



ISO/IEC 14776-261

Edition 1.0 2012-11

INTERNATIONAL STANDARD



Information technology – Small Computer System Interface (SCSI) –
Part 261: SAS protocol layer (SPL)

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2012 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
Fax: +41 22 919 03 00
info@iec.ch
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 corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you 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It also gives information on projects, replaced and withdrawn publications.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

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

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - 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: csc@iec.ch.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14716-261:2012



ISO/IEC 14776-261

Edition 1.0 2012-11

INTERNATIONAL STANDARD



Information technology – Small Computer System Interface (SCSI) –
Part 261: SAS protocol layer (SPL)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE **XC**

ICS 35.200

ISBN 978-2-83220-483-2

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

CONTENTS

FOREWORD.....	18
INTRODUCTION.....	20
General	20
SCSI standards family.....	21
1 Scope	23
2 Normative references	23
3 Terms, definitions, symbols, abbreviations, keywords, and conventions.....	25
3.1 Terms and definitions	25
3.2 Symbols and abbreviations	46
3.3 Keywords	48
3.4 Editorial conventions	49
3.5 Class diagram and object diagram conventions.....	51
3.6 State machine conventions	55
3.6.1 State machine conventions overview	55
3.6.2 Transitions.....	56
3.6.3 Messages, requests, indications, confirmations, responses, and event notifications	57
3.6.4 State machine counters, timers, and variables	57
3.6.5 State machine arguments	57
3.7 Bit and byte ordering	58
3.8 Notation for procedures and functions	58
4 General	60
4.1 Architecture	60
4.1.1 Architecture overview	60
4.1.2 Physical links and phys	61
4.1.3 Logical links.....	66
4.1.4 Ports (narrow ports and wide ports)	66
4.1.5 Application clients and device servers	69
4.1.6 SAS devices.....	70
4.1.7 Expander devices.....	71
4.1.8 Service delivery subsystem.....	72
4.1.9 Domains	73
4.1.10 Expander device topologies	75
4.1.11 Pathways.....	77
4.1.12 Connections	77
4.1.13 Broadcasts	79
4.2 Names and identifiers	81
4.2.1 Names and identifiers overview	81
4.2.2 NAA IEEE Registered format identifier	82
4.2.3 NAA Locally Assigned format identifier	83
4.2.4 SAS address	84
4.2.5 Hashed SAS addresses	84
4.2.6 Device names and expander device SAS addresses	84
4.2.7 Device name for SATA devices with world wide names	85
4.2.8 Port names.....	85
4.2.9 Port identifiers and SAS port SAS addresses	85
4.2.10 Phy identifiers.....	86
4.3 State machines	87
4.3.1 State machine overview	87
4.3.2 Transmit data path	88
4.3.3 Receive data path	93

4.3.4 State machines and SAS Device, SAS Port, and SAS Phy classes	96
4.4 Resets	98
4.4.1 Reset overview	98
4.4.2 Hard reset	100
4.5 I_T nexus loss	100
4.6 Expander device model	101
4.6.1 Expander device model overview	101
4.6.2 Expander ports	102
4.6.3 Expander connection manager (ECM)	103
4.6.4 Expander connection router (ECR)	103
4.6.5 Broadcast propagation processor (BPP)	104
4.6.6 Expander device interfaces	104
4.6.7 Expander device routing	111
4.6.8 Expander device reduced functionality	115
4.6.9 Broadcast (Expander) handling	116
4.7 Discover process	116
4.7.1 Discover process overview	116
4.7.2 Starting the discover process (Broadcast (Change) handling)	116
4.7.3 Discover process traversal	116
4.7.4 Discover process in a self-configuring expander device	118
4.7.5 Enabling multiplexing	119
4.8 Configuration subprocess	120
4.8.1 Configuration subprocess overview	120
4.8.2 Allowed expander device topologies	120
4.8.3 Externally configurable expander device route table optimization	121
4.8.4 Externally configurable expander device expander route index order	122
4.9 Zoning	128
4.9.1 Zoning overview	128
4.9.2 Zoning expander device requirements	132
4.9.3 Zoning operation	134
4.9.4 Zone phy information and link reset sequences	140
4.9.5 Broadcast processing in a zoning expander device with zoning enabled	142
4.9.6 Zone configuration	143
4.10 Phy power conditions	146
4.10.1 Low phy power conditions	146
4.10.2 SATA phy power conditions	148
4.11 SAS device and expander device power conditions	148
4.12 Phy test functions	149
4.12.1 Phy test functions overview	149
4.12.2 Transmit pattern phy test function	149
4.13 Phy events	150
5 Phy layer	155
5.1 Phy layer overview	155
5.2 8b10b coding	155
5.2.1 8b10b coding overview	155
5.2.2 8b10b coding notation conventions	155
5.3 Character encoding and decoding	156
5.3.1 Introduction	156
5.3.2 Bit transmission order	156
5.3.3 Character transmission order	156
5.3.4 Frame transmission order	156
5.3.5 Running disparity (RD)	157
5.3.6 Data characters	157

5.3.7 Control characters.....	163
5.3.8 Encoding characters in the transmitter.....	164
5.3.9 Decoding characters in the receiver.....	164
5.4 Dwords, primitives, data dwords, and invalid dwords.....	164
5.5 Bit order.....	165
5.6 Out of band (OOB) signals.....	166
5.6.1 OOB signals overview.....	166
5.6.2 SP Transmission of OOB signals.....	166
5.6.3 Receiver detection of OOB signals.....	167
5.6.4 SATA port selection signal.....	169
5.6.5 Phy power conditions.....	170
5.7 Phy reset sequences.....	170
5.7.1 Phy reset sequences overview.....	170
5.7.2 SATA phy reset sequence.....	171
5.7.3 SAS to SATA phy reset sequence.....	172
5.7.4 SAS to SAS phy reset sequence.....	174
5.7.5 Phy reset sequence after devices are attached.....	196
5.8 Phy power condition sequences.....	197
5.8.1 Transitioning from the active phy power condition to a low phy power condition.....	197
5.8.2 Transitioning from a low phy power condition to the active phy power condition.....	197
5.8.3 Events during low phy power condition.....	198
5.9 SP (phy layer) state machine.....	200
5.9.1 SP state machine overview.....	200
5.9.2 SP transmitter and receiver.....	202
5.9.3 OOB sequence states.....	204
5.9.4 SAS speed negotiation states.....	209
5.9.5 SAS phy power conditions states.....	219
5.9.6 SATA host emulation states.....	222
5.9.7 SATA port selector state SP25:SATA_PortSel.....	227
5.9.8 SATA spinup hold state SP26:SATA_SpinupHold.....	228
5.10 SP_DWS (phy layer dword synchronization) state machine.....	228
5.10.1 SP_DWS state machine overview.....	228
5.10.2 SP_DWS receiver.....	230
5.10.3 SP_DWS0:AcquireSync state.....	231
5.10.4 SP_DWS1:Valid1 state.....	231
5.10.5 SP_DWS2:Valid2 state.....	232
5.10.6 SP_DWS3:SyncAcquired state.....	232
5.10.7 SP_DWS4:Lost1 state.....	232
5.10.8 SP_DWS5:Lost1Recovered state.....	233
5.10.9 SP_DWS6:Lost2 state.....	233
5.10.10 SP_DWS7:Lost2Recovered state.....	234
5.10.11 SP_DWS8:Lost3 state.....	234
5.10.12 SP_DWS9:Lost3Recovered state.....	234
5.11 Multiplexing.....	235
5.12 Spinup.....	236
6 Link layer.....	237
6.1 Link layer overview.....	237
6.2 Primitives.....	237
6.2.1 Primitives overview.....	237
6.2.2 Primitive summary.....	238
6.2.3 Primitive encodings.....	243
6.2.4 Primitive sequences.....	248
6.2.5 Deletable primitives.....	252
6.2.6 Primitives not specific to type of connections.....	256

6.2.7 Primitives used only inside SSP and SMP connections.....	262
6.2.8 Primitives used only inside STP connections and on SATA physical links.....	263
6.3 Physical link rate tolerance management	264
6.3.1 Physical link rate tolerance management overview	264
6.3.2 Phys originating dwords	266
6.3.3 Expander phys forwarding dwords.....	266
6.4 Idle physical links	267
6.5 CRC	267
6.5.1 CRC overview	267
6.5.2 CRC generation	269
6.5.3 CRC checking	271
6.6 Scrambling	272
6.7 Bit order of CRC and scrambler	274
6.8 Address frames	277
6.8.1 Address frames overview.....	277
6.8.2 IDENTIFY address frame.....	279
6.8.3 OPEN address frame	282
6.9 Link reset sequence	285
6.9.1 Link reset sequence overview.....	285
6.9.2 Expander device handling of link reset sequences	288
6.10 SL_IR (link layer identification and hard reset) state machines.....	288
6.10.1 SL_IR state machines overview.....	288
6.10.2 SL_IR transmitter and receiver	290
6.10.3 SL_IR_TIR (transmit IDENTIFY or HARD_RESET) state machine	290
6.10.4 SL_IR_RIF (receive IDENTIFY address frame) state machine.....	292
6.10.5 SL_IR_IRC (identification and hard reset control) state machine	293
6.11 Entering a low phy power condition	294
6.12 SAS domain changes (Broadcast (Change) usage)	295
6.13 Connections	296
6.13.1 Connections overview	296
6.13.2 Opening a connection	296
6.13.3 Arbitration fairness	298
6.13.4 Arbitration inside an expander device	300
6.13.5 BREAK handling	304
6.13.6 Aborting a connection request	305
6.13.7 Closing a connection.....	307
6.13.8 Breaking a connection.....	308
6.14 Rate matching	309
6.15 SL (link layer for SAS logical phys) state machines.....	312
6.15.1 SL state machines overview	312
6.15.2 SL transmitter and receiver	315
6.15.3 SL_RA (receive OPEN address frame) state machine	317
6.15.4 SL_CC (connection control) state machine.....	317
6.16 XL (link layer for expander logical phys) state machine.....	330
6.16.1 XL state machine overview	330
6.16.2 XL transmitter and receiver	335
6.16.3 XL0:Idle state	336
6.16.4 XL1:Request_Path state	338
6.16.5 XL2:Request_Open state.....	340
6.16.6 XL3:Open_Confirm_Wait state	340
6.16.7 XL4:Open_Reject state	342
6.16.8 XL5:Forward_Open state.....	342
6.16.9 XL6:Open_Response_Wait state.....	343
6.16.10 XL7:Connected state.....	344

6.16.11 XL8:Close_Wait state.....	345
6.16.12 XL9:Break state.....	346
6.16.13 XL10:Break_Wait state	347
6.16.14 XL11:PS_Request state.....	347
6.16.15 XL12:PS_Quiet state.....	348
6.17 SSP link layer	349
6.17.1 Opening an SSP connection	349
6.17.2 Full duplex.....	349
6.17.3 SSP frame transmission and reception.....	350
6.17.4 SSP flow control.....	350
6.17.5 Interlocked frames.....	351
6.17.6 Breaking an SSP connection	353
6.17.7 Closing an SSP connection.....	353
6.17.8 SSP (link layer for SSP phys) state machines	354
6.18 STP link layer	365
6.18.1 STP frame transmission and reception	365
6.18.2 STP flow control	366
6.18.3 Continued primitive sequence	370
6.18.4 Affiliations.....	371
6.18.5 Opening an STP connection	373
6.18.6 Closing an STP connection.....	375
6.18.7 STP connection management examples	376
6.18.8 STP (link layer for STP phys) state machines.....	379
6.18.9 SMP target port support.....	379
6.19 SMP link layer	379
6.19.1 SMP frame transmission and reception	379
6.19.2 SMP flow control	379
6.19.3 Opening an SMP connection	379
6.19.4 Closing an SMP connection.....	380
6.19.5 SMP (link layer for SMP phys) state machines	380
7 Port layer.....	385
7.1 Port layer overview.....	385
7.2 PL (port layer) state machines.....	385
7.2.1 PL state machines overview	385
7.2.2 PL_OC (port layer overall control) state machine	387
7.2.3 PL_PM (port layer phy manager) state machine.....	398
8 Transport layer.....	409
8.1 Transport layer overview.....	409
8.2 SSP transport layer	410
8.2.1 SSP frame format.....	410
8.2.2 Information units.....	414
8.2.3 Sequences of SSP frames	424
8.2.4 SSP transport layer handling of link layer errors.....	427
8.2.5 SSP transport layer error handling summary.....	432
8.2.6 ST (transport layer for SSP ports) state machines.....	434
8.3 STP transport layer	470
8.3.1 Initial FIS	470
8.3.2 BIST Activate FIS.....	470
8.3.3 TT (transport layer for STP ports) state machines	470
8.4 SMP transport layer	471
8.4.1 SMP transport layer overview	471
8.4.2 SMP_REQUEST frame.....	472
8.4.3 SMP_RESPONSE frame	472
8.4.4 Sequence of SMP frames	473

8.4.5 MT (transport layer for SMP ports) state machines	473
9 Application layer	478
9.1 Application layer overview	478
9.2 SCSI application layer	478
9.2.1 SCSI transport protocol services	478
9.2.2 Application client error handling	490
9.2.3 Device server error handling	491
9.2.4 Task router and task manager error handling	492
9.2.5 SCSI transport protocol event notifications	492
9.2.6 SCSI commands	492
9.2.7 SCSI mode parameters	493
9.2.8 SCSI log parameters	504
9.2.9 SCSI diagnostic parameters	507
9.2.10 SCSI power conditions	514
9.2.11 SCSI vital product data (VPD)	523
9.3 ATA application layer	526
9.4 Management application layer	526
9.4.1 READY LED signal behavior	526
9.4.2 Management protocol services	527
9.4.3 SMP functions	528
Annex A (normative) Jitter tolerance patterns	641
Annex B (informative) SAS to SAS phy reset sequence examples	649
Annex C (informative) CRC	654
Annex D (informative) SAS address hashing	658
Annex E (informative) Scrambling	665
Annex F (informative) ATA architectural notes	669
Annex G (informative) Minimum deletable primitive insertion rate summary	672
Annex H (informative) Zone permission configuration descriptor examples	673
Annex I (informative) SAS addressing	676
Annex J (informative) Expander device handling of connections	677
Annex K (informative) Primitive encoding	698
Bibliography	701

Figure 1 — Organization of this standard	20
Figure 2 — SCSI document relationships	21
Figure 3 — ATA document relationships	21
Figure 4 — Classes in class diagrams	51
Figure 5 — Association relationships in class diagrams	52
Figure 6 — Aggregation relationships in class diagrams	53
Figure 7 — Generalization relationships in class diagrams	54
Figure 8 — Dependency relationships in class diagrams	54
Figure 9 — Objects in object diagrams	55
Figure 10 — State machine conventions	56
Figure 11 — SAS Domain class diagram	61
Figure 12 — Phy class diagram	63
Figure 13 — SAS phy object diagram	64
Figure 14 — Expander phy object diagram	65
Figure 15 — Ports (narrow ports and wide ports)	67
Figure 16 — Port class diagram	68
Figure 17 — Port object diagram	69
Figure 18 — SAS devices	71
Figure 19 — Expander device	72
Figure 20 — Domains	73
Figure 21 — SAS domain bridging to ATA domains	73
Figure 22 — SAS domains bridging to ATA domains with SATA port selectors	74
Figure 23 — Devices spanning SAS domains	75
Figure 24 — Single expander device topology example	75
Figure 25 — Multiple expander device topologies	76
Figure 26 — Potential pathways	77
Figure 27 — Multiple connections on wide ports	79
Figure 28 — State machines for SAS devices	87
Figure 29 — State machines for expander devices	88
Figure 30 — Transmit data path in a SAS phy	89
Figure 31 — SSP link, port, SSP transport, and SCSI application layer state machines	90
Figure 32 — SMP link, port, SMP transport, and management application layer state machines	91
Figure 33 — STP link, port, STP transport, and ATA application layer state machines	92
Figure 34 — Transmit data path and state machines in an expander phy	93
Figure 35 — Receive data path in a SAS phy	94
Figure 36 — Receive data path in an expander phy	95
Figure 37 — State machines and SAS Device classes	96
Figure 38 — State machines and Expander Device classes	97
Figure 39 — Reset terminology	99
Figure 40 — Expander device model	102
Figure 41 — Expander device interfaces	105
Figure 42 — Expander device interface detail	106
Figure 43 — Phy-based expander route table	114
Figure 44 — Expander-based expander route table	115
Figure 45 — Level-order traversal example	117
Figure 46 — Examples of invalid topologies	121
Figure 47 — Externally configurable expander device and table-to-table attachment	124
Figure 48 — Expander route index levels example	125
Figure 49 — Expander route index order example	127
Figure 50 — Zoning example	129
Figure 51 — One ZPSDS example	129
Figure 52 — Zone manager location examples	130
Figure 53 — Three ZPSDSes example	131
Figure 54 — Extending a ZPSDS example	133
Figure 55 — Overtaking a ZPSDS example	134
Figure 56 — Zoning expander route table	139
Figure 57 — SAS bit transmission logic	165
Figure 58 — SAS bit reception logic	166

Figure 59 — OOB signal transmission	167
Figure 60 — OOB signal detection	169
Figure 61 — SATA OOB sequence	171
Figure 62 — SATA speed negotiation sequence	172
Figure 63 — SAS to SATA OOB sequence	173
Figure 64 — SAS to SAS OOB sequence	175
Figure 65 — SNW-1, SNW-2, and Final-SNW	178
Figure 66 — SNW-3	180
Figure 67 — Train-SNW	185
Figure 68 — SAS speed negotiation sequence SNW flowchart	187
Figure 69 — SAS speed negotiation sequence (both phys SNW-1 through SNW-3)	189
Figure 70 — SAS speed negotiation sequence (phy A: SNW-1 through SNW-3, phy B: SNW-2 only)	190
Figure 71 — SAS speed negotiation sequence (phy A: SNW-3 only, phy B: SNW-1 only)	191
Figure 72 — SAS speed negotiation sequence - phy reset problem in Final-SNW	192
Figure 73 — SAS speed negotiation sequence - phy reset problem in SNW-3	193
Figure 74 — SAS speed negotiation sequence - phy reset problem in Train-SNW	194
Figure 75 — SAS speed negotiation sequence - multiple Train-SNWs	195
Figure 76 — Hot-plug and the phy reset sequence	197
Figure 77 — Transition to active phy power condition	198
Figure 78 — Hot plug and low phy power condition	199
Figure 79 — SP (phy layer) state machine - OOB sequence states	204
Figure 80 — SP (phy layer) state machine - SAS speed negotiation states	210
Figure 81 — SP (phy layer) state machine - SAS speed negotiation states for SNW-3 and Train-SNW	211
Figure 82 — SP (phy layer) state machine - SAS phy power condition states	220
Figure 83 — SP (phy layer) state machine - SATA host emulation states	223
Figure 84 — SP (phy layer) state machine – SATA port selector state	227
Figure 85 — SP (phy layer) state machine - SATA spinup hold state	228
Figure 86 — SP_DWS (phy layer dword synchronization) state machine	230
Figure 87 — Multiplexing disabled	235
Figure 88 — Multiplexing enabled	235
Figure 89 — Transmitting a repeated primitive sequence	249
Figure 90 — Receiving a repeated primitive sequence	249
Figure 91 — Extended primitive sequences	250
Figure 92 — Triple primitive sequences	251
Figure 93 — Redundant primitive sequences	252
Figure 94 — Elasticity buffer	265
Figure 95 — Address frame, SSP frame, and SMP frame CRC bit ordering	270
Figure 96 — STP frame CRC bit ordering	271
Figure 97 — Transmit path bit ordering	274
Figure 98 — Receive path bit ordering	275
Figure 99 — STP transmit path bit ordering	276
Figure 100 — STP receive path bit ordering	277
Figure 101 — Address frame transmission	277
Figure 102 — Identification sequence	286
Figure 103 — Hard reset sequence	287
Figure 104 — SL_IR (link layer identification and hard reset) state machines	289
Figure 105 — Transitioning from the active phy power condition to a low phy power condition	294
Figure 106 — Example simultaneous connection recommendations for wide ports	297
Figure 107 — Aborting a connection request with a BREAK primitive sequence	306
Figure 108 — Connection request timeout example	307
Figure 109 — Closing a connection example	308
Figure 110 — Rate matching example	311
Figure 111 — SL (link layer for SAS logical phys) state machines (part 1)	313
Figure 112 — SL (link layer for SAS logical phys) state machines (part 2)	314
Figure 113 — SL (link layer for SAS logical phys) state machines (part 3)	315
Figure 114 — XL (link layer for expander logical phys) state machine (part 1)	332
Figure 115 — XL (link layer for expander logical phys) state machine (part 2)	333
Figure 116 — XL (link layer for expander logical phys) state machine (part 3)	334

