name</b><br>Attribute01[1]<br>Attribute02[1] | <b>Class Name</b><br>Attribute01[1]<br>Attribute02[1]                                      |                   | A class with attributes and no operations.                                               |
|                                                       | <b>Class Name</b><br>Operation01()<br>Operation02()                                        |                   | A class with operations and no attributes.                                               |
|                                                       | <b>Class Name</b><br>Attribute01[1]<br>Attribute02[1]<br>Operation01()<br>Operation02()    |                   | A class with attributes and operations.                                                  |
|                                                       | <b>Class Name</b><br>Attribute01[1..*]<br>Attribute02[1]<br>Operation01()<br>Operation02() |                   | A class with attributes that have a specified multiplicity (see table 4) and operations. |

Table 4 shows the notation used to indicate multiplicity of instances of classes and attributes in class diagrams.

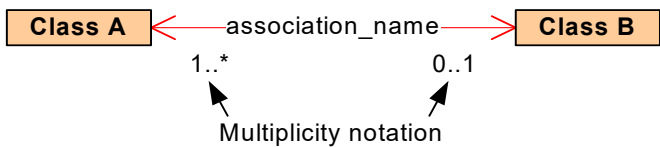
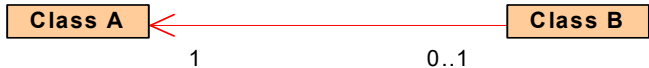
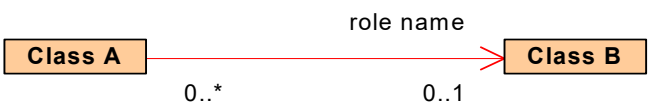
**Table 4 — Multiplicity notation**

| Notation <sup>a</sup>                                                         | Description                                                                   |
|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
|                                                                               | The number of instances of a class or 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). |
| <sup>a</sup> See figure 2 and figure 3 for examples of multiplicity notation. |                                                                               |

Table 5 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 a multiplicity notation (see table 4) at each end of the relationship line.

**Table 5 — 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.

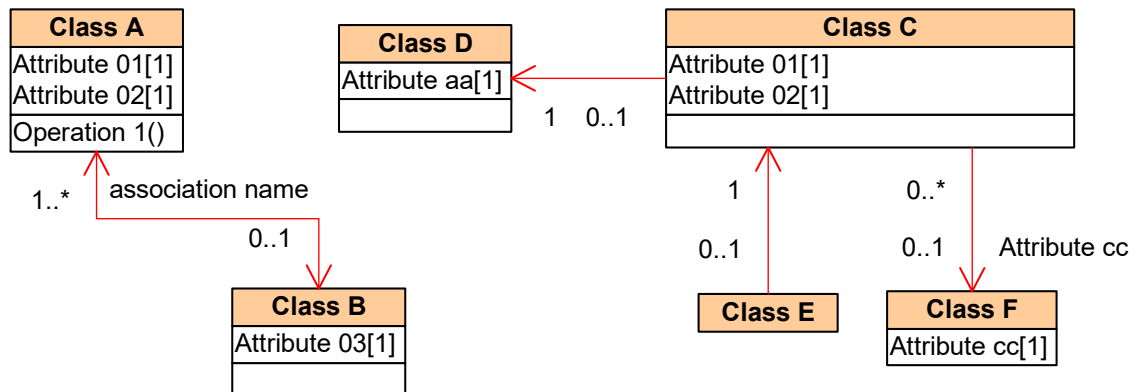


**Figure 2 — Examples of association relationships in class diagrams**

Table 6 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 (see table 5) and always includes multiplicity notation (see table 4) at each end of the relationship line.

**Table 6 — Class diagram notation for aggregations**

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

See figure 3 for examples of aggregation relationships between classes.