Figure 117 — SSP frame transmission	350
Figure 118 — Interlocked frames	352
Figure 119 — Non-interlocked frames with the same initiator port transfer tags	352
Figure 120 — Non-interlocked frames with different initiator port transfer tags	353
Figure 121 — Closing an SSP connection example	354
Figure 122 — SSP (link layer for SSP phys) state machines (part 1 - frame transmission)	356
Figure 123 — SSP (link layer for SSP phys) state machines (part 2 - frame reception)	357
Figure 124 — STP frame transmission	365
Figure 125 — STP flow control	369
Figure 126 — Transmitting a continued primitive sequence	371
Figure 127 — Receiving a continued primitive sequence	371
Figure 128 — Example simultaneous connection recommendations for an expander device	375
Figure 129 — STP initiator port opening an STP connection	377
Figure 130 — STP target port opening an STP connection	378
Figure 131 — SMP frame transmission	379
Figure 132 — SMP_IP (link layer for SMP initiator phys) state machine	381
Figure 133 — SMP_TP (link layer for SMP target phys) state machine	383
Figure 134 — Port layer examples	386
Figure 135 — PL_OC (port layer overall control) state machine	389
Figure 136 — PL_PM (port layer phy manager) state machine (part 1)	400
Figure 137 — PL_PM (port layer phy manager) state machine (part 2)	401
Figure 138 — Task management function sequence of SSP frames	424
Figure 139 — Non-data command sequence of SSP frames	425
Figure 140 — Write command sequence of SSP frames	425
Figure 141 — Read command sequence of SSP frames	426
Figure 142 — Bidirectional command sequence of SSP frames	427
Figure 143 — ST_I (transport layer for SSP initiator ports) state machines	436
Figure 144 — ST_T (transport layer for SSP target ports) state machines	452
Figure 145 — Sequence of SMP frames	473
Figure 146 — MT_IP (transport layer for SMP initiator ports) state machine	474
Figure 147 — MT_TP (transport layer for SMP target ports) state machine	476
Figure 148 — SA_PC (SCSI application layer power condition) state machine for SAS	516
Figure A.1 — CJTPAT pre-scrambling	642
Figure B.1 — SAS speed negotiation sequence (phy A:SNW-1 only, phy B:SNW-1 only)	649
Figure B.2 — SAS speed negotiation sequence (phy A:SNW-1, SNW-2, phy B:SNW-1, SNW-2)	650
Figure B.3 — SAS speed negotiation sequence (phy A:SNW-1, SNW-2, SNW-3, phy B:SNW-1, SNW-2)	651
Figure B.4 — SAS speed negotiation sequence (phy A:SNW-2, SNW-3, phy B:SNW-1, SNW-2)	652
Figure B.5 — SAS speed negotiation sequence (phy A:SNW-1 only, phy B:SNW-2 only)	653
Figure C.1 — CRC generator example	654
Figure C.2 — CRC checker example	654
Figure D.1 — BCH(69, 39, 9) code generator	659
Figure E.1 — Scrambler	665
Figure J.1 — Example topology	677
Figure J.2 — Connection request - OPEN_ACCEPT	679
Figure J.3 — Connection request - OPEN_REJECT by end device	680
Figure J.4 — Connection request - OPEN_REJECT by expander device	681
Figure J.5 — Connection request - arbitration lost	682
Figure J.6 — Connection request - backoff and retry	683
Figure J.7 — Connection request - backoff and reverse path	684
Figure J.8 — Connection close - single step	685
Figure J.9 — Connection close - simultaneous	686
Figure J.10 — BREAK handling during path arbitration when the BREAK_REPLY method is disabled	687
Figure J.11 — BREAK handling during a connection when the BREAK_REPLY method is disabled	688
Figure J.12 — BREAK handling during path arbitration when the BREAK_REPLY method is enabled	689
Figure J.13 — BREAK handling during a connection when the BREAK_REPLY method is enabled	690
Figure J.14 — STP connection - originated by STP initiator port	691
Figure J.15 — STP connection - originated by STP target port in an STP/SATA bridge	692
Figure J.16 — STP connection close - originated by STP initiator port	693

Figure J.17 — STP connection close - originated by STP target port in an STP/SATA bridge	694
Figure J.18 — XL1:Request_Path to XL5:Forward_Open transition	695
Figure J.19 — Partial pathway recovery	696

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Table 1 — Standards bodies.....	23
Table 2 — Numbering conventions.....	50
Table 3 — Multiplicity notation in class diagrams	52
Table 4 — Data dword containing a value	58
Table 5 — Data dword containing four one-byte fields	58
Table 6 — Logical links	66
Table 7 — Broadcast types.....	80
Table 8 — Names and identifiers.....	82
Table 9 — SAM-5 attribute mapping.....	82
Table 10 — NAA IEEE Registered format	82
Table 11 — NAA Locally Assigned format.....	83
Table 12 — Hashed SAS address code parameters	84
Table 13 — Device name created from the IDENTIFY (PACKET) DEVICE world wide name.....	85
Table 14 — Expander logical phy to ECM requests	107
Table 15 — Expander logical phy to ECM responses.....	107
Table 16 — ECM to expander logical phy confirmations	108
Table 17 — Expander logical phy to ECR to expander logical phy requests and indications.....	109
Table 18 — Expander logical phy to ECR to expander logical phy responses and confirmations.....	109
Table 19 — Expander logical phy to BPP requests	110
Table 20 — BPP to expander logical phy indications	111
Table 21 — Routing attributes and routing methods.....	112
Table 22 — Expander route table types.....	113
Table 23 — Expander route table levels for externally configurable expander device R phy A.....	125
Table 24 — Expander route table levels for externally configurable expander device N.....	126
Table 25 — Expander route entries for externally configurable expander device E0 phy 1	127
Table 26 — Expander route entries for externally configurable expander device F phy 0.....	128
Table 27 — Zone manager password.....	132
Table 28 — Zone phy information.....	134
Table 29 — Zone phy information usage	135
Table 30 — Zone groups	137
Table 31 — Zone permission table	138
Table 32 — Zone permission table granting minimal permissions.....	138
Table 33 — Source zone group determination	140
Table 34 — Destination zone group determination.....	140
Table 35 — Zone phy information fields after a link reset sequence	141
Table 36 — Conditions that cause the ZONE GROUP field to be reset if the ZONE GROUP PERSISTENT bit is set to zero	142
Table 37 — PHY EVENT SOURCE field	151
Table 38 — Bit designations	155
Table 39 — Conversion from byte notation to character name example	156
Table 40 — Data characters	158
Table 41 — Control characters	163
Table 42 — Control character usage	163
Table 43 — Delayed code violation example.....	164
Table 44 — Phy reset sequence timing specifications.....	171
Table 45 — SATA speed negotiation sequence timing specifications	172
Table 46 — SAS speed negotiation sequence timing specifications	177
Table 47 — SNW rates used in SNW-1, SNW-2, and Final-SNW	179
Table 48 — SNW-3 phy capabilities bit.....	180
Table 49 — SNW-3 phy capabilities	181
Table 50 — Requested logical link rate	182
Table 51 — Multiplexing negotiation	182
Table 52 — Supported settings bit priorities	183
Table 53 — Example SNW-3 phy capabilities values.....	184
Table 54 — Training patterns.....	185
Table 55 — SP state machine timers.....	202
Table 56 — Messages to SP transmitter and SP receiver at start of RCDT.....	212
Table 57 — SP_DWS state machine timers	229

Table 58 — Primitive format.....	237
Table 59 — Deletable primitives	238
Table 60 — Primitives not specific to type of connection.....	239
Table 61 — Primitives used inside SSP and SMP connections.....	241
Table 62 — Primitives used inside STP connections and on SATA physical links	242
Table 63 — Primitive encoding for deletable primitives	243
Table 64 — Primitive encoding for primitives not specific to type of connection.....	244
Table 65 — Primitive encoding for primitives used only inside SSP and SMP connections.....	246
Table 66 — Primitive encoding for primitives used only inside STP connections and only on SATA physical links	247
Table 67 — Primitive sequences	248
Table 68 — ALIGN primitives.....	253
Table 69 — MUX primitives	254
Table 70 — NOTIFY primitives	254
Table 71 — AIP primitives.....	256
Table 72 — BROADCAST primitives	257
Table 73 — CLOSE primitives	258
Table 74 — Abandon-class OPEN_REJECT primitives	259
Table 75 — Retry-class OPEN_REJECT primitives	260
Table 76 — PS_REQ primitives.....	261
Table 77 — DONE primitives	262
Table 78 — NAK primitives	263
Table 79 — RRDY primitives	263
Table 80 — Physical link rate tolerance management deletable primitive insertion requirement.....	266
Table 81 — CRC notation and definitions.....	268
Table 82 — Scrambling for different data dword types.....	272
Table 83 — Address frame format.....	278
Table 84 — ADDRESS FRAME TYPE field	278
Table 85 — IDENTIFY address frame format	279
Table 86 — DEVICE TYPE field.....	280
Table 87 — REASON field.....	280
Table 88 — OPEN address frame format	282
Table 89 — PROTOCOL field.....	283
Table 90 — FEATURES field.....	283
Table 91 — CONNECTION RATE field.....	283
Table 92 — ARBITRATION WAIT TIME field	285
Table 93 — SL_IR_IRC state machine timers	288
Table 94 — PS_ACK pattern.....	295
Table 95 — Connection Results of a connection request.....	298
Table 96 — Arbitration priority for OPEN address frames passing on a logical link	299
Table 97 — Arbitration priority for a Request Path request in the ECM.....	302
Table 98 — Pathway recovery priority	304
Table 99 — Results of aborting a connection request	305
Table 100 — Results of closing a connection	307
Table 101 — Results of breaking a connection	309
Table 102 — Rate matching deletable primitive insertion requirements.....	310
Table 103 — SL_CC state machine timers.....	319
Table 104 — SL_CC state machine variables	319
Table 105 — OPEN_REJECT Received message to Open Failed confirmation mapping.....	322
Table 106 — XL state machine timers.....	331
Table 107 — XL state machine variable	331
Table 108 — SSP frame interlock requirements.....	351
Table 109 — SSP state machines timers	355
Table 110 — STP link layer differences from SATA link layer during an STP connection.....	366
Table 111 — Affiliation policies.....	372
Table 112 — Affiliation context relative identifier example.....	373
Table 113 — PL_OC state machine timers.....	388
Table 114 — Confirmations from Unable To Connect messages.....	395

Table 115 — PL_PM state machine timers.....	399
Table 116 — Messages from Open Failed confirmations	403
Table 117 — SSP frame format	410
Table 118 — FRAME TYPE field	411
Table 119 — TLR CONTROL field for COMMAND frames	412
Table 120 — COMMAND frame - Command information unit	414
Table 121 — TASK ATTRIBUTE field	415
Table 122 — TASK frame - Task Management Function information unit.....	416
Table 123 — TASK MANAGEMENT FUNCTION field	417
Table 124 — XFER_RDY frame - Transfer Ready information unit.....	419
Table 125 — DATA frame - Data information unit	420
Table 126 — RESPONSE frame - Response information unit	421
Table 127 — DATAPRES field	422
Table 128 — RESPONSE DATA field	423
Table 129 — RESPONSE CODE field.....	423
Table 130 — Sequences of SSP frames	424
Table 131 — Confirmations sent to the SCSI application layer if a frame transmission error or reception error occurs	440
Table 132 — ST_ITS state machine variables.....	441
Table 133 — ST_ITS state machine arguments.....	442
Table 134 — Messages sent to the ST_IFR state machine.....	444
Table 135 — Transmission Complete messages for XFER_RDY frame verification failures	445
Table 136 — Reception Complete messages for read DATA frame verification failures.....	450
Table 137 — ST_T state machine timers.....	451
Table 138 — Task Management Function Executed Service Response argument mapping to Request (Send Transport Response) service response argument	456
Table 139 — Confirmations sent to the SCSI application layer	457
Table 140 — ST_TTS state machine variables	459
Table 141 — ST_TTS state machine arguments.....	459
Table 142 — Messages sent to the ST_TFR state machine	463
Table 143 — Additional messages sent to the ST_TFR state machine.....	464
Table 144 — Reception Complete message for write DATA frame verification failures	467
Table 145 — Request argument to RESPONSE frame response data field mapping.....	469
Table 146 — SMP frame format	471
Table 147 — SMP FRAME TYPE field	471
Table 148 — SMP_REQUEST frame format.....	472
Table 149 — SMP_RESPONSE frame format	472
Table 150 — MT_TP time limits	476
Table 151 — Execute Command procedure call transport protocol services	478
Table 152 — Task management function procedure call transport protocol services	479
Table 153 — Send SCSI Command transport protocol service arguments.....	480
Table 154 — SCSI Command Received transport protocol service arguments	481
Table 155 — Send Command Complete transport protocol service arguments.....	482
Table 156 — Command Complete Received transport protocol service arguments	483
Table 157 — Send Data-In transport protocol service arguments	484
Table 158 — Data-In Delivered transport protocol service arguments	484
Table 159 — Receive Data-Out transport protocol service arguments.....	485
Table 160 — Data-Out Received transport protocol service arguments.....	485
Table 161 — Terminate Data Transfer transport protocol service arguments	486
Table 162 — Data Transfer Terminated transport protocol service arguments	486
Table 163 — Send Task Management Request transport protocol service arguments.....	487
Table 164 — Task Management Request Received transport protocol service arguments	488
Table 165 — Task Management Function Executed transport protocol service arguments.....	489
Table 166 — Received Task Management Function Executed transport protocol service arguments.....	490
Table 167 — Delivery Result to additional sense code mapping.....	491
Table 168 — SCSI transport protocol events.....	492
Table 169 — SSP target port mode pages	493
Table 170 — Disconnect-Reconnect mode page for SSP	494

Table 171 — Protocol-Specific Logical Unit mode page for SAS SSP	496
Table 172 — Protocol-Specific Port mode page for SAS SSP	497
Table 173 — I_T NEXUS LOSS TIME field	498
Table 174 — Phy Control And Discover mode page	499
Table 175 — SAS phy mode descriptor	500
Table 176 — Shared Port Control mode page	501
Table 177 — Enhanced Phy Control mode page	502
Table 178 — Enhanced phy control mode descriptor	503
Table 179 — Protocol-Specific Port log page for SAS SSP	504
Table 180 — Protocol-Specific Port log parameter for SAS	505
Table 181 — Parameter control byte in the Protocol-Specific Port log parameter for SAS	505
Table 182 — SAS phy log descriptor	506
Table 183 — SSP target port diagnostic pages	507
Table 184 — Diagnostic pages affected by zoning	508
Table 185 — Protocol-Specific diagnostic page for SAS SSP	509
Table 186 — PHY TEST FUNCTION field	510
Table 187 — PHY TEST PATTERN field	511
Table 188 — PHY TEST FUNCTION SSC field	512
Table 189 — PHY TEST FUNCTION PHYSICAL LINK RATE field	512
Table 190 — PHY TEST PATTERN DWORDS CONTROL field	512
Table 191 — TWO_DWORDS phy test pattern examples	513
Table 192 — Summary of states in the SA_PC state machine	515
Table 193 — VPD pages with special requirements for SAS SSP	523
Table 194 — Device Identification VPD page designation descriptors for the SAS target port	524
Table 195 — Device Identification VPD page designation descriptors for the SAS target device	524
Table 196 — Protocol-Specific Logical Unit Information VPD page for SAS SSP	525
Table 197 — Logical unit information descriptor for SAS SSP	526
Table 198 — READY LED signal behavior	527
Table 199 — SMP functions (FUNCTION field)	528
Table 200 — SMP request frame format	530
Table 201 — SMP response frame format	532
Table 202 — FUNCTION RESULT field	533
Table 203 — Function result priority	536
Table 204 — REPORT GENERAL request	540
Table 205 — REPORT GENERAL response	541
Table 206 — NUMBER OF ZONE GROUPS field	545
Table 207 — REPORT MANUFACTURER INFORMATION request	547
Table 208 — REPORT MANUFACTURER INFORMATION response	548
Table 209 — REPORT SELF-CONFIGURATION STATUS request	550
Table 210 — REPORT SELF-CONFIGURATION STATUS response	551
Table 211 — Self-configuration status descriptor	553
Table 212 — STATUS TYPE field	553
Table 213 — REPORT ZONE PERMISSION TABLE request	555
Table 214 — REPORT TYPE field	555
Table 215 — REPORT ZONE PERMISSION TABLE response	556
Table 216 — Zone permission descriptors	557
Table 217 — Zone permission descriptor for a source zone group (i.e., s) with 128 zone groups	557
Table 218 — Zone permission descriptor for a source zone group (i.e., s) with 256 zone groups	558
Table 219 — Zone permission descriptor bit requirements	558
Table 220 — REPORT ZONE MANAGER PASSWORD request	559
Table 221 — REPORT TYPE field	559
Table 222 — REPORT ZONE MANAGER PASSWORD response	560
Table 223 — REPORT BROADCAST request	561
Table 224 — REPORT BROADCAST response	562
Table 225 — Broadcast descriptor	563
Table 226 — BROADCAST REASON field for originated Broadcasts	564
Table 227 — DISCOVER request	565
Table 228 — DISCOVER response	566

Table 229 — ATTACHED DEVICE TYPE field	569
Table 230 — NEGOTIATED LOGICAL LINK RATE field and NEGOTIATED PHYSICAL LINK RATE field	570
Table 231 — NEGOTIATED LOGICAL LINK RATE field	570
Table 232 — ATTACHED SATA PORT SELECTOR and ATTACHED SATA DEVICE bits	571
Table 233 — PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field	574
Table 234 — The HARDWARE MINIMUM PHYSICAL LINK RATE field and the HARDWARE MAXIMUM PHYSICAL LINK RATE field	574
Table 235 — ROUTING ATTRIBUTE field	575
Table 236 — PHY POWER CONDITION field	575
Table 237 — ATTACHED DEVICE NAME field	576
Table 238 — SELF-CONFIGURATION STATUS field	577
Table 239 — SELF-CONFIGURATION LEVELS COMPLETED field	578
Table 240 — REPORT PHY ERROR LOG request	580
Table 241 — REPORT PHY ERROR LOG response	581
Table 242 — REPORT PHY SATA request	582
Table 243 — REPORT PHY SATA response	584
Table 244 — REPORT ROUTE INFORMATION request	586
Table 245 — REPORT ROUTE INFORMATION response	587
Table 246 — REPORT PHY EVENT request	589
Table 247 — REPORT PHY EVENT response	590
Table 248 — Phy event descriptor	591
Table 249 — DISCOVER LIST request	592
Table 250 — PHY FILTER field	593
Table 251 — DESCRIPTOR TYPE field	593
Table 252 — DISCOVER LIST response	594
Table 253 — SHORT FORMAT descriptor	596
Table 254 — REPORT PHY EVENT LIST request	597
Table 255 — REPORT PHY EVENT LIST response	598
Table 256 — Phy event list descriptor	599
Table 257 — REPORT EXPANDER ROUTE TABLE LIST request	600
Table 258 — REPORT EXPANDER ROUTE TABLE LIST response	602
Table 259 — Expander route table descriptor	604
Table 260 — CONFIGURE GENERAL request	605
Table 261 — STP SMP I_T NEXUS LOSS TIME field	606
Table 262 — CONFIGURE GENERAL response	607
Table 263 — ENABLE DISABLE ZONING request	608
Table 264 — SAVE field	608
Table 265 — ENABLE DISABLE ZONING field	609
Table 266 — ENABLE DISABLE ZONING response	609
Table 267 — ZONED BROADCAST request	610
Table 268 — BROADCAST TYPE field	611
Table 269 — ZONED BROADCAST response	611
Table 270 — ZONE LOCK request	612
Table 271 — ZONE LOCK response	613
Table 272 — ZONE ACTIVATE request	614
Table 273 — ZONE ACTIVATE response	614
Table 274 — ZONE UNLOCK request	615
Table 275 — ZONE UNLOCK response	616
Table 276 — CONFIGURE ZONE MANAGER PASSWORD request	617
Table 277 — SAVE field	617
Table 278 — CONFIGURE ZONE MANAGER PASSWORD response	618
Table 279 — CONFIGURE ZONE PHY INFORMATION request	619
Table 280 — SAVE field	620
Table 281 — Zone phy configuration descriptor	620
Table 282 — CONFIGURE ZONE PHY INFORMATION response	621
Table 283 — CONFIGURE ZONE PERMISSION TABLE request	622
Table 284 — NUMBER OF ZONE GROUPS field	623

Table 285 — SAVE field	623
Table 286 — Zone permission configuration descriptors	623
Table 287 — Zone permission configuration descriptor for source zone group for 128 zone groups	624
Table 288 — Zone permission configuration descriptor for source zone group for 256 zone groups	624
Table 289 — Zone permission configuration descriptor bit requirements	624
Table 290 — CONFIGURE ZONE PERMISSION TABLE response	625
Table 291 — CONFIGURE ROUTE INFORMATION request	626
Table 292 — CONFIGURE ROUTE INFORMATION response	627
Table 293 — PHY CONTROL request	628
Table 294 — PHY OPERATION field	630
Table 295 — PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field	632
Table 296 — ENABLE SAS SLUMBER field	633
Table 297 — ENABLE SAS PARTIAL field	633
Table 298 — ENABLE SATA SLUMBER field	633
Table 299 — ENABLE SATA PARTIAL field	634
Table 300 — PHY CONTROL response	634
Table 301 — PHY TEST FUNCTION request	635
Table 302 — PHY TEST FUNCTION field	636
Table 303 — PHY TEST FUNCTION PHYSICAL LINK RATE field	637
Table 304 — PHY TEST FUNCTION response	637
Table 305 — CONFIGURE PHY EVENT request	638
Table 306 — Phy event configuration descriptor	639
Table 307 — CONFIGURE PHY EVENT response	639
Table A.1 — JTPAT for RD+ and RD-	641
Table A.2 — CJTPAT	643
Table A.3 — CJTPAT with fixed content	647
Table C.1 — CRC examples	657
Table D.1 — Monte-Carlo simulation results	659
Table D.2 — Hash results for simple SAS addresses	661
Table D.3 — Hash results for realistic SAS addresses	662
Table D.4 — Hash results for a walking ones pattern	663
Table D.5 — Hash results for a walking zeros pattern	664
Table E.1 — Scrambler examples	667
Table E.2 — Initial scrambler output	668
Table G.1 — Minimum deletable primitive insertion rate examples	672
Table H.1 — Zone permission table example initial value	673
Table H.2 — CONFIGURE ZONE PERMISSION TABLE request example	674
Table H.3 — Zone permission table after processing first zone permission configuration descriptor	675
Table H.4 — Zone permission table after processing second zone permission configuration descriptor	675
Table J.1 — Column descriptions for connection examples	678
Table K.1 — Primitives with Hamming distance of at least 7	698
Table K.2 — Primitives without Hamming distance of 7	700

INFORMATION TECHNOLOGY - SMALL COMPUTER SYSTEM INTERFACE (SCSI) - Part 261: SAS protocol layer (SPL)

FOREWORD

- 1) ISO (International Organization for Standardization) and IEC (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. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.
- 2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.
- 3) 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 and ISO member bodies.
- 4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC 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.
- 5) In order to promote international uniformity, IEC 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 publication and the corresponding national or regional publication should be clearly indicated in the latter.
- 6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.
- 7) All users should ensure that they have the latest edition of this publication.
- 8) 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 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.
- 9) 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.
- 10) Attention is drawn to the possibility that some of the elements of this International Standard 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 14776-261 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 IEC web site.

This International Standard has been approved by vote of the member bodies and the voting results may be obtained from the address given on the second title page.

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 publication using a colour printer.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

INTRODUCTION

General

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

This standard makes obsolete the following concepts from SAS-2:

- a) NOTIFY (RESERVED 2) primitive; and
- b) ATTACHED DEVICE TYPE field code (see 9.4.3.10) value 011b.

Figure 1 shows the organization of the layers of this standard.

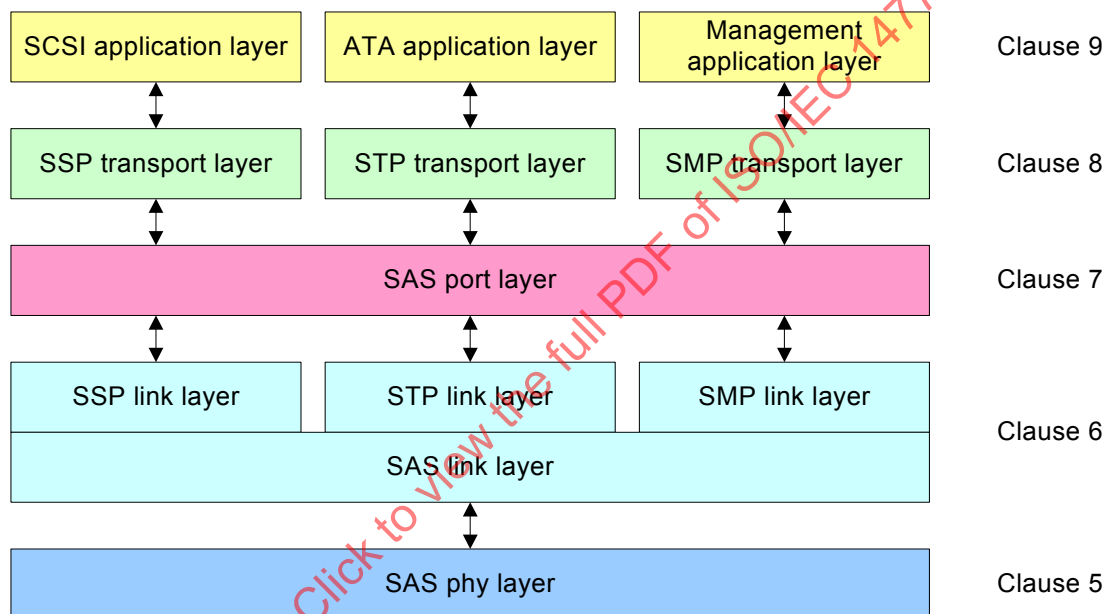


Figure 1 — Organization of this standard

SCSI standards family

Figure 2 shows the relationship of this standard to the other standards and related projects in the SCSI family of standards.

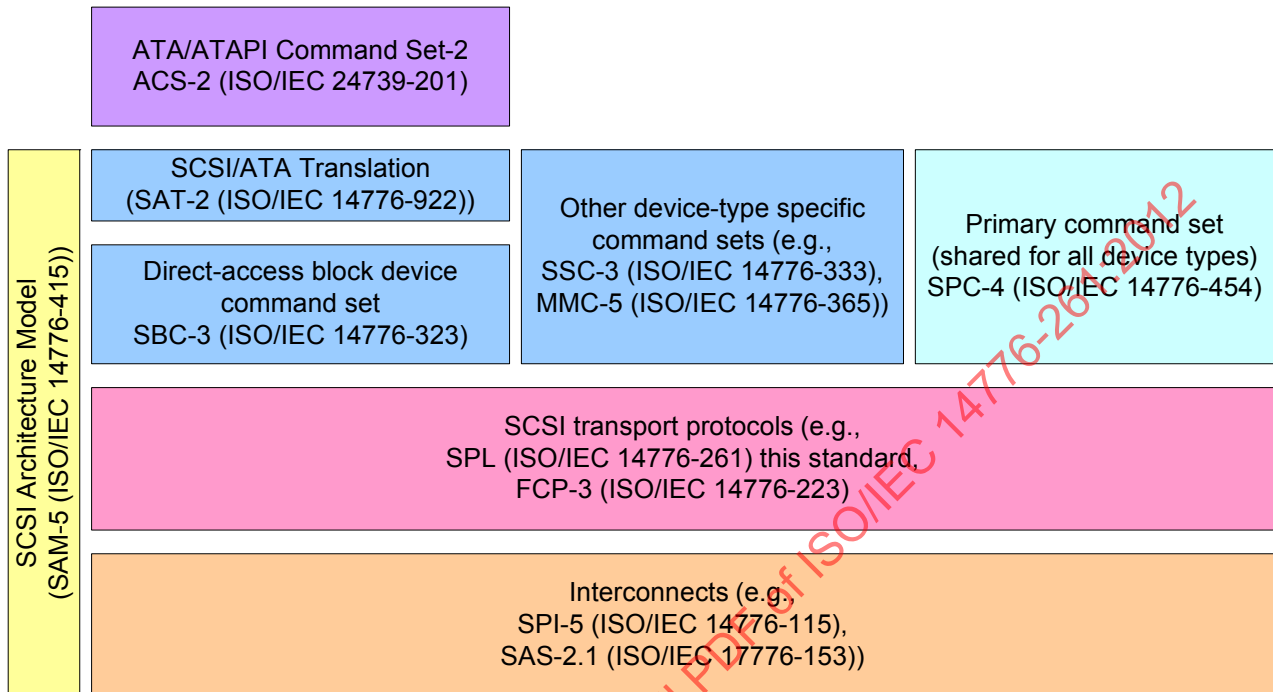


Figure 2 — SCSI document relationships

This standard also defines the rules for exchanging information between ATA hosts and ATA devices using the same serial interconnect. Other ATA transport protocol standards define the rules for exchanging information between ATA hosts and ATA devices using other interconnects.

Figure 3 shows the relationship of this standard to other standards and related projects in the ATA family of standards.

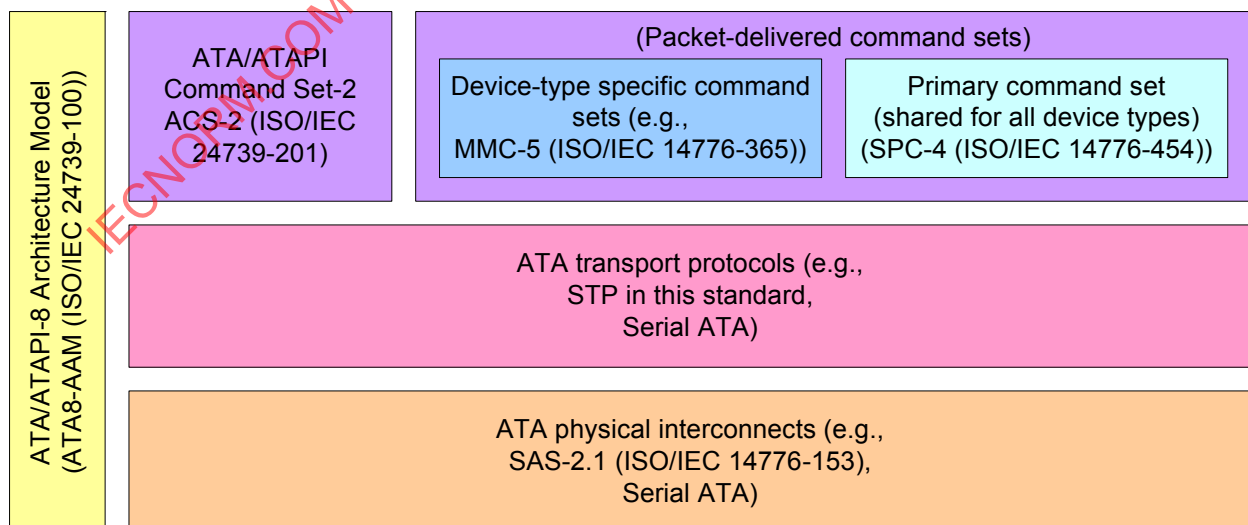


Figure 3 — ATA document relationships

Figure 2 and figure 3 show the general relationship of the documents to one another, and do not imply a relationship such as a hierarchy, protocol stack or system architecture.

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

INFORMATION TECHNOLOGY - SMALL COMPUTER SYSTEM INTERFACE (SCSI) - Part 261: SAS protocol layer (SPL)

1 Scope

This part of ISO/IEC 14776-261 defines the protocol layer of the Serial Attached SCSI (SAS) interconnect and three transport protocols that use the SAS interconnect:

- a) Serial SCSI Protocol (SSP): a mapping of SCSI supporting multiple initiators and targets;
- b) Serial ATA Tunneled Protocol (STP): a mapping of Serial ATA expanded to support multiple initiators and targets; and
- c) Serial Management Protocol (SMP): a management protocol.

2 Normative references

The following referenced documents are indispensable for the application 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.

The provisions of the referenced specifications other than ISO, IEC, ISO and ITU standards, as identified in this clause, are valid within the context of this International Standard. The reference to such a specification within this International Standard does not give it any further status within ISO or IEC. In particular, it does not give the referenced specification the status of an International Standard.

Additional availability contact information is provided below as needed.

Table 1 shows standards bodies and their web sites.

Table 1 — Standards bodies

Abbreviation	Standards body	Web site
ANSI®	American National Standards Institute	http://www.ansi.org
DIN	German Institute for Standardization	http://www.din.de
IEC®	International Electrotechnical Commission	http://www.iec.ch
IEEE®	Institute of Electrical and Electronics Engineers	http://www.ieee.org
INCITS	International Committee for Information Technology Standards	http://www.incits.org
ISO®	International Organization for Standardization	http://www.iso.ch
ITIC	Information Technology Industry Council	http://www.itic.org
JIS	Japanese Industrial Standards Committee	http://www.jisc.co.jp
T10	INCITS T10 SCSI storage interfaces	http://www.t10.org
T11	INCITS T11 Fibre Channel interfaces	http://www.t11.org
T13	INCITS T13 ATA storage interface	http://www.t13.org

NOTE 1 ANSI is a registered trademark of the American National Standards Institute.

NOTE 2 ISO is a registered trademark of the International Organization for Standardization.

NOTE 3 IEC is a registered trademark of the International Electrotechnical Commission.

NOTE 4 IEEE is a registered trademark of the Institute of Electrical Electronics Engineers, Inc.

ISO/IEC 14776-151, *Information technology – Small Computer System Interface (SCSI) – Part 151: Serial Attached SCSI - 1.1 (SAS-1.1)*

ISO/IEC 14776-153, *Information technology – Small Computer System Interface (SCSI) – Part 153: Serial Attached SCSI-2.1 (SAS-2.1) (T10/2125-D) (under consideration)*

INCITS 465-2010, *Information technology – Small Computer System Interface (SCSI) – Part 922: SCSI/ATA Translation-2 (SAT-2) (planned as ISO/IEC 14776-922)*

T10/1731-D, *Information technology – Small Computer System Interface (SCSI) – SCSI Primary Commands-4 (SPC-4) (planned as ISO/IEC 14776-454)*

T10/1799-D, *Information technology – Small Computer System Interface (SCSI) – SCSI Block Commands-3 (SBC-3) (planned as ISO/IEC 14776-323)*

T10/2149-D, *Information technology - Small Computer System Interface (SCSI) – SCSI Enclosure Services-3 (SES-3)(planned as ISO/IEC 14776-373)*

T10/2104-D, *Information technology – Small Computer System Interface (SCSI) – SCSI Architecture Model-5 (SAM-5) (planned as ISO/IEC 14776-415)*

T13/2015-D, *Information technology – AT Attachment – Part 201: ATA/ATAPI Command Set-2 (ACS-2) (planned as ISO/IEC 24739-201)*

T13/1700-D, *Information technology – AT Attachment – Part 100: ATA Attachment-8 ATA/ATAPI Architecture Model (ATA8-AAM) (planned as ISO/IEC 24739-100)*

NOTE 5 For more information on the current status of these documents, contact the INCITS Secretariat at 202-737-8888 (phone), 202-638-4922 (fax) or via Email at incits@itic.org. To obtain copies of these documents, contact Global Engineering at 15 Inverness Way, East Englewood, CO 80112-5704 at 303-792-2181 (phone), 800-854-7179 (phone), or 303-792-2192 (fax) or see <http://www.incits.org>.

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

Serial AT Attachment Revision 3.0 (SATA). 27-May-2009

NOTE 6 For information on the current status of Serial ATA documents, contact the Serial ATA International Organization (see <http://www.sata-io.org>).

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

3.1 Terms and definitions

For the purposes of this document the following terms and definitions apply.

3.1.1

8b10b coding

coding scheme that represents an 8-bit byte (i.e., a control byte or data byte) as a 10-bit character (i.e., a control character or data character) (see 5.2)

3.1.2

8b10b encoding

method of encoding an 8-bit byte (i.e., a control byte or data byte) into a 10-bit character (i.e., a control character or data character) (see 5.2)

3.1.3

10b8b decoding

method for decoding a 10-bit character (i.e., a control character or data character) into an 8-bit byte (i.e., a control byte or data byte) (see 5.2)

3.1.4

actual lock time

time during SNW-1, SNW-2, and Final-SNW (see 5.7.4.2.3.2) at which a phy locks on received ALIGN (0)s and/or ALIGN (1)s and changes from transmitting ALIGN (0) to ALIGN (1)

3.1.5

actual training time

time during Train-SNW (see 5.7.4.2.3.4) at which training (see 3.1.234) is complete

3.1.6

active phy power condition

normal power condition for a SAS phy or expander phy (see 4.10.2)

3.1.7

active zone manager

zone manager that locks a zoning expander device (see 4.9.6)

3.1.8

affiliation

STP target port state of limiting acceptance of connection requests to those from one or more STP initiator ports (see 6.18.4)

3.1.9

affiliation context

set of registers maintained by an STP target port for an STP initiator port holding an affiliation (see 6.18.4)

3.1.10

aggregation

when referring to classes (see 3.1.26), a form of association (see 3.1.12) that defines a whole-part relationship between the whole (i.e., aggregate) class and its parts (see 3.5)

3.1.11

application client

object that is the source of SCSI commands and task management function requests (see SAM-5), ATA commands (see ATA8-AAM), or management function requests (see 4.1.5)

3.1.12

association

when referring to classes (see 3.1.27), a relationship between two or more classes that specifies connections among their objects (see 3.1.124) (i.e., a relationship that specifies that objects of one class are connected to objects of another class) (see 3.5)

3.1.13

attached SAS address

SAS address (see 3.1.162) of the attached phy (e.g., received by a logical phy in the incoming IDENTIFY address frame during the initialization sequence (see 4.1.2)), or the SAS address of the STP target port in an STP/SATA bridge (see 4.6.2)

3.1.14

attribute

when referring to classes (see 3.1.27) or objects (see 3.1.124), a named property of a class that describes the range of values that its objects may hold (see 3.5)

3.1.15

AT Attachment (ATA)

standard for the internal attachment of storage devices to hosts (see ATA8-AAM)

3.1.16

ATA device

storage peripheral that processes ATA commands and device management functions; Analogous to a SCSI target device (see ATA8-AAM)

3.1.17

ATA domain

I/O system consisting of an ATA host and one or more ATA devices that communicate with one another by means of a service delivery subsystem: analogous to a SCSI domain (see ATA8-AAM)

3.1.18

ATA host

host device that originates requests to be processed by an ATA device; analogous to a SCSI initiator device (see ATA8-AAM)

3.1.19

big-endian

format for storage or transmission of binary data in which the most significant byte appears first; in a multi-byte value, the byte containing the most significant bit is stored in the lowest memory address and transmitted first, and the byte containing the least significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 00h is stored in the lowest memory address, and the byte containing 80h is stored in the highest memory address)

3.1.20

bit cell time

time for transmitting a phy capabilities bit during SNW-3 (see 5.7.4.2.3.3)

3.1.21

Broadcast

information about an event in the SAS domain, communicated between phys with the BROADCAST primitive sequence (see 6.2.6.4) and/or the SMP ZONED BROADCAST function (see 9.4.3.20) and (see 4.1.13)

3.1.22

Broadcast propagation processor (BPP)

object within an expander function that manages Broadcasts (see 4.6.5)

3.1.23**burst time**

part of an OOB signal where the OOB burst is transmitted (see SAS-2.1)

3.1.24**byte**

sequence of eight contiguous bits considered as a unit; a byte is encoded as a character (see 3.1.25) using 8b10b coding (see 5.2)

3.1.25**character**

sequence of ten contiguous bits considered as a unit; a byte (see 3.1.24) is encoded as a character using 8b10b coding (see 5.2)

3.1.26**class**

description of a set of objects (see 3.1.124) that share the same attributes (see 3.1.14), operations (see 3.1.130), relationships, and semantics; classes may have attributes and may support operations

3.1.27**class diagram**

diagram that shows a collection of classes (see 3.1.27) and their contents and relationships (see 3.5)

3.1.28**command descriptor block (CDB)**

structure used to communicate a command from a SCSI application client to a SCSI device server (see SAM-5)

3.1.29**compliant jitter tolerance pattern (CJTPAT)**

test pattern for jitter testing (see SAS-2.1 and A.2)

3.1.30**configuration subprocess**

subprocess invoked from the discover process to configure an externally configurable expander device (see 4.8)

3.1.31**confirmation**

information passed from a lower layer state machine to a higher layer state machine, usually responding to a request (see 3.1.155) from that higher layer state machine, and sometimes relaying a response (see 3.1.157) from a peer higher layer state machine (e.g., see figure 42 in 4.6.6.2) (see 3.6)