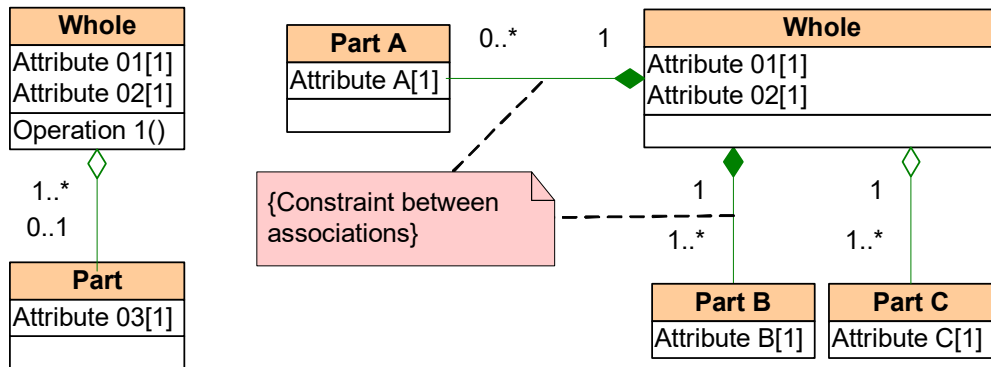


**Figure 3 — Examples of aggregation relationships in class diagrams**

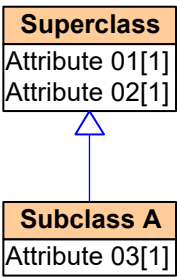
Table 7 shows the notation used to denote generalization (i.e., “is a kind of”) relationships between classes.

**Table 7 — Class diagram notation for generalizations**

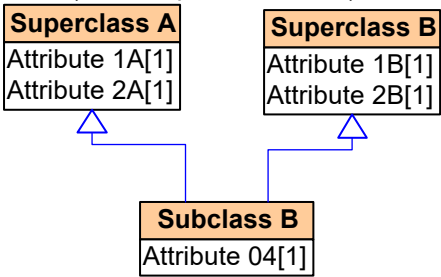
| Notation | Description                                                                                                                                                                                                                |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <p>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). Inherited attributes are not duplicated in UML drawings.</p> |

See figure 4 for examples of generalization relationships between classes.

Single superclass/single subclass:



Multiple superclasses/single subclass  
(i.e., multiple inheritance):



Single superclass/multiple subclasses:

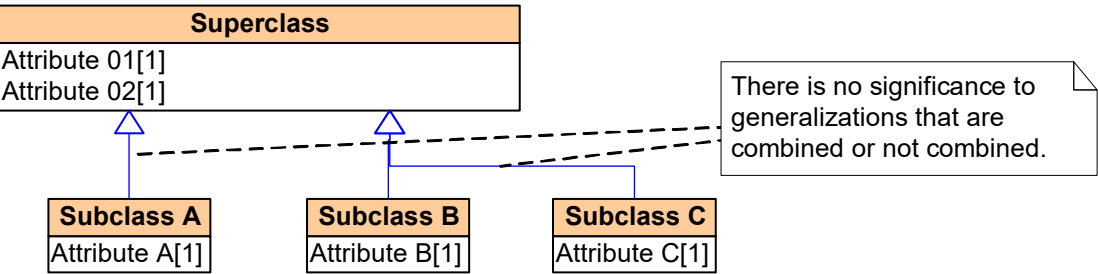


Figure 4 — Example of generalization relationships in class diagrams

Table 8 shows the notation used to denote dependency (i.e., “depends on”) relationships between classes.

Table 8 — 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

### 3.6.4 Object diagram conventions

Table 9 shows the notation used for objects in object diagrams.

**Table 9 — Object diagram notation for objects**

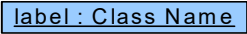
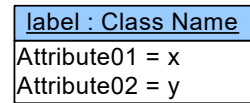
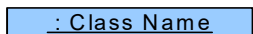
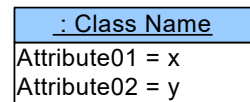
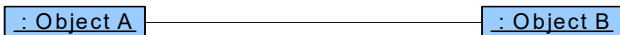
| Notation                                                                            | Description                                          |
|-------------------------------------------------------------------------------------|------------------------------------------------------|
|    | Notation for a named object with no attributes.      |
|    | Notation for a named object with attributes.         |
|    | Notation for an anonymous object with no attributes. |
|  | Notation for an anonymous object with attributes.    |

Table 10 shows the notation used to denote link relationships between objects.

**Table 10 — Object diagram notation for link**

| Notation                                                                            | Description                                                  |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------|
|  | An instance of an association between object A and object B. |

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

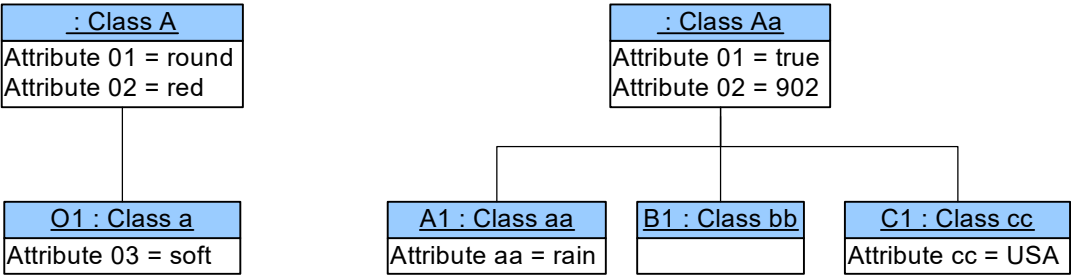


Figure 6 — Examples of link relationships for object diagrams

### 3.7 State machine conventions

#### 3.7.1 State machine conventions overview

Figure 7 shows how state machines are described.

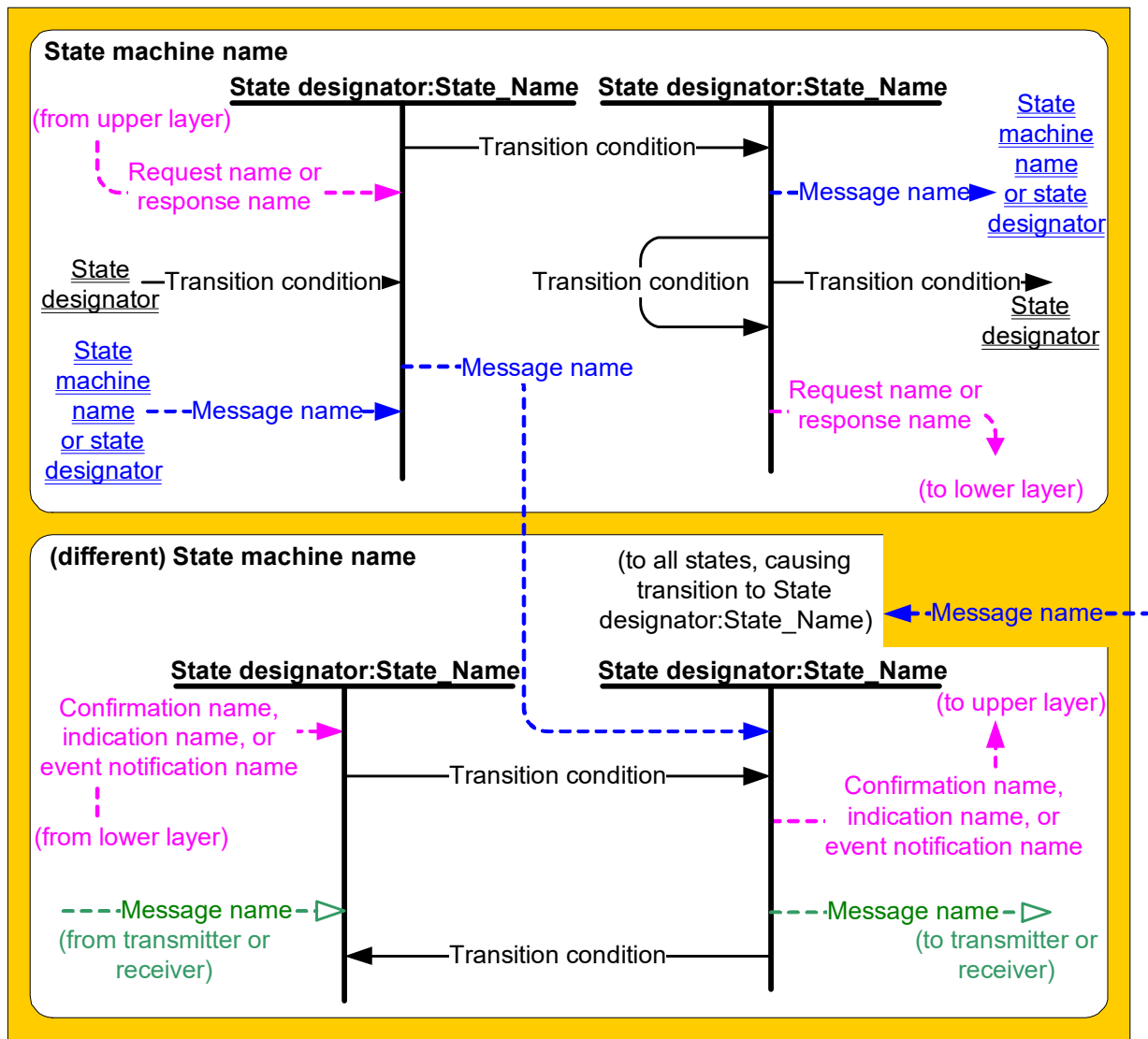


Figure 7 — State machine conventions

Figure 8 shows an example description for one state machine.