3.1.32**connection**

temporary association between a SAS initiator port and a SAS target port using a pathway (see 6.13)

3.1.33**connection rate**

effective rate of dwords through the pathway (see 3.1.134) between a SAS initiator phy and a SAS target phy, established through the connection request

3.1.34**connection request**

request to establish a connection originated by a SAS phy (see 3.1.168) using an OPEN address frame (see 6.8.3) (see 6.13.2.1)

3.1.35

constraint

when referring to classes or objects, a mechanism for specifying semantics or conditions that are maintained as true between entities (e.g., a required condition between associations) (see 3.5)

3.1.36

control byte

byte containing control information defined in table 41 (see 5.2)

3.1.37

control character (Kxx.y)

character containing control information defined in table 41 (see 5.2)

3.1.38

cyclic redundancy check (CRC)

error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum (see 6.5)

3.1.39

D.C. idle

differential signal level that is nominally 0 V peak-to-peak (see SAS-2.1)

3.1.40

D.C. mode

mode in which D.C. idle is used, at times including, during the idle time and negation time of an OOB signal (see SAS-2.1), and during the RCDT time of speed negotiation windows (see 5.7.4.2.2)

3.1.41

data byte

byte containing data information defined in table 40 (see 5.2)

3.1.42

data character (Dxx.y)

character containing data information defined in table 40 (see 5.2)

3.1.43

data dword

dword containing four data bytes, or four data characters with correct disparity

3.1.44

deadlock

condition in which two or more processes (e.g., connection requests) are waiting on the others to complete, resulting in none of the processes completing

3.1.45

deletable primitive

ALIGN (see 6.2.5.1), NOTIFY (see 6.2.5.3), or MUX (see 6.2.5.2), which may be deleted by a receiver instead of being placed into its elasticity buffer (see 6.3) (see 6.2.5)

3.1.46

dependency

when referring to classes, a relationship between two classes where a change to one class (i.e., the independent class) may cause a change in the other class (i.e., the dependent class) (see 3.5)

3.1.47

device name

worldwide unique name for a device within a transport protocol (see 4.2.6)

3.1.48**device server**

object that processes SCSI commands (see SAM-5), ATA commands (see ATA8-AAM), or management functions (see 4.1.5)

3.1.49**direct current (D.C.)**

non-A.C. component of a signal (see SAS-2.1)

3.1.50**direct routing attribute**

attribute of an expander phy that indicates that the expander phy may be used by the ECM (see 3.1.60) to route a connection request to an end device (see 4.6.7.1)

3.1.51**direct routing method**

method the ECM (see 3.1.60) uses to route connection requests to an attached end device or an attached expander device (see 4.6.7.1)

3.1.52**discover process**

process performed by a management application client to discover all the SAS devices and expander devices in the SAS domain; The discover process invokes the configuration subprocess (see 3.1.30) as needed (see 4.7)

3.1.53**disparity**

difference between the number of ones and zeros in a character (see 5.2)

3.1.54**domain**

SAS domain (see 3.1.164), a SCSI domain (see 3.1.184), or an ATA domain (see 3.1.17)

3.1.55**dword**

sequence of four contiguous bytes or four contiguous characters considered as a unit;

NOTE 7 the meaning depends on the context (e.g., when discussing the bits being transmitted over a physical link, dword represents four characters (i.e., 40 bits) and when discussing the contents of a frame before 8b10b encoding (see 3.1.2) or after 10b8b decoding (see 3.1.3), dword represents four bytes (i.e., 32 bits))

3.1.56**dword synchronization**

detection of an incoming stream of dwords from a physical link by a phy (see 5.10)

3.1.57**enclosure**

box, rack, or set of boxes providing the powering, cooling, mechanical protection, EMI protection, and external electronic interfaces for one or more end device(s) and/or expander device(s); the enclosure provides the outermost electromagnetic boundary and acts as an EMI barrier; an enclosure is not a class in this standard

3.1.58**end device**

SAS device (see 3.1.163) or SATA device (see 3.1.174) that is not contained within an expander device (see 3.1.62)

3.1.59

event notification

information passed from the SSP transport layer to the SCSI application layer notifying the SCSI application layer that a SCSI event has occurred (see SAM-5)

3.1.60

expander connection manager (ECM)

object within an expander function (see 3.1.63) that manages routing (see 4.6.3)

3.1.61

expander connection router (ECR)

portion of an expander function (see 3.1.63) that routes messages between expander phys (see 4.6.4)

3.1.62

expander device

device that is part of a service delivery subsystem, facilitates communication between SAS devices and SATA devices, and is either an externally configurable expander device or a self-configuring expander device (see 4.1.7)

3.1.63

expander function

object within an expander device that contains an expander connection manager, expander connection router, and Broadcast propagation processor (see 4.6.1)

3.1.64

expander logical phy

expander phy (see 3.1.65) or a multiplexed portion of an expander phy (see 4.1.2)

3.1.65

expander phy

phy in an expander device (see 3.1.62) that interfaces to a service delivery subsystem (see 3.1.196)

3.1.66

expander port

expander device object that interfaces to a service delivery subsystem (see 3.1.196) and to SAS ports in other devices (see 4.6.2)

3.1.67

expander route entry

SAS address and an enable/disable bit in an expander route table (see 3.1.69)

3.1.68

expander route index

value used in combination with a phy identifier to select an expander route entry in an expander route table in an externally configurable expander device (see 4.6.7.4)

3.1.69

expander route table

table of expander route entries (see 3.1.67) within an expander device (see 3.1.62); the table is used by the expander function (see 3.1.63) to resolve connection requests (see 4.6.7.4)

3.1.70

externally configurable expander device

non-self-configuring expander device (see 3.1.62) that contains an expander route table (see 3.1.69) that is configurable with the SMP CONFIGURE ROUTE INFORMATION function (see 9.4.3.27) (see 4.1.7)

3.1.71**field**

group of one or more contiguous bits

3.1.72**Final-SNW**

final speed negotiation window for 1.5 or 3 Gbps without training (see 5.7.4.2.3.2)

3.1.73**frame**

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

3.1.74**frame information structure (FIS)**

SATA frame format (see SATA)

3.1.75**generalization**

when referring to classes (see 3.1.27), a relationship among classes where one class (i.e., the superclass) shares the attributes (see 3.1.14) and operations (see 3.1.130) of one or more other classes (i.e., the subclasses) (see 3.5)

3.1.76**hard reset**

SAS device or expander device action in response to a reset event in which the device performs the operations described in 4.4

3.1.77**hard reset sequence**

sequence that causes a hard reset (see 4.4 and 6.9)

3.1.78**hardware maximum physical link rate**

maximum physical link rate capability of a phy

3.1.79**hardware minimum physical link rate**

minimum physical link rate capability of a phy

3.1.80**hash**

mathematical function that maps values from a larger set of values into a smaller set of values, reducing a long value into a shorter hashed value

3.1.81**I_T nexus**

when referring to SAS ports (see 3.1.170), a nexus between a SAS initiator port and a SAS target port; when referring to SCSI ports (see 3.1.187), a nexus between a SCSI initiator port and a SCSI target port (see SAM-5)

3.1.82**I_T nexus loss**

when referring to SAS ports (see 3.1.170), a condition where a SAS port determines that another SAS port is no longer available (see 4.5); when referring to SCSI ports (see 3.1.187) (e.g., SSP ports), a condition resulting from the events defined by SAM-5 in which the SCSI device performs the I_T nexus loss operations described in SAM-5, SPC-4, and the appropriate command standards (see SAM-5)

3.1.83

I_T_L nexus

nexus between a SCSI initiator port, a SCSI target port, and a logical unit (see SAM-5)

3.1.84

I_T_L_Q nexus

nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a command (see SAM-5)

3.1.85

IDENTIFY (PACKET) DEVICE data

IDENTIFY DEVICE data from an ATA device, or IDENTIFY PACKET DEVICE data from an ATAPI device (see ACS-2)

3.1.86

identification sequence

sequence where phys exchange IDENTIFY address frames (see 4.4 and 6.9)

3.1.87

idle dword

vendor specific data dword that is scrambled and is transmitted outside a frame (see 6.4)

3.1.88

idle time

part of an OOB signal (see 3.1.129) where OOB idle (see 3.1.127) is being transmitted (see 5.6)

3.1.89

indication

information passed from a lower layer state machine to a higher layer state machine, usually relaying a request (see 3.1.155) from a peer higher layer state machine (e.g., see figure 42 in 4.6.6.1) (see 3.6)

3.1.90

information unit

portion of an SSP frame that carries command, task management function, data, response, or transfer ready information (see 8.2.2)

3.1.91

initiator connection tag

value in the OPEN address frame used for SSP and STP connection requests to provide a SAS initiator port an alternative to using the SAS target port's SAS address for context lookup when the SAS target port originates a connection request (see 6.8.3)

3.1.92

initiator port transfer tag

value that allows an SSP initiator port to establish a context for commands and task management functions (see 8.2.1)

3.1.93

invalid character

character that is not a control character (see 3.1.37) or a data character (see 3.1.42)

3.1.94

invalid dword

dword that is not a data dword (see 3.1.43) or a primitive (see 3.1.146) (i.e., in the character context, a dword that contains an invalid character, a control character in other than the first character position, a control character other than K28.3 or K28.5 in the first character position, or one or more characters with a running disparity error)

3.1.95**jitter tolerance pattern (JTPAT)**

test pattern for jitter testing (see SAS-2.1)

3.1.96**least significant bit (LSB)**

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

3.1.97**left-aligned**

type of field containing ASCII data in which unused bytes are placed at the end of the field (highest offset) and are filled with ASCII space (20h) characters (see SPC-4)

3.1.98**link reset**

performing the link reset sequence (see 3.1.99)

3.1.99**link reset sequence**

for SATA, a phy reset sequence (see 3.1.139); for SAS, a phy reset sequence followed by an identification sequence (see 3.1.86), or a phy reset sequence followed by a hard reset sequence (see 3.1.77), another phy reset sequence, and an identification sequence (see 4.4 and 6.9)

3.1.100**little-endian**

format for storage or transmission of binary data in which the least significant byte appears first; in a multi-byte value, the byte containing the least significant bit is stored in the lowest memory address and transmitted first, and the byte containing the most significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 80h is stored in the lowest memory address, and the byte containing 00h is stored in the highest memory address)

3.1.101**livelock**

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

3.1.102**locked zoning expander device**

zoning expander device that has been locked by a zone manager (see 4.9.6.2)

3.1.103**logical link**

physical link (see 3.1.140) or a multiplexed portion of a physical link (see 4.1.3)

3.1.104**logical link rate**

link rate between two logical phys established as a result of speed negotiation and multiplexing negotiation between the physical phys containing those logical phys

3.1.105**logical phy**

phy (see 3.1.136) or a multiplexed portion of a phy (see 4.1.2)

3.1.106**logical unit**

externally addressable entity within a SCSI target device that implements a SCSI device model and contains a device server (see SAM-5)

3.1.107

logical unit number (LUN)

identifier for a logical unit (see 3.1.106) (see SAM-5)

3.1.108

low phy power condition:

partial phy power condition or the slumber phy power condition (see 4.10.1)

3.1.109

maximum training time (MTT)

maximum time during Train-SNW (see 5.7.4.2.3.4) for a phy receiver to complete training (see 3.1.234)

3.1.110

maximum Train-SNW window time (MTWT)

maximum duration of Train-SNW (see 5.7.4.2.3.4)

3.1.111

media

plural of medium (see 3.1.112)

3.1.112

medium

material on which data is stored (e.g., a magnetic disk)

3.1.113

message

information sent between state machines (see 3.6)

3.1.114

most significant bit (MSB)

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

3.1.115

multiplexing

dividing a physical link into two logical links, and dividing a phy into two logical phys, by interleaving dwords (see 5.11)

3.1.116

multiplicity

when referring to classes (see 3.1.27), an indication of the range of allowable instances of an object or an attribute (see 3.5)

3.1.117

narrow link

physical link that attaches a narrow port to another narrow port (see 4.1.4)

3.1.118

narrow port

port that contains exactly one phy (see 4.1.4)

3.1.119

negation time

part of an OOB signal during which OOB idle is transmitted after the last OOB burst (see 5.6)

3.1.120**negotiation idle**

transmission of D.C. idle when D.C. mode (see 3.1.40) is enabled, or a defined sequence of dwords when optical mode is enabled (see 5.9.4.2)

3.1.121**negotiated logical link rate**

current operational logical link rate (see 3.1.104)

3.1.122**negotiated physical link rate**

current operational physical link rate (see 3.1.140)

3.1.123**nexus**

when referring to SAS devices (see 3.1.163), a relationship between two SAS devices, and the SAS initiator port and the SAS target port objects within those SAS devices; when referring to SCSI devices (see 3.1.183), a relationship between two SCSI devices, and the SCSI initiator port and the SCSI target port objects within those SCSI devices (see SAM-5)

3.1.124**object**

entity with a well-defined boundary and identity that encapsulates state and behavior; all objects are instances of classes (see 3.1.27) (i.e., a concrete manifestation of a class is an object)

3.1.125**object diagram**

diagram that encompasses objects (see 3.1.124) and their relationships at a point in time (see 3.5)

3.1.126**OOB burst**

transmission of signal transitions for a burst time (see 5.6.1)

3.1.127**OOB idle**

transmission of D.C. idle when D.C. mode (see 3.1.40) is enabled, or a defined sequence of dwords when optical mode (see 3.1.131) is enabled (see SAS-2.1)

3.1.128**OOB sequence**

sequence where two phys exchange OOB signals (see 5.7.2.1 and 5.7.4.1)

3.1.129**OOB signal**

pattern of idle time (see 3.1.88), burst time (see 3.1.23), and negation time (see 3.1.119) used during the link reset sequence (see 5.6)

3.1.130**operation**

when referring to classes (see 3.1.27), a service that may be requested from any object (see 3.1.124) of the class in order to affect behavior; operations describe what a class is allowed to do and may be a request or a query; a request may change the state of the object but a query should not (see 3.5)

3.1.131**optical mode**

mode in which a defined sequence of dwords is used, at times including, during the idle time and negation time of an OOB signal (see SAS-2.1), and during the RCDT time of speed negotiation windows (see 5.7.4.2.2)

3.1.132

partial pathway

set of logical links participating in a connection request that have not yet conveyed a connection response (see 4.1.11)

3.1.133

partial phy power condition:

low phy power condition for a SAS phy or expander phy (see 4.10.1.3)

3.1.134

pathway

set of logical links between a SAS initiator phy and a SAS target phy being used by a connection (see 4.1.11)

3.1.135

pathway blocked count

number of times the port has retried this connection request due to receiving OPEN_REJECT (PATHWAY BLOCKED), OPEN_REJECT (RESERVED STOP 0), or OPEN_REJECT (RESERVED STOP 1)

3.1.136

phy

object in a device that is used to interface to other devices (e.g., an expander phy or a SAS phy) (see 4.1.2 and SAS-2.1)

3.1.137

phy identifier

value by which a phy is identified within a device (see 4.2.10)

3.1.138

phy ready state:

condition of a phy when its SP state machine is in the SP15:SAS_PHY_Ready state (see 5.9.4.10)

3.1.139

phy reset sequence

OOB sequence followed by a speed negotiation sequence (see 4.4 and 5.7)

3.1.140

physical link rate

link rate between two physical phys established as a result of speed negotiation between those phys

3.1.141

port

SAS port (see 3.1.170) or an expander port (see 3.1.66); each port contains one or more phys (see 3.1.136) (see 4.1.4)

3.1.142

port identifier

value by which a port is identified within a domain (see 4.2.9)

3.1.143

port name

worldwide unique name for a port within a transport protocol (see 4.2.8)

3.1.144

potential pathway

set of logical links between a SAS initiator phy and a SAS target phy (see 4.1.11)

3.1.145**power on**

power being applied

3.1.146**primitive**

for a phy that supports being attached to SATA phy, a dword containing a 7Ch or BCh control byte followed by three data bytes, or a K28.3 or K28.5 control character with correct disparity followed by three data characters with correct disparity; for a phy that does not support being attached to SATA phy, a dword containing a BCh control byte followed by three data bytes, or a K28.5 control character with correct disparity followed by three data characters with correct disparity (see 6.2)

3.1.147**primitive sequence**

set of primitives treated as a single entity (see 6.2.4)

3.1.148**programmed maximum physical link rate**

maximum operational physical link rate of a phy (e.g., as programmed via the SMP PHY CONTROL function (see 9.4.3.28) or the Phy Control and Discover mode page (see 9.2.7.5))

3.1.149**programmed minimum physical link rate**

minimum operational physical link rate of a phy (e.g., as programmed via the SMP PHY CONTROL function (see 9.4.3.28) or the Phy Control and Discover mode page (see 9.2.7.5))

3.1.150**protocol**

either SSP (see 3.1.195), SMP (see 3.1.194), or STP (see 3.1.192)

3.1.151**rate**

data transfer rate of a physical or logical link (e.g., 1.5 Gbps, 3 Gbps, or 6 Gbps)

3.1.152**rate change delay time (RCDT)**

time between rates during the speed negotiation sequence (see 5.7.4.2)

3.1.153**read data**

data transferred to the SCSI application client's data-in buffer from the SCSI device server, as requested by the Send Data-In transport protocol service (see 9.2.1.6)

3.1.154**receiver device (Rx)**

device downstream from a receiver device compliance point containing a portion of the physical link and a receiver circuit (see SAS-2.1)

3.1.155**request**

information passed from a higher layer state machine to a lower layer state machine, usually to initiate some action (see 3.6)

3.1.156**reset event**

event that triggers a hard reset (see 4.4.2) in a SAS device

3.1.157

response

information passed from a higher layer state machine to a lower layer state machine, usually in response to an indication (see 3.6)

3.1.158

role

when referring to classes and objects (see 3.1.124), a label at the end of an association (see 3.1.12) or aggregation (see 3.1.10) that defines a relationship to the class on the other side of the association or aggregation (see 3.5)

3.1.159

route table optimization

configuration subprocess algorithm that reduces the number of entries required in an expander route table in an externally configurable expander device (see 4.8.3)

3.1.160

run length

number of consecutive identical bits in the transmitted signal (e.g., the pattern 0011111010 includes the following run lengths: five 1s, one 0, one 1, and indeterminate run lengths of 0s at the start and end)

3.1.161

running disparity (RD)

binary parameter with a negative (-) or positive (+) value indicating the cumulative encoded signal imbalance between the one and zero signal state of all characters since dword synchronization has been achieved (see 5.3.5)

3.1.162

SAS address

identifier assigned to a SAS port or expander device (see 4.2.4)

3.1.163

SAS device

SAS initiator device and/or a SAS target device

3.1.164

SAS domain

I/O system defined by this standard that may serve as a SCSI domain (see 4.1.9)

3.1.165

SAS initiator device

device containing SSP, STP, and/or SMP initiator ports in a SAS domain (see 4.1.6)

3.1.166

SAS initiator phy

logical phy (see 3.1.105) in a SAS initiator device

3.1.167

SAS initiator port

SSP initiator port (see 3.1.212), STP initiator port (see 3.1.219), and/or SMP initiator port (see 3.1.199) in a SAS domain

3.1.168

SAS phy

phy in a SAS device that interfaces to a service delivery subsystem (see 3.1.196)

3.1.169**SAS logical phy**

SAS phy (see 3.1.168) or a multiplexed portion of a SAS phy (see 4.1.2)

3.1.170**SAS port**

SAS initiator port (see 3.1.167) and/or a SAS target port (see 3.1.173)

3.1.171**SAS target device**

device containing SSP, STP, and/or SMP target ports in a SAS domain (see 4.1.6)

3.1.172**SAS target phy**

logical phy (see 3.1.105) in a SAS target device

3.1.173**SAS target port**

SSP target port (see 3.1.216), STP target port (see 3.1.224), and/or SMP target port (see 3.1.203) in a SAS domain

3.1.174**SATA device**

ATA device that contains a SATA device port in an ATA domain; analogous to a SAS target device (see 3.1.171)

3.1.175**SATA device port**

ATA device object that interfaces to a service delivery subsystem (see 3.1.196) with SATA; analogous to a SAS target port (see 3.1.173)

3.1.176**SATA host**

ATA host that contains a SATA host port in an ATA domain; analogous to a SAS initiator device (see 3.1.165)

3.1.177**SATA host port**

ATA host object that interfaces to a service delivery subsystem (see 3.1.196) with SATA; analogous to a SAS initiator port (see 3.1.167)

3.1.178**SATA phy**

phy in a SATA device or SATA port selector that interfaces to a service delivery subsystem (see 3.1.196); analogous to a SAS phy (see 3.1.168)

3.1.179**SATA port selector**

device that attaches to two SATA host ports (i.e., two ATA domains) and one SATA device port, and provides the means for one SATA host to access the device at any given time (see SATA)

3.1.180**SATA spinup hold**

state entered by an expander phy attached to a SATA device in which it halts the automatic phy reset sequence to delay temporary consumption of additional power (e.g., to spin up rotating media) (see 5.12)

3.1.181**saturating counter**

counter that remains at its maximum value after reaching its maximum value

3.1.182

scrambling

modifying data by XORing each bit with a pattern generated by a linear feedback shift register to minimize repetitive character patterns (see 6.6)

3.1.183

SCSI device

device that contains one or more SCSI ports that are connected to a service delivery subsystem (see 3.1.196) and supports a SCSI application protocol (see SAM-5)

3.1.184

SCSI domain

I/O system consisting of a set of SCSI devices that communicate with one another by means of a service delivery subsystem (see SAM-5)

3.1.185

SCSI initiator device

SCSI device containing SCSI application clients and SCSI initiator ports that originates device service and task management requests to be processed by a SCSI target device and receives device service and task management responses from SCSI target devices (see SAM-5)

3.1.186

SCSI initiator port

SCSI initiator device object that acts as the connection between SCSI application clients and a service delivery subsystem (see 3.1.196) through which requests and confirmations are routed (see SAM-5)

3.1.187

SCSI port

SCSI initiator port and/or a SCSI target port (see SAM-5)

3.1.188

SCSI target device

SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing and sends device service and task management responses to SCSI initiator devices (see SAM-5)

3.1.189

SCSI target port

SCSI target device object that contains a task router and acts as the connection between SCSI device servers and task managers and a service delivery subsystem (see 3.1.196) through which requests, indicates, responses, and confirmations are routed (see SAM-5)

3.1.190

self-configuring expander device

expander device (see 3.1.62) containing an SMP initiator port and a management application client to perform the discover process (see 3.1.52) to configure its own expander route table (see 3.1.69) (see 4.7.4)

3.1.191

Serial ATA (SATA)

protocol defined by SATA (see clause 2)

3.1.192

Serial ATA Tunneled Protocol (STP)

protocol defined in this standard used by STP initiator ports (see 3.1.219) to communicate with STP target ports (see 3.1.224) in a SAS domain (see 6.18 and 8.3)

3.1.193**Serial Attached SCSI (SAS)**

set of protocols and the interconnect defined by this standard

3.1.194**Serial Management Protocol (SMP)**

protocol defined in this standard used by SMP initiator ports (see 3.1.199) to communicate with SMP target ports (see 3.1.203) in a SAS domain (see 6.19 and 8.4)

3.1.195**Serial SCSI Protocol (SSP)**

protocol defined in this standard used by SSP initiator ports (see 3.1.212) to communicate with SSP target ports (see 3.1.216) in a SAS domain (see 6.17 and 8.2)

3.1.196**service delivery subsystem**

part of a SCSI I/O system that transmits information between a SCSI initiator port and a SCSI target port, or the part of an ATA I/O system that transmits information between an ATA host and an ATA device, or the part of a SAS I/O system that transmits information between a SAS initiator port and a SAS target port

3.1.197**slumber phy power condition:**

low phy power condition for a SAS phy or expander phy (see 4.10.1.4)

3.1.198**SMP initiator phy**

SAS initiator phy (see 3.1.166) in an SMP initiator port (see 3.1.199)

3.1.199**SMP initiator port**

SAS initiator device object in a SAS domain that interfaces to a service delivery subsystem with SMP

3.1.200**SMP phy**

SAS logical phy (see 3.1.169) in an SMP port

3.1.201**SMP port**

SMP initiator port (see 3.1.199) and/or an SMP target port (see 3.1.203)

3.1.202**SMP target phy**

SAS target phy (see 3.1.172) in an SMP target port (see 3.1.203)

3.1.203**SMP target port**

SAS target device object in a SAS domain that interfaces to a service delivery subsystem (see 3.1.196) with SMP

3.1.204**SMP zone configuration function**

SMP function that configures zoning expander shadow values and is only accepted by a locked zoning expander device (see 4.9.6.3)

3.1.205**speed negotiation lock time (SNLT)**

maximum time during an SNW (see 3.1.208) for a transmitter to reply with ALIGN (1) (see 5.7.4.2)

3.1.206

speed negotiation sequence

sequence in which two phys negotiate the operational physical link rate (see 5.7.2.2 and 5.7.4.2)

3.1.207

speed negotiation transmit time (SNTT)

during SNW-1, SNW-2, and Final-SNW, the time after RCDT during which ALIGN (0) or ALIGN (1) is transmitted; during SNW-3, the time after RCDT in which bit cells and OOB idle are transmitted (see 5.7.4.2.2)

3.1.208

speed negotiation window (SNW)

portion of the SAS speed negotiation sequence (see 5.7.4.2.3)

3.1.209

speed negotiation window time (SNWT)

duration of an SNW (see 3.1.209) (see 5.7.4.2.3)

3.1.210

spread spectrum clocking (SSC)

technique of modulating the operating frequency of a transmitted signal (i.e., the physical link rate) to reduce the measured peak amplitude of radiated emissions (see SAS-2.1)

3.1.211

SSP initiator phy

SAS initiator phy (see 3.1.166) in an SSP initiator port (see 3.1.212)

3.1.212

SSP initiator port

SCSI initiator port in a SAS domain that implements SSP

3.1.213

SSP phy

SAS logical phy (see 3.1.169) in an SSP port

3.1.214

SSP port

SSP initiator port (see 3.1.212) and/or an SSP target port (see 3.1.216)

3.1.215

SSP target phy

SAS target phy (see 3.1.172) in an SSP target port (see 3.1.216)

3.1.216

SSP target port

SCSI target port in a SAS domain that implements SSP

3.1.217

state machine variable

variable that exists within the context of a state machine that may contain status from one state that is used in another state of the same state machine, affecting subsequent state transitions or state machine outputs.

3.1.218

STP initiator phy

SAS initiator phy (see 3.1.166) in an STP initiator port (see 3.1.219)

3.1.219**STP initiator port**

SAS initiator device object in a SAS domain that interfaces to a service delivery subsystem (see 3.1.196) with STP

3.1.220**STP phy**

SAS logical phy (see 3.1.169) in an STP port

3.1.221**STP port**

STP initiator port (see 3.1.219) and/or an STP target port (see 3.1.224)

3.1.222**STP primitive**

primitive used only inside STP connections and on SATA physical links (see 6.2.2)

3.1.223**STP target phy**

SAS target phy (see 3.1.172) in an STP target port (see 3.1.224)

3.1.224**STP target port**

SAS target device object in a SAS domain that interfaces to a service delivery subsystem (see 3.1.196) with STP

3.1.225**STP/SATA bridge**

expander device object containing an STP target port, a SATA host port, and the functions required to forward information between the STP target port and SATA host port to enable STP initiator ports in a SAS domain to communicate with SATA devices in an ATA domain

3.1.226**subtractive routing attribute**

attribute of an expander phy that indicates that the expander phy may be used by the ECM (see 3.1.60) to route connection requests to an attached expander device that were not resolved using the direct routing method or table routing method (see 4.6.7.1)

3.1.227**subtractive routing method**

method the ECM uses to route connection requests not resolved using the direct routing method or table routing method to an attached expander device (see 4.6.7.1)

3.1.228**table routing attribute**

attribute of an expander phy that indicates that the expander phy may be used by the ECM to route connection requests using an expander route table (see 4.6.7.1)

3.1.229**table routing method**

method the ECM uses to route connection requests to an attached expander device using an expander route table (see 4.6.7.1)

3.1.230**target port transfer tag**

optional value that allows an SSP target port to establish the write data context when receiving a write DATA frame (see 8.2.1)

3.1.231

task management function

task manager service capable of being requested by a SCSI application client to affect the processing of one or more commands (see SAM-5)

3.1.232

task manager

object within a SCSI target device that controls the sequencing of SCSI commands and processes SCSI task management functions (see SAM-5)

3.1.233

trained

physical link rate negotiated with Train-SNW (see 5.7.4.2)

3.1.234

training

process of adapting equalization circuitry in a receiver device to an incoming pattern

3.1.235

training lock time (TLT)

maximum time during Train-SNW (see 5.7.4.2.3.4) for a receiver to complete training (see 3.1.234) and transmit TRAIN_DONE patterns instead of TRAIN patterns

3.1.236

Train-SNW

speed negotiation window with training (see 5.7.4.2.3.4)

3.1.237

Train-SNW window time

actual duration of Train-SNW (see 5.7.4.2.3.4)

3.1.238

transmitter device (Tx)

device upstream from a transmitter device compliance point containing a portion of the physical link and a transmitter circuit (see SAS-2.1)

3.1.239

transport protocol service confirmation

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

3.1.240

transport protocol service indication

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

3.1.241

transport protocol service request

information passed from the SCSI application layer to the SSP transport layer (i.e., from the SCSI application client to the SCSI initiator port) to begin a SCSI transport protocol service

3.1.242

transport protocol service response

information passed from the SCSI application layer to the SSP transport layer (i.e., from the SCSI device server to the SSP target port) that completes the SCSI transport protocol service

3.1.243**valid character**

character that is a control character (see 3.1.37) or a data character (see 3.1.42)

3.1.244**valid dword**

dword that is a data dword (see 3.1.43) or a primitive (see 3.1.146)

3.1.245**virtual phy**

phy (see 3.1.136) that interfaces with a vendor specific interface to another virtual phy inside the same device (see 4.1.2)

3.1.246**wide link**

group of physical links that attaches a wide port to another wide port (see 4.1.4)

3.1.247**wide port**

port that contains more than one phy (see 4.1.4)

3.1.248**wrapping counter**

counter that wraps back to zero after reaching its maximum value

3.1.249**write data**

data transferred from the SCSI application client's data-out buffer to the SCSI device server, as requested by the Request Data-Out transport protocol service (see 9.2.1.8)

3.1.250**zone group**

set of phys in a ZPSDS (see 3.1.253) that all have the same access permission (see 4.9)

3.1.251**zone manager**

entity responsible for configuring a ZPSDS (see 4.9.1)

3.1.252**zone permission table**

table that defines access permission between the zone group of a source phy and the zone group of the destination phy (see 4.9.3.3)

3.1.253**zoned portion of a service delivery subsystem (ZPSDS)**

group of zoning expander devices that cooperate to control access between phys; the ZPSDS may include all or part of a service delivery subsystem (see 4.9)

3.1.254**zoning expander current values**

current zone permission table and zone phy information in a zoning expander device (see 4.9.6)

3.1.255**zoning expander device**

expander device (see 3.1.62) that supports zoning (see 4.9)

3.1.256

zoning expander phy

expander phy in a zoning expander device (see 3.1.255)

3.1.257

zoning expander shadow values

shadow zone permission table (see 3.1.252) and zone phy information in a zoning expander device, which are changed by SMP zone configuration functions (see 3.1.204) but do not become active until the activate step (see 4.9.6.4) is performed (see 4.9.6)

3.2 Symbols and abbreviations

See clause 2 for abbreviations of standards bodies (e.g., ISO).

Units and abbreviations used in this standard:

Abbreviation	Meaning
AA	ATA application layer (see 9.3)
ACA	auto contingent allegiance (see SAM-5)
ACK	acknowledge primitive (see 6.2.7.1)
AIP	arbitration in progress primitive (see 6.2.6.1)
ATA	AT attachment (see 3.1.15)
ATAPI	AT attachment packet interface
ATA8-AAM	AT Attachment - 8 ATA/ATAPI Architecture Model standard (see clause 2)
ACS-2	ATA/ATAPI Command Set-2 standard (see clause 2)
AWT	arbitration wait time
BCH	Bose, Chaudhuri and Hocquenghem code (see 4.2.5)
BIST	built in self test
BPP	Broadcast propagation processor (see 3.1.22)
CDB	command descriptor block (see 3.1.28)
CJTPAT	compliant jitter tolerance pattern (see 3.1.29)
CRC	cyclic redundancy check (see 3.1.38)
CRN	command reference number (see SAM-5)
D.C.	direct current
Dxx.y	data character (see 3.1.42)
ECM	expander connection manager (see 3.1.60)
ECR	expander connection router (see 3.1.61)
EMI	electromagnetic interference (see 3.1.57)
EOAF	end of address frame primitive (see 6.2.6.6)
EOF	end of frame primitive (see 6.2.7.4)
FIS	frame information structure (see)
G1	generation 1 physical link rate (i.e., 1.5 Gbps)
G2	generation 2 physical link rate (i.e., 3 Gbps)
G3	generation 3 physical link rate (i.e., 6 Gbps)
Gbps	gigabits per second (i.e., 10 ⁹ bits per second)
ID	identifier
JTPAT	jitter tolerance pattern (see 3.1.95)

Abbreviation	Meaning
Kxx.y	control character (see 3.1.37)
LED	light-emitting diode
LSB	least significant bit (see 3.1.96)
LUN	logical unit number (see 3.1.107)
µs	microsecond (i.e., 10 ⁻⁶ seconds)
m	meter
MA	management application layer (see 9.4)
MSB	most significant bit (see 3.1.114)
ms	millisecond (i.e., 10 ⁻³ seconds)
MT	SMP transport layer state machines (see 8.4)
MTT	maximum training time (see 3.1.109)
MTWT	maximum Train-SNW window time (see 3.1.110)
N/A	not applicable
NAA	name address authority (see 4.2)
NAK	negative acknowledge primitive (see 6.2.7.5)
ns	nanosecond (i.e., 10 ⁻⁹ seconds)
NVRAM	non-volatile random-access memory
OOB	out-of-band
OObI	out-of-band interval
OUI	organizationally unique identifier (i.e., company identifier)
PL	port layer state machines (see 7.2)
RCDT	rate change delay time (see 3.1.152)
RD	running disparity (see 3.1.161)
RRDY	receiver ready primitive (see 6.2.7.6)
Rx	receiver device (see 3.1.154)
SA	SCSI application layer (see 9.2)
SAM-5	SCSI Architecture Model - 5 standard (see clause 2)
SAS	Serial Attached SCSI (see 3.1.193)
SATA	Serial ATA (see 3.1.191) or the Serial ATA 3.0 specification (see clause 2)
SBC-3	SCSI Block Commands - 3 standard (see clause 2)
SCSI	Small Computer System Interface family of standards
SL	link layer for SAS phys state machines (see 6.15)
SLIR	link layer identification and hard reset state machines (see 6.10)
SMP	Serial Management Protocol (see 3.1.194), or link layer for SMP phys state machines (see 6.19.5)
SNLT	speed negotiation lock time (see 3.1.205)
SNTT	speed negotiation transmit time (see 3.1.207)
SNW	speed negotiation window (see 3.1.208)
SNW-1	speed negotiation window for 1.5 Gbps without training (see 5.7.4.2.3.2)
SNW-2	speed negotiation window for 3 Gbps without training (see 5.7.4.2.3.2)
SNW-3	speed negotiation window negotiating physical link rates with training (see 5.7.4.2.3.3)

Abbreviation	Meaning
SNWT	speed negotiation window time (see 3.1.209)
SOAF	start of address frame primitive (see 6.2.6.14)
SOF	start of frame primitive (see 6.2.7.7)
SP	phy layer state machine (see 5.9)
SP_DWS	phy layer dword synchronization state machine (see 5.10)
SPC-4	SCSI Primary Commands - 4 standard (see clause 2)
SSP	Serial SCSI Protocol (see 3.1.195), or link layer for SSP phys state machines (see 6.17.8)
ST	SSP transport layer state machines (see 8.2)
STP	Serial ATA Tunneled Protocol (see 3.1.192), or link layer for STP phys state machines (see 6.18.8)
s	second (unit of time)
TLT	training lock time (see 3.1.235)
TT	STP transport layer state machines (see 8.3)
Tx	transmitter device (see 3.1.238)
VPD	vital product data (see 9.2.11)
XL	link layer for expander phys state machine (see 6.16)
XOR	exclusive logical OR
ZP[s, d]	Zone permission bit for a source zone group (i.e., s) and a destination zone group (i.e., d) in the zone permission table (see 4.9.3.3)
ZPSDS	Zoned portion of a service delivery subsystem (see 3.1.253)
^	exclusive logical OR
≤	less than or equal to
≥	greater than or equal to
+	plus
-	minus
±	plus or minus
×	multiplied by
/	divided by
v	the absolute value (i.e., magnitude) of v
~	approximately equal to
∈	set membership
®	registered trademark

3.3 Keywords

3.3.1

invalid

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

3.3.2

mandatory

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

3.3.3**may**

keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

3.3.4**may not**

keywords that indicate flexibility of choice with no implied preference (equivalent to “may or may not”)

3.3.5**obsolete**

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

3.3.6**optional**

keyword that describes features that are not required to be implemented by this standard; however, if any optional feature defined in this standard is implemented, then it shall be implemented as defined in this standard

3.3.7**reserved**

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

3.3.8**restricted**

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

3.3.9**shall**

keyword indicating a mandatory requirement; designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard

3.3.10**should**

keyword indicating flexibility of choice with a strongly preferred alternative (equivalent to “is strongly recommended”)

3.3.11**vendor specific**

something (e.g., a bit, field, or code value) that is not defined by this standard and may be used differently in various implementations

3.4 Editorial conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 3 or in the text where they first appear.

Names of signals, address frames, primitives and primitive sequences, SMP functions, state machines, SCSI and ATA commands, SCSI statuses, SCSI sense keys, and SCSI additional sense codes are in all uppercase (e.g., REQUEST SENSE command).

Names of messages, requests, confirmations, indications, responses, event notifications, timers, SCSI diagnostic pages, SCSI mode pages, and SCSI log pages are in mixed case (e.g., Disconnect-Reconnect mode page).

Names of fields are in small uppercase (e.g., DESTINATION SAS ADDRESS). Normal case is used when the contents of a field are being discussed. Fields containing only one bit are usually referred to as the NAME bit instead of the NAME field.

Normal case is used for words having the normal English meaning.

A binary number is represented in this standard 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 between characters in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

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

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

This standard uses the following conventions for representing decimal numbers:

- a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
- b) the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space; and
- c) the thousands separator is used in both the integer portion and the fraction portion of a number.

Table 2 shows some examples of decimal numbers using various numbering conventions.

Table 2 — Numbering conventions

ISO/IEC	USA	This standard
0,6	0.6	0.6
3,141 592 65	3.14159265	3.141 592 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

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

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 colors named:

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

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

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

- 1) top;

- 2) middle; and
- 3) bottom.

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

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

3.5 Class diagram and object diagram conventions

The notation used in class diagrams and object diagrams is based on the Unified Modeling Language (UML) specification.

Figure 4 shows the notation used for classes in class diagrams.

Notation for a class with no attributes or operations:

Class Name

Notation for a class with attributes and no operations:

Class Name
Attribute 1
Attribute 2

Notation for a class with operations and no attributes:

Class Name
Operation 1()
Operation 2()

Notation for a class with attributes and operations:

Class Name
Attribute 1
Attribute 2
Operation 1()
Operation 2()

Notation for a class with attributes showing multiplicity and operations:

Class Name
Attribute 1[1..*]
Attribute 2[1]
Operation 1()
Operation 2()

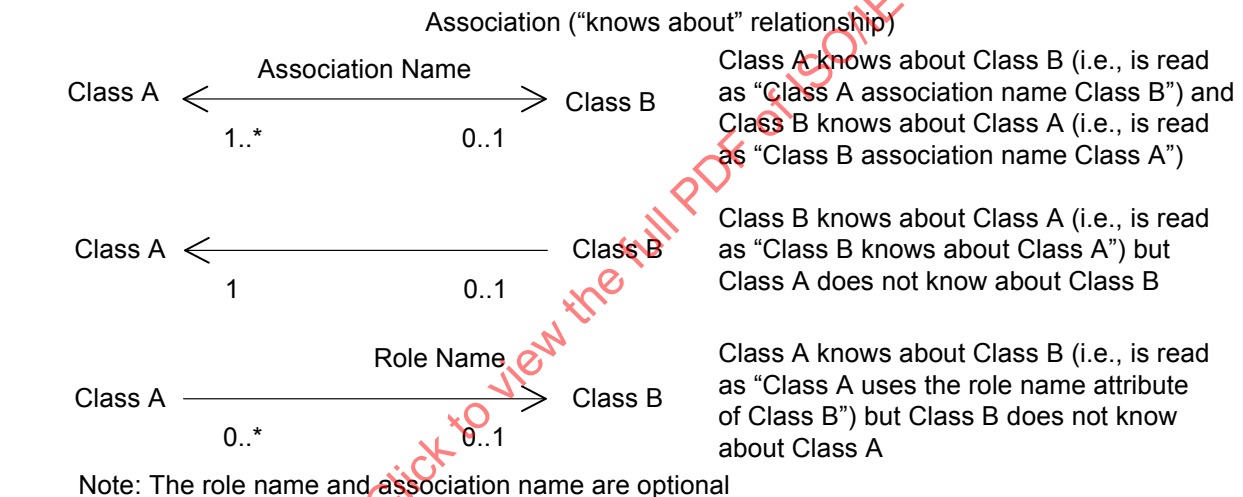
Figure 4 — Classes in class diagrams

Table 3 defines the notation used for multiplicity in class diagrams.