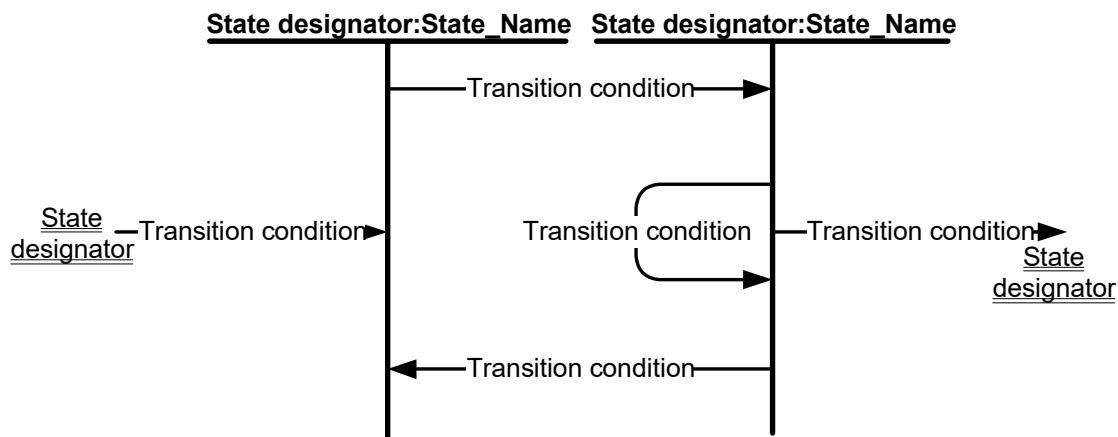


**Figure 8 — Example description for one state machine**

When multiple state machines are present in a figure, they are enclosed in boxes with rounded corners.

Each state is identified by a state designator and a state name. The state designator (e.g., SL1) is unique among all state machines in this document. The state name (e.g., Idle) is a brief description of the primary action taken during the state, and the same state name may be used by other state machines. Actions taken while in each state are described in the state description text.

### 3.7.2 Transitions

Transitions between states are shown with solid lines, with an arrow pointing to the destination state. A transition may be labeled with a transition condition label (i.e., a brief description of the event or condition that causes the transition to occur).

If the state transition exits a figure or enters a figure, then the transition label goes to or from a state designator label with double underlines rather than to or from a state.

The conditions and actions are described fully in the transition description text. In case of a conflict between a figure and the text, the text shall take precedence.

Transitions between states are instantaneous.

Upon entry into a state, all actions to be processed in that state are processed. If a state is re-entered from itself, all actions to be processed in the state are processed again. A state may be entered and exited in zero time if the conditions for exiting the state are valid upon entry into the state.

### 3.7.3 Messages, requests, indications, confirmations, responses, and event notifications

Messages passed between state machines are shown with dashed lines labeled with a message name. When messages are passed between state machines, they are identified by either:

- a dashed line to or from a state machine name label with double underlines and/or state name label with double underlines, if the destination is in a different figure from the source;
- a dashed line to or from a state in another state machine in the same figure; or
- a dashed line from a state machine name label with double underlines to a “(to all states)” label, if the destination is every state in the state machine.

Requests, indications, confirmations, responses, and event notifications are shown with curved dashed lines originating from or going to the top or bottom of the figure. Each request, indication, confirmation, response, and event notification is labeled. The meaning of each request, indication, confirmation, response, and event notification is described in the state description text.

Messages with unfilled arrowheads are passed to or from the state machine's transmitter or receiver, not shown in the state machine figures, and are directly related to data being transmitted on or received from the physical link.

The state machine description text for each state wholly defines the messages sent while the state machine is in that state. If a state is repeatedly sending a message transitions to another state, then that state stops sending that message before making the transition.

### 3.7.4 State machine counters, timers, and variables

State machines may contain counters, timers, and variables that affect the operation of the state machine. The following apply to counters, timers, and variables:

- a) their scope is a single state machine;
- b) they are created and deleted with the state machine or state machines with which they are associated;
- c) their initialization and modification is specified in the state descriptions and the transition descriptions; and
- d) their current values may be used to determine the behavior of a state and to select the transition out of a state.

State machine timers may continue to run while a state machine is in a given state, and a timer may cause a state transition upon reaching a defined threshold value (e.g., zero for a timer that counts down).

## 3.8 Bit and byte ordering

In this document, 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 document, the ordering of elements within a table may constrain the order of storage or transmission of the structure.

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

If a table shows bit numbering (see table 11), the least significant bit (LSB) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If a table shows numbering of bytes or characters (see table 12), the most significant byte or character is represented at the lowest number and each less significant byte or character has the next greater number than the immediately more significant byte.

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

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

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

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

A data dword consists of 32 bits. Table 11 shows a data dword containing a single value, where the MSB is on the upper left in bit 31 and the LSB is on the lower right in bit 0.

**Table 11 — Example of ordering of bits and bytes within a data dword**

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

Table 12 shows a data dword containing four one-byte elements, where byte 0 (the first byte) is on the top and byte 3 (the fourth byte) is on the bottom. Each byte has an MSB on the left and an LSB on the right.

**Table 12 — Example of ordering of bits and bytes within a data dword element**

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

### 3.9 Notation for procedure calls

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

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

where:

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

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

The following is an example of a procedure call specification:

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

Input arguments:

**Pattern:** Argument containing the search pattern.

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

Output arguments:

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

## 4 SCSI architecture model

### 4.1 Overview

The purpose of the SCSI architecture model is to:

- a) provide a basis for the coordination of SCSI standards development that allows all parts in the ISO/IEC 14776 series to be placed into perspective within the overall SCSI architecture model;
- b) establish a layered model in which standards may be developed;
- c) provide a common reference for maintaining consistency among related standards; and
- d) provide the foundation for application compatibility across all SCSI interconnect and SCSI transport protocol environments by specifying generic requirements that apply uniformly to all implementation standards within each functional area.

The development of this document is assisted by the use of a UML based abstract model. To specify the external behavior of a SCSI system, elements in a system are replaced by functionally equivalent components within this model. Only externally observable behavior is retained as the standard of behavior. The description of internal behavior in this document is provided only to support the definition of the observable aspects of the model. Those aspects are limited to the generic properties and characteristics needed for host applications to interoperate with SCSI devices in any SCSI interconnect and SCSI transport protocol environment. The model does not address other requirements that may be essential to some I/O system implementations (e.g., the mapping from SCSI device addresses to network addresses, the procedure for discovering SCSI devices on a network, and the definition of network authentication policies for SCSI initiator devices or SCSI target devices). These considerations are outside the scope of this document.

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

The SCSI architecture model is described in terms of classes, protocol layers, and service interfaces between classes. As used in this document, classes are abstractions, encapsulating a set of related attributes, operations, data types, and other classes. Certain classes are essential to the SCSI architecture (e.g., an interconnect), while others are needed to understand the functioning of the SCSI architecture but have implementation definitions outside the scope of the SCSI architecture (e.g., a command). These classes exhibit well-defined and observable behaviors, but they do not exist as separate physical elements. A class may contain a single attribute or be a complex entity that may:

- a) contain multiple attributes; or
- b) perform a set of operations or services on behalf of another class.

Service interfaces are defined between classes and protocol layers. The template for a distributed service interface is the client-server model described in 4.3. The structure of a SCSI I/O system is specified in 4.5 by defining the relationships among classes. The set of distributed services to be provided are described in clause 5 and clause 7.

Requirements that apply to each SCSI transport protocol standard are described in the SCSI transport protocol service model described in 5.4, 6.4, and 7.12. The model describes required behavior in terms of layers, classes within layers and SCSI transport protocol service transactions between layers.

### 4.2 Compliance requirements

This document defines generic requirements that pertain to SCSI implementation standards and implementation requirements. An implementation requirement defines behavior in terms of measurable or observable parameters that apply to an implementation. Examples of implementation requirements defined in this document are the status values to be returned upon command completion and the service responses to be returned upon task management function completion.