Table 3 — Multiplicity notation in class diagrams

Notation	Description
<none>	The number of instances of the object or attribute is not specified.
1	One instance of the object or attribute exists.
0..*	Zero or more instances of the object or attribute exist.
1..*	One or more instances of the object or attribute exist.
0..1	Zero or one instances of the object or attribute exist.
n..m	n to m instances of the object or attribute exist (e.g., 2..8).
x, n..m	Multiple disjoint instances of the object or attribute exist (e.g., 2, 8..15).

Figure 5 defines the notation used for association relationships between classes.



Examples of class diagrams using associations:

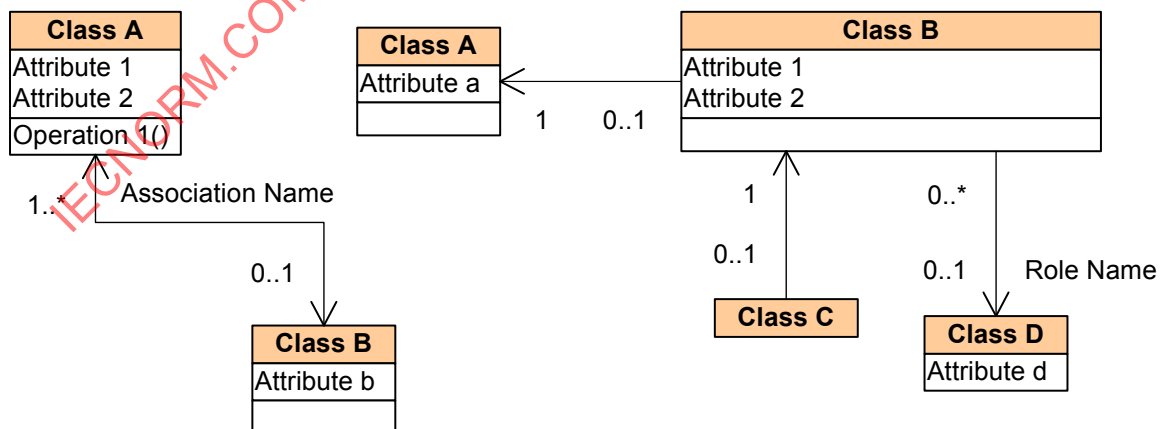


Figure 5 — Association relationships in class diagrams

Figure 6 defines the notation used for aggregation relationships between classes.

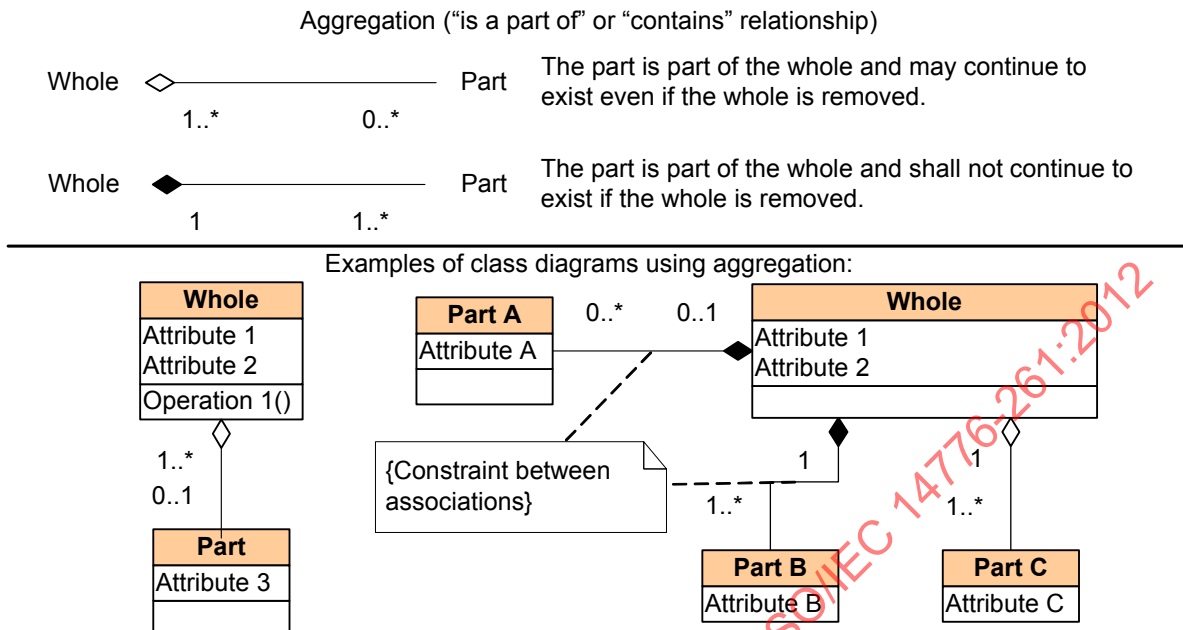


Figure 6 — Aggregation relationships in class diagrams

Figure 7 defines the notation used for generalization relationships between classes.

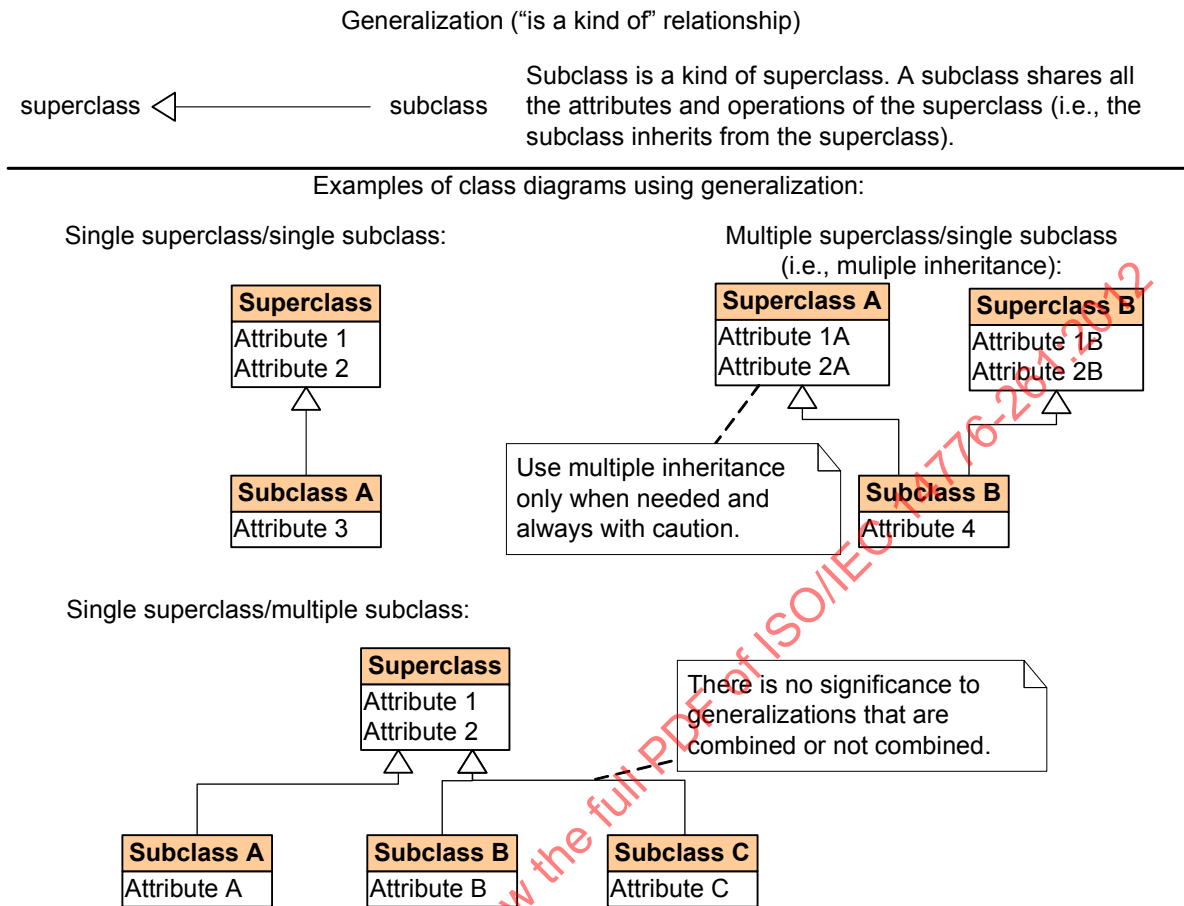


Figure 7 — Generalization relationships in class diagrams

Figure 8 defines the notation used for dependency relationships between classes.

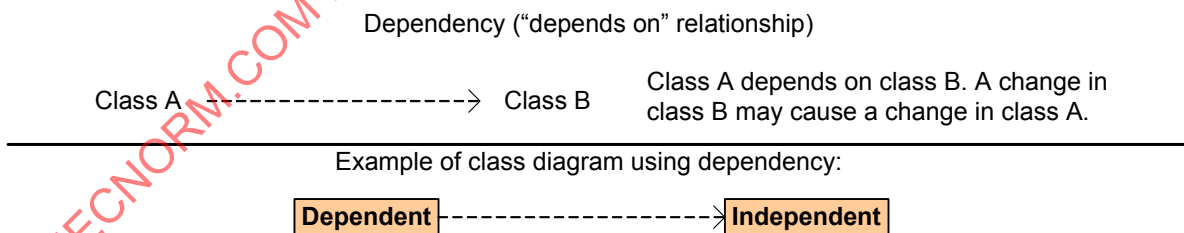


Figure 8 — Dependency relationships in class diagrams

Figure 9 defines the notation used for objects in object diagrams.

Notation for a named object with no attributes:

label : Class Name

Notation for a named object with attributes:

label : Class Name
Attribute 1 = x
Attribute 2 = y

Notation for an anonymous object with no attributes:

: Class Name

Notation for an anonymous object with attributes:

: Class Name
Attribute 1 = x
Attribute 2 = y

Figure 9 — Objects in object diagrams

A constraint is specified in a class diagram or an object diagram as text encapsulated with a { } notation within a box. See figure 6 for an example of a constraint.

A note is specified in a class diagram or an object diagram as text within a box. See figure 7 for an example of a note.

3.6 State machine conventions

3.6.1 State machine conventions overview

Figure 10 shows how state machines are described. See 4.3 for a summary of the state machines in this standard.

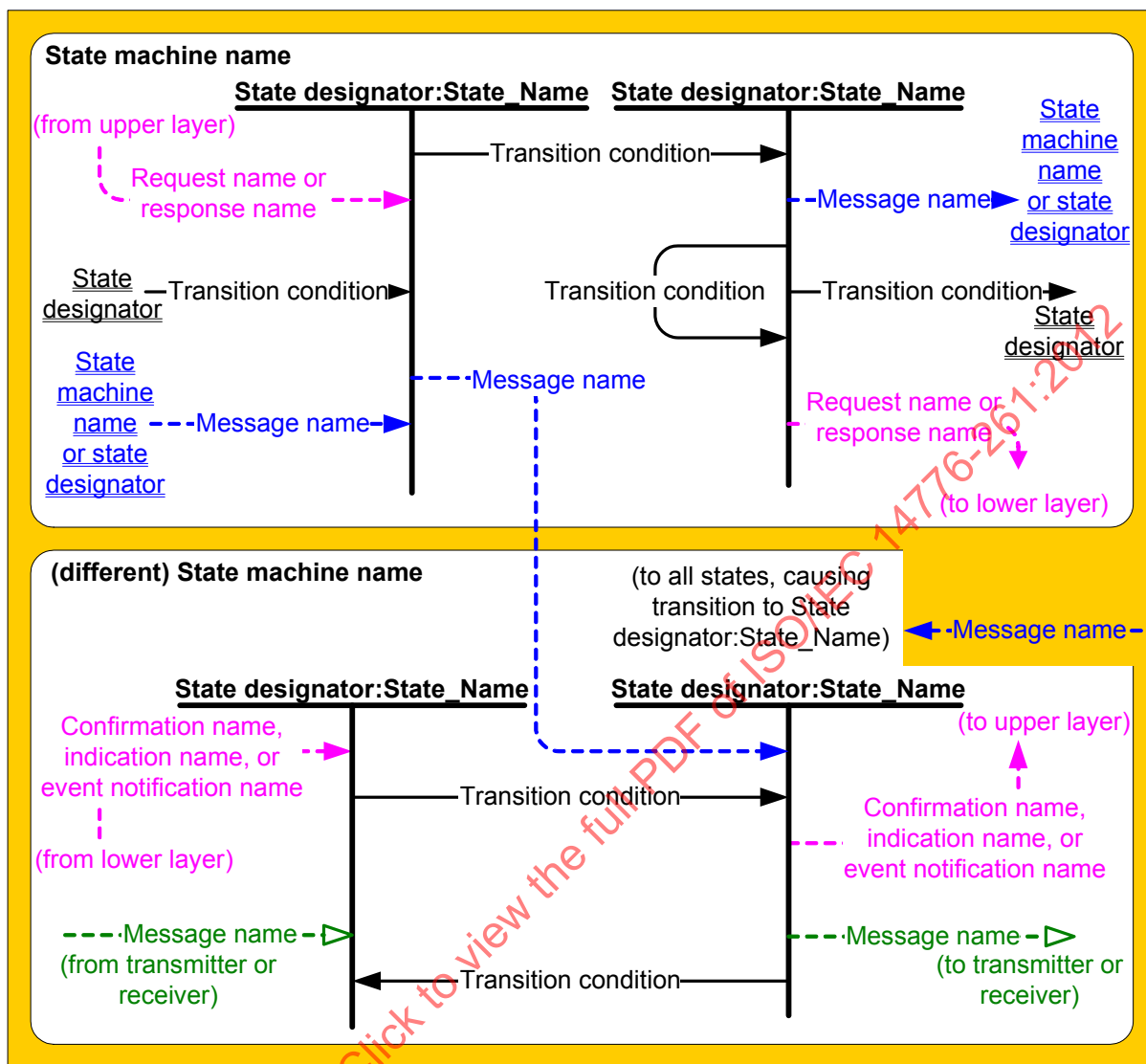


Figure 10 — State machine conventions

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

Each state machine is identified by a state machine name. In state machines with one state, the state machine is identified by a state designer. In state machines with multiple states, each state is identified by a state designer and a state name. The state designer (e.g., SL1) is unique among all state machines in this standard. 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.

3.6.2 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 in one figure goes to or comes from a state machine or state in a different figure, then the transition is shown going to or coming from a state machine name or a state designer label with double underlines.

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.

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. Transitions between states are instantaneous.

3.6.3 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 within the same layer of the protocol, they are identified by either:

- a) 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;
- b) a dashed line to or from a state in another state machine in the same figure; or
- c) 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.

The meaning of each message is described in the state description text.

Requests, indications, confirmations, responses, and event notifications are shown with curved dashed lines originating from or going toward 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 machine in one state that is repeatedly sending a message transitions to another state, then it stops repeatedly sending that message unless stated otherwise in the new state.

3.6.4 State machine counters, timers, and variables

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

- a) their scope is the state machine itself;
- b) they are created and deleted with the 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 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).

3.6.5 State machine arguments

State machines may contain one or more arguments received in a message or confirmation as state machine arguments. The following properties apply to state machine arguments:

- a) the state machine that sends an argument owns that argument's value;
- b) the state machine that receives an argument shall not modify that argument's value;
- c) the state machine that sends an argument may resend that argument with a different value;
- d) the scope of a state machine argument is the state machine itself; and
- e) state machine argument usage is described in the state descriptions and the transition descriptions.

3.7 Bit and byte ordering

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 8 or fewer bits.

In a field in a table consisting of more than one byte that contains a single value (e.g., a number), the byte containing the MSB is stored at the lowest address and the byte containing the LSB is stored at the highest address (i.e., big-endian byte ordering). The MSB and LSB are labeled.

NOTE 8 SATA numbers bits within fields the same as this standard, but uses little-endian byte ordering.

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), the MSB label is the MSB of the first character, and the LSB label is the LSB of the last character.

Multiple byte fields are represented by only two rows, where a non-sequentially increasing byte number indicates the presence of additional bytes.

A data dword consists of 32 bits. Table 4 shows a data dword containing a single value, where the MSB is on the left in bit 31, and the LSB is on the right in bit 0.

Table 4 — Data dword containing a value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
MSB																Value														LSB					

Table 5 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the left and byte 3 (the fourth byte) is on the right. Each byte has an MSB on the left and an LSB on the right.

Table 5 — Data dword containing four one-byte fields

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
MSB				Byte 0 (First byte)				LSB				MSB				Byte 1 (Second byte)				LSB				MSB				Byte 2 (Third byte)				LSB				MSB				Byte 3 (Fourth byte)				LSB			

3.8 Notation for procedures and functions

In this standard, the model for functional interfaces between objects is the callable procedure. 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 data objects to be specified as inputs and outputs.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

4 General

4.1 Architecture

4.1.1 Architecture overview

A SAS domain (see 4.1.9) contains one or more SAS devices and a service delivery subsystem. A SAS domain may be a SCSI domain (see SAM-5).

A SAS device (see 4.1.6) contains one or more SAS ports (see 4.1.4). A SAS device may be a SCSI device (see SAM-5).

A SAS port (see 4.1.4) contains one or more phys (see 4.1.2). A SAS port may be a SCSI port (see SAM-5).

The service delivery subsystem (see 4.1.8) in a SAS domain may contain expander devices (see 4.1.7).

Expander devices contain expander ports (see 4.1.4) and one SMP port.

An expander port contains one or more phys (see SAS-2.1).

An expander device shares its phys with the SAS device(s) contained within the expander device.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Figure 11 shows the class diagram for a SAS domain, showing the relationships between SAS Domain, SCSI Domain, Service Delivery Subsystem, Expander Device, Expander Port, SAS Device, SCSI Device, SAS Port, SCSI Port, and Phy classes. Not all attributes are shown.

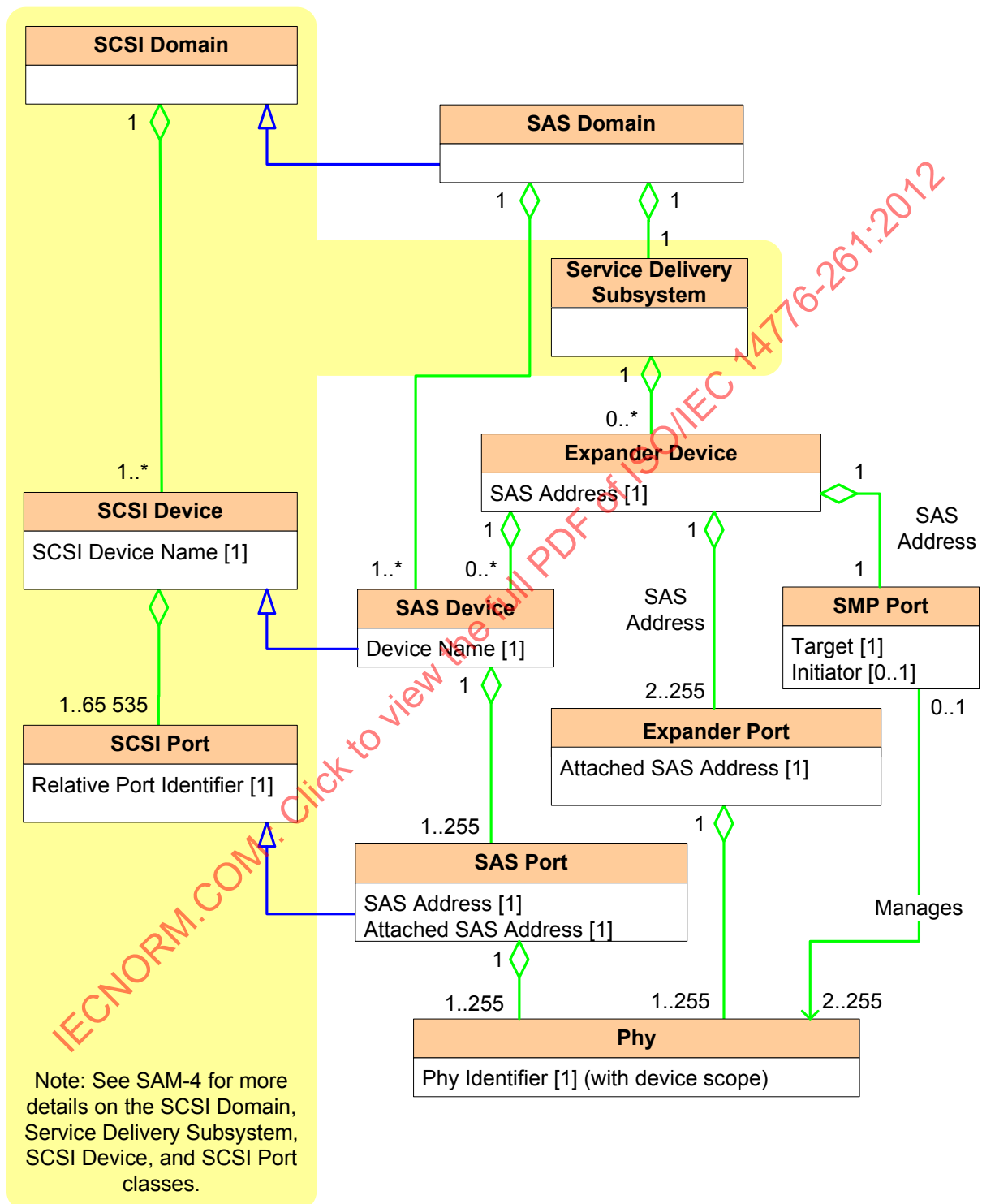


Figure 11 — SAS Domain class diagram

4.1.2 Physical links and phys

For the electrical and mechanical specifications of the physical link and physical phy, see SAS-2.1.

A device (i.e., a SAS device (see 4.1.6) or expander device (see 4.1.7)) contains one or more phys.

Each phy has:

- a) a SAS address (see 4.2.4), inherited from the SAS port (see 4.1.4) or expander device;
- b) a phy identifier (see 4.2.10) which is unique within the device;
- c) optionally, support for being an SSP initiator phy;
- d) optionally, support for being an STP initiator phy;
- e) optionally, support for being an SMP initiator phy;
- f) optionally, support for being an SSP target phy;
- g) optionally, support for being an STP target phy; and
- h) optionally, support for being an SMP target phy.

A phy may be used as one or two logical phys based on multiplexing (see 5.11).

During the identification sequence (see 6.9), a logical phy:

- a) transmits an IDENTIFY address frame including the device type (i.e., end device or expander device) of the device containing the phy, the SAS address of the SAS port or expander device containing the logical phy, and other information; and
- b) receives an IDENTIFY address frame containing the same set of information from the attached logical phy, including the attached device type, the attached SAS address, the attached device name, and other attached information.

Figure 12 defines the Phy classes, showing the relationships between the following classes:

- a) Phy;
- b) Logical Phy;
- c) SAS Phy;
- d) Logical SAS Phy;
- e) Expander Phy;
- f) Logical Expander Phy;
- g) SAS Initiator Phy;
- h) SAS Target Phy;
- i) SSP Phy;
- j) STP Phy; and
- k) SMP Phy.

SATA phys are also referenced in this standard but are defined by SATA.

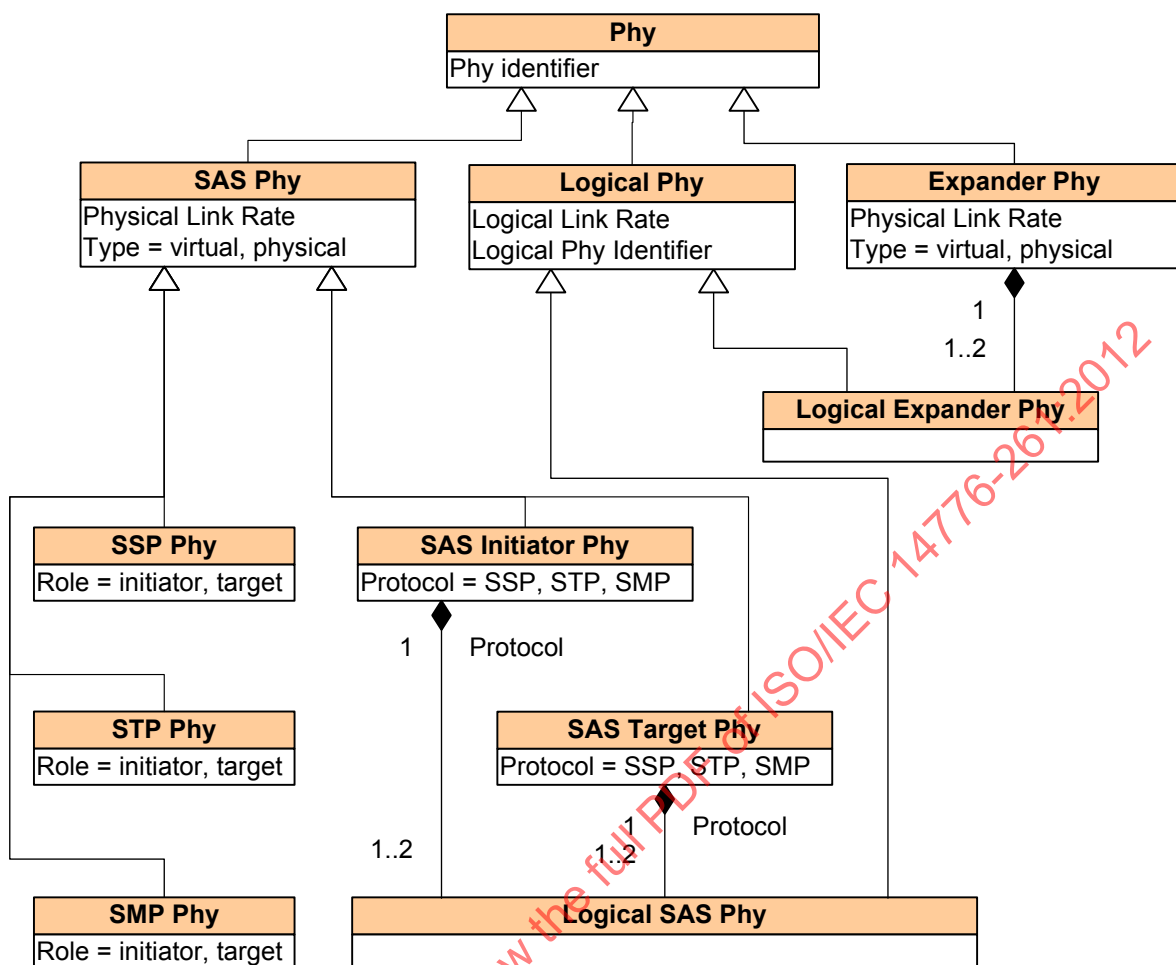


Figure 12 — Phy class diagram

Figure 13 shows example objects instantiated from the SAS Phy class, including:

- a) SSP initiator phy;
- b) SSP target phy;
- c) virtual SMP initiator phy;
- d) virtual SMP target phy;
- e) STP initiator phy; and
- f) logical SAS phy.

A SAS phy is represented by one of these objects during each connection. A SAS phy may be represented by different phy objects in different connections.

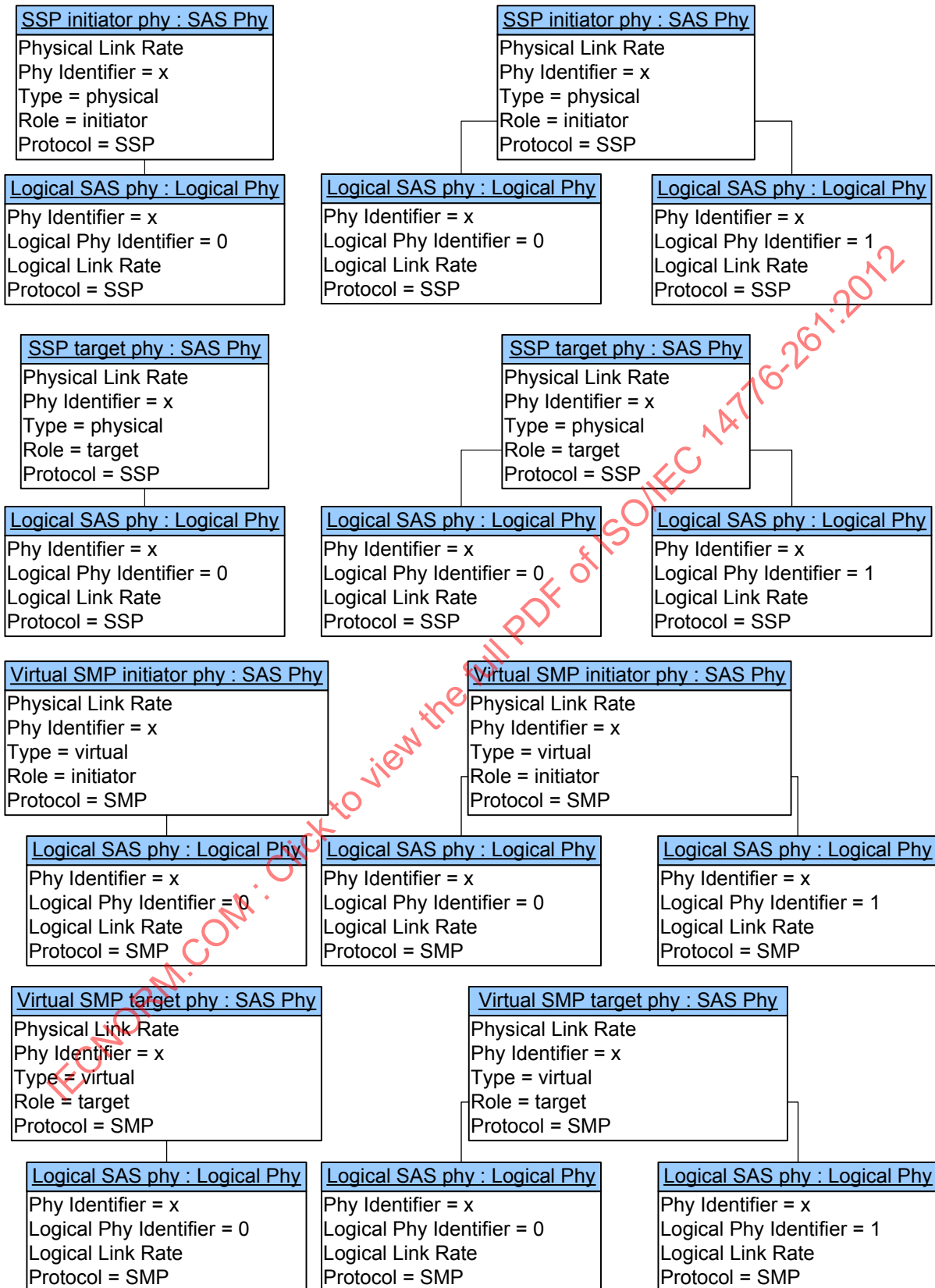


Figure 13 — SAS phy object diagram

Figure 14 shows the objects instantiated from the Expander Phy class, including:

- a) expander phy;
- b) virtual expander phy; and
- c) logical expander phy.

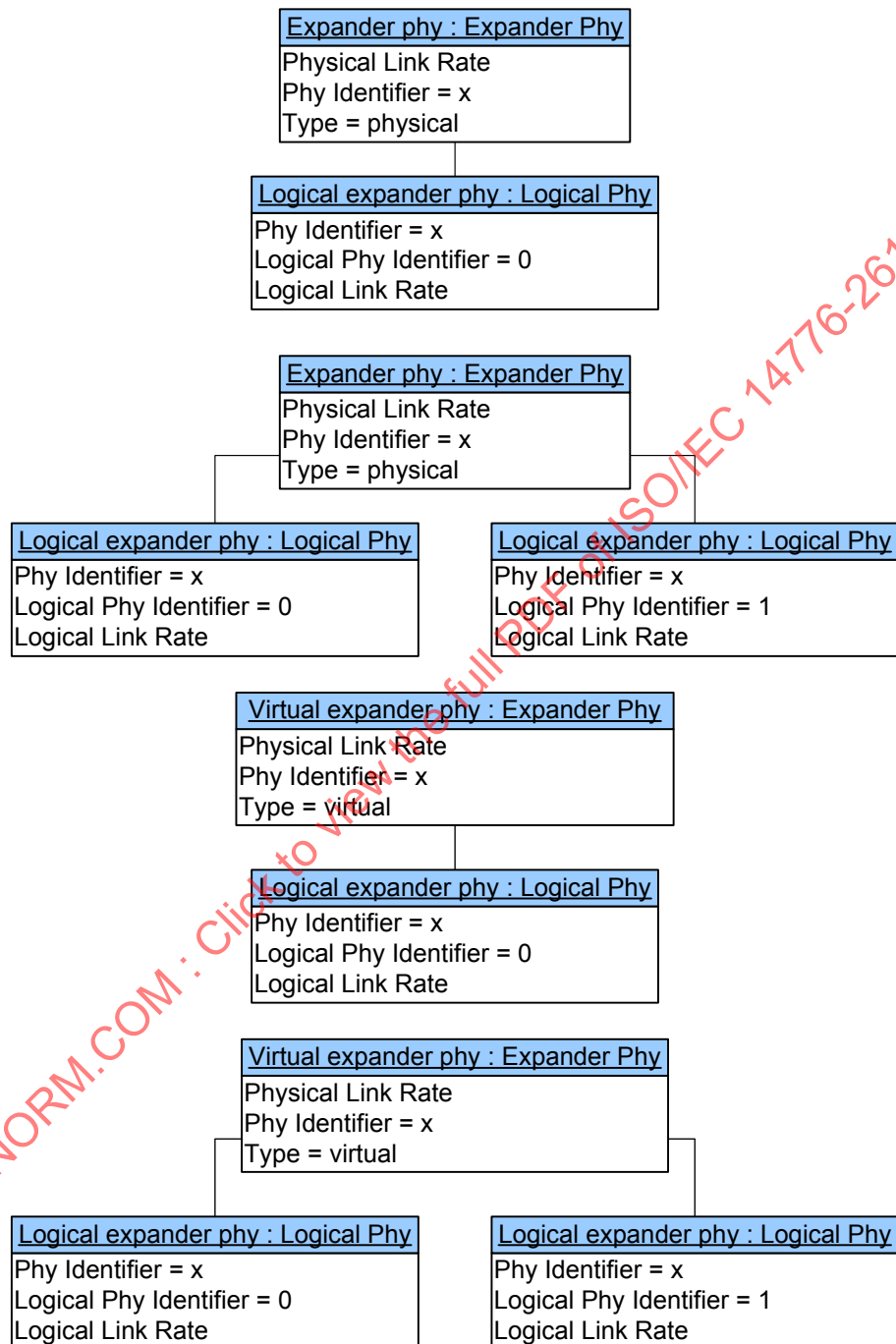


Figure 14 — Expander phy object diagram

4.1.3 Logical links

A physical link with a physical link rate greater than 1.5 Gbps may be multiplexed into two logical links as defined in table 6.

Table 6 — Logical links

Physical link rate	Logical link(s)
6 Gbps	One 6 Gbps logical link
	Two 3 Gbps logical links
3 Gbps	One 3 Gbps logical link
	Two 1.5 Gbps logical links
1.5 Gbps	One 1.5 Gbps logical link

Multiplexing is defined in 5.11.

4.1.4 Ports (narrow ports and wide ports)

A port contains one or more phys. Ports in a device are associated with physical phys based on the identification sequence (see 6.9). Ports are associated with virtual phys based on the design of the device.

A port is a narrow port if there is only one phy in the port.

A port is a wide port if there is more than one phy in the port.

A narrow port is created after transmitting and receiving SAS addresses, unless a wide port is created.

A wide port is created from two or more physical phys if, during the identification sequence (see 6.9), the phys:

- transmitted the same SAS address (see 4.2.4) that the other physical phys in that port also transmitted in their outgoing IDENTIFY address frames (i.e., the SAS address is the same); and
- received the same SAS address that the other physical phys in that port also received in their incoming IDENTIFY address frames (i.e., the attached SAS address is the same).

A narrow link is the physical link that attaches a narrow port to another narrow port. A wide link is the set of physical links that attach a wide port to another wide port.

Attaching a phy within a wide port to another phy in the same port (i.e., the SAS address transmitted in the outgoing IDENTIFY address frame is the same as the SAS address received in the incoming IDENTIFY address frame) is outside the scope of this standard.

Phys that are able to become part of the same wide port shall set the following bits and fields in the IDENTIFY address frame (see 6.8.2) transmitted during the identification sequence to the same set of values on each phy:

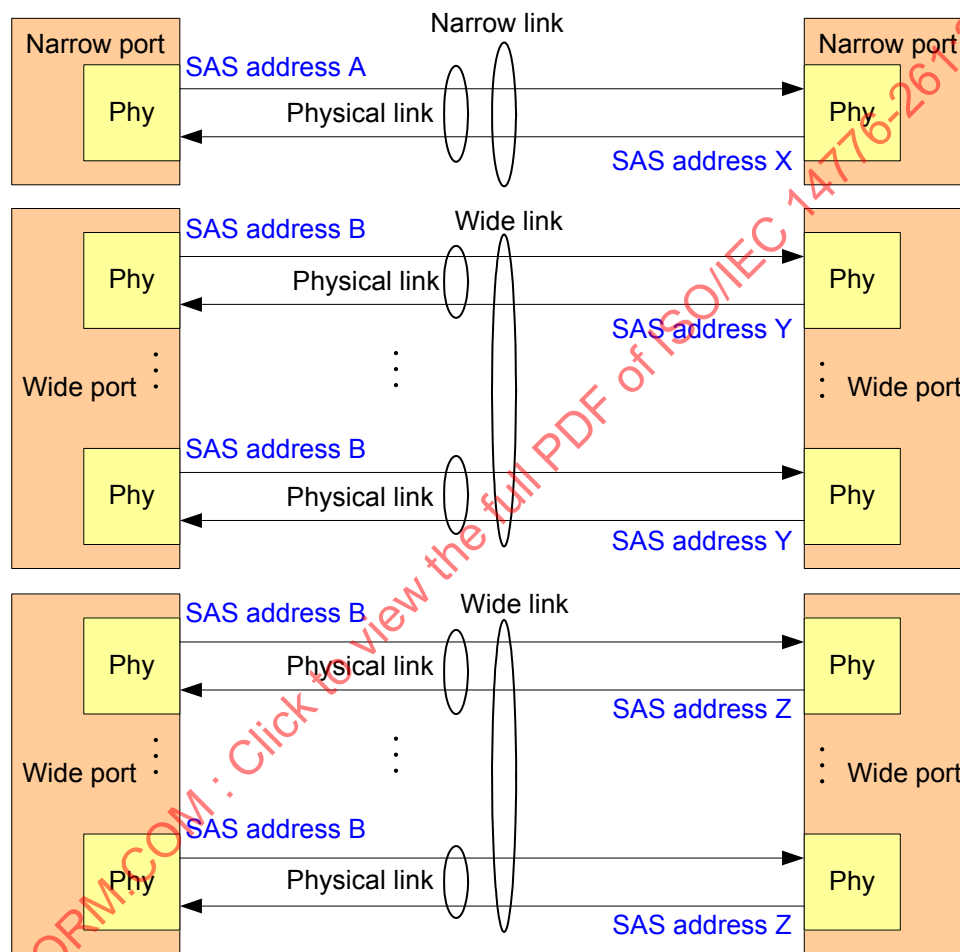
- DEVICE TYPE field;
- BREAK_REPLY CAPABLE bit;
- SSP INITIATOR PORT bit;
- STP INITIATOR PORT bit;
- SMP INITIATOR PORT bit;
- SSP TARGET PORT bit;
- STP TARGET PORT bit;
- SMP TARGET PORT bit;
- SAS ADDRESS field;
- INSIDE ZPSDS persistent bit;
- REQUESTED INSIDE ZPSDS bit;

- l) SLUMBER CAPABLE bit; and
- m) PARTIAL CAPABLE bit.

Recipient wide ports are not required to check the consistency of these fields across their phys.

Each phy in a port may be in a different phy power condition (see 4.10).

Figure 15 shows examples of narrow ports and wide ports, with a representation of the SAS address transmitted during the identification sequence. Although several phys on the left transmit SAS addresses of B, only phys attached to the same SAS addresses become part of the same ports. The set of phys with SAS address B attached to the set of phys with SAS address Y become one port, while the set of phys with SAS address B attached to the set of phys with SAS address Z become another port.



Each horizontal line represents a differential signal pair

Figure 15 — Ports (narrow ports and wide ports)

In figures in this standard that show ports but not phys, the phy level of detail is not shown; however, each port always contains one or more phys.

Figure 16 defines the Port classes, showing the relationships between the following classes:

- a) Port;
- b) Expander Port;
- c) SAS Port;
- d) SAS Initiator Port;
- e) SAS Target Port;
- f) SSP Port;
- g) STP Port; and

h) SMP Port.

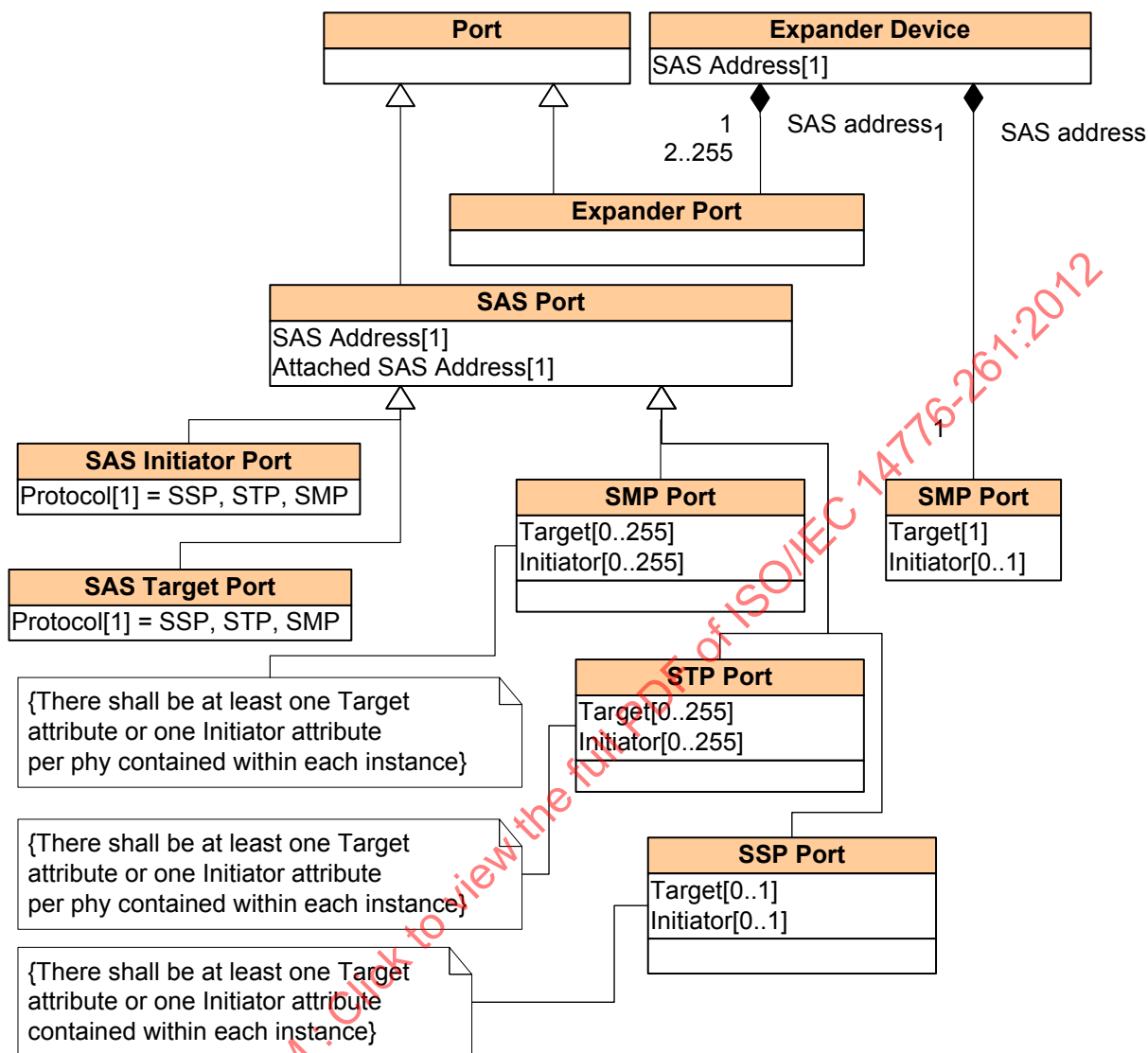


Figure 16 — Port class diagram

Figure 17 shows the objects instantiated from the Port classes:

- SAS Target Port class (i.e., SSP target port, STP target port, SMP target port);
- SAS Initiator Port class (i.e., SSP initiator port, STP initiator port, SMP initiator port);
- STP Port class (i.e., STP initiator port, STP target port, STP port);
- SMP Port class (i.e., SMP initiator port, SMP target port, SMP port);
- SSP Port class (i.e., SSP initiator port, SSP target port, SSP port);
- Expander Device SMP Port class (i.e., SMP target port, SMP port); and
- Expander Port class (i.e., expander port).

Port objects remain instantiated even while no connection is open on any of the phys within the port.

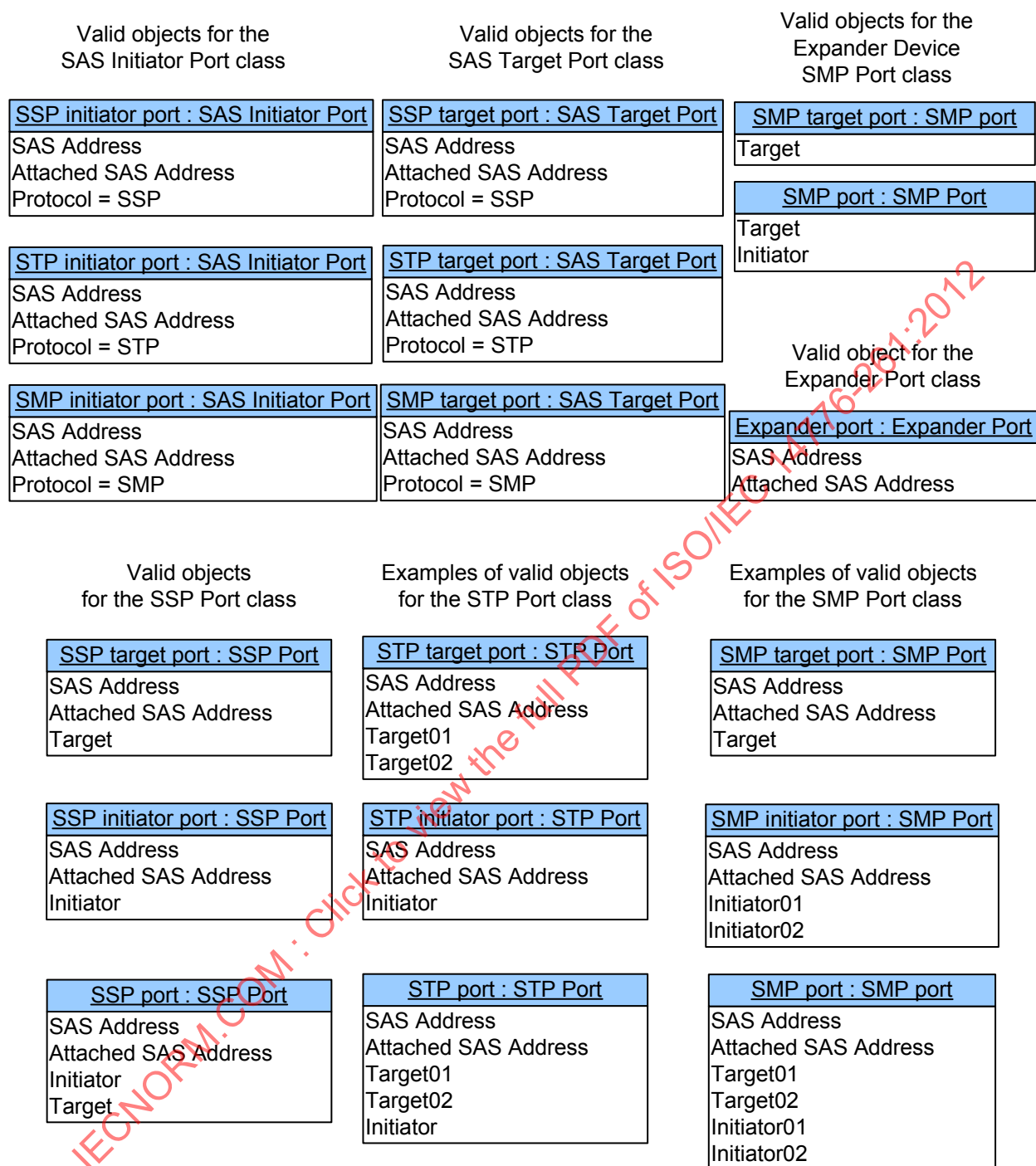


Figure 17 — Port object diagram

4.1.5 Application clients and device servers

This standard defines the following application clients:

- a SCSI application client (see SAM-5) is the source of SCSI commands and task management function requests. A SCSI application client uses an SSP initiator port to interface to a service delivery subsystem;

- b) an ATA application client (see ATA8-AAM) is the source of ATA commands and device management operation requests. An ATA application client uses an STP initiator port to interface to a service delivery subsystem; and
- c) a management application client is the source of management function requests. A management application client uses an SMP initiator port to interface to a service delivery subsystem.

This standard defines the following device servers:

- a) a SCSI device server (see SAM-5) processes SCSI commands. A SCSI device server uses an SSP target port to interface to a service delivery subsystem;
- b) an ATA device server (see ATA8-AAM) processes ATA commands. An ATA device server uses an STP target port to interface to a service delivery subsystem; and
- c) a management device server processes management functions. A management device server uses an SMP target port to interface to a service delivery subsystem.

A SCSI to ATA translation layer (see SAT-2) may be implemented to enable SCSI application clients to communicate with ATA devices.

4.1.6 SAS devices

A SAS device contains one or more SAS ports, each containing one or more phys (i.e., a SAS port may be a narrow port or a wide port).

Figure 18 shows examples of SAS devices with different port and phy configurations.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

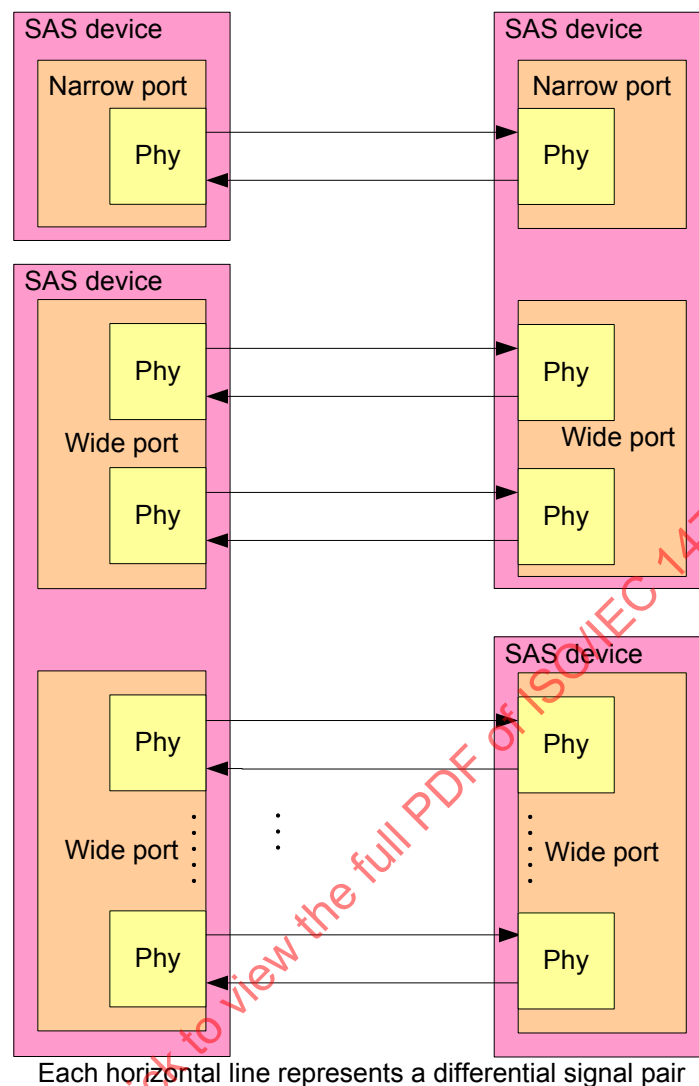


Figure 18 — SAS devices

An end device is a SAS device that is not contained in an expander device (see 4.1.7).

4.1.7 Expander devices

Expander devices are part of a service delivery subsystem and facilitate communication between multiple SAS devices. Expander devices contain two or more external expander ports. Each expander device:

- a) contains one SMP target port and one management device server;
- b) contains one SMP initiator port and one management application client, if the expander device is self-configuring;
- c) may contain one SMP initiator port and one management application client, if the expander device is not self-configuring; and
- d) may contain SAS devices (e.g., an expander device may include an SSP target port for access to a logical unit with a peripheral device type set to 0Dh (i.e., enclosure services device) (see SPC-4 and SES-3)).

Figure 19 shows an expander device.

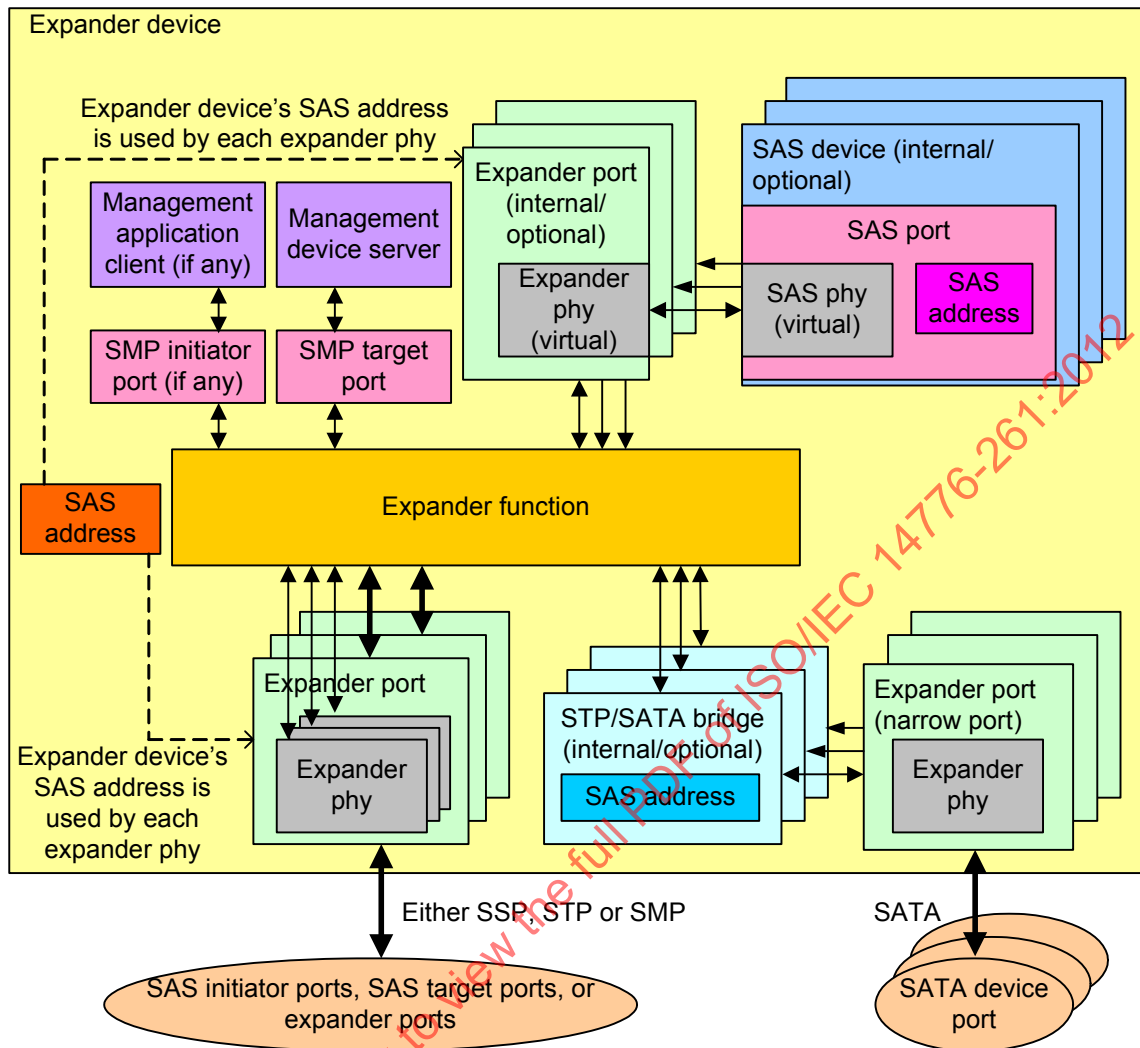


Figure 19 — Expander device

See 4.6 for a detailed model of an expander device.

Each expander phy has one of the following routing attributes (see 4.6.7.1):

- a) direct routing attribute;
- b) table routing attribute; or
- c) subtractive routing attribute.

Expander devices containing expander phys with the table routing attribute also contain an expander route table (see 4.6.7.4). An externally configurable expander device depends on a management application client within the SAS domain to use the discover process (see 4.7) and the configuration subprocess (see 4.8) to configure the expander route table. A self-configuring expander device contains a management application client and an SMP initiator port to perform the discover process (see 4.7) to configure its own expander route table.

4.1.8 Service delivery subsystem

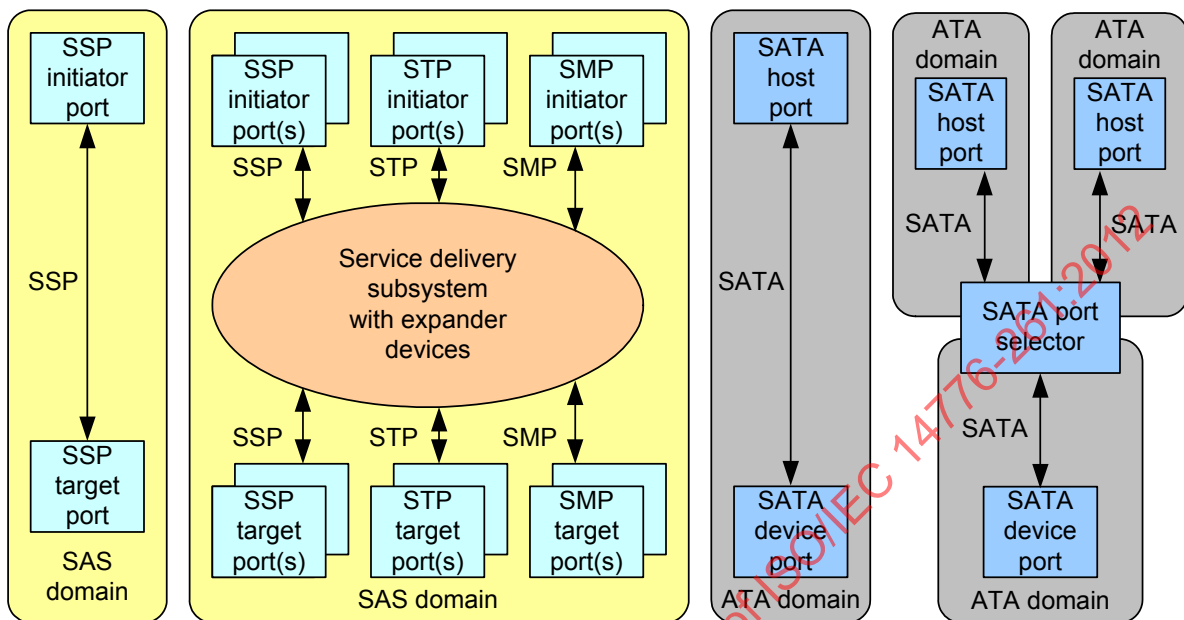
A service delivery subsystem is either:

- a) a set of physical links between a SAS initiator port and a SAS target port; or
- b) a set of physical links and expander devices, supporting more than two SAS ports.

See 4.1.10 for rules on constructing service delivery subsystems from multiple expander devices.

4.1.9 Domains

Figure 20 shows examples of SAS domains and ATA domains.



Note: When expander devices are present, SAS target ports and SAS initiator ports may be located in SAS devices contained in expander devices.

Figure 20 — Domains

Figure 21 shows a SAS domain bridging to one or more ATA domains.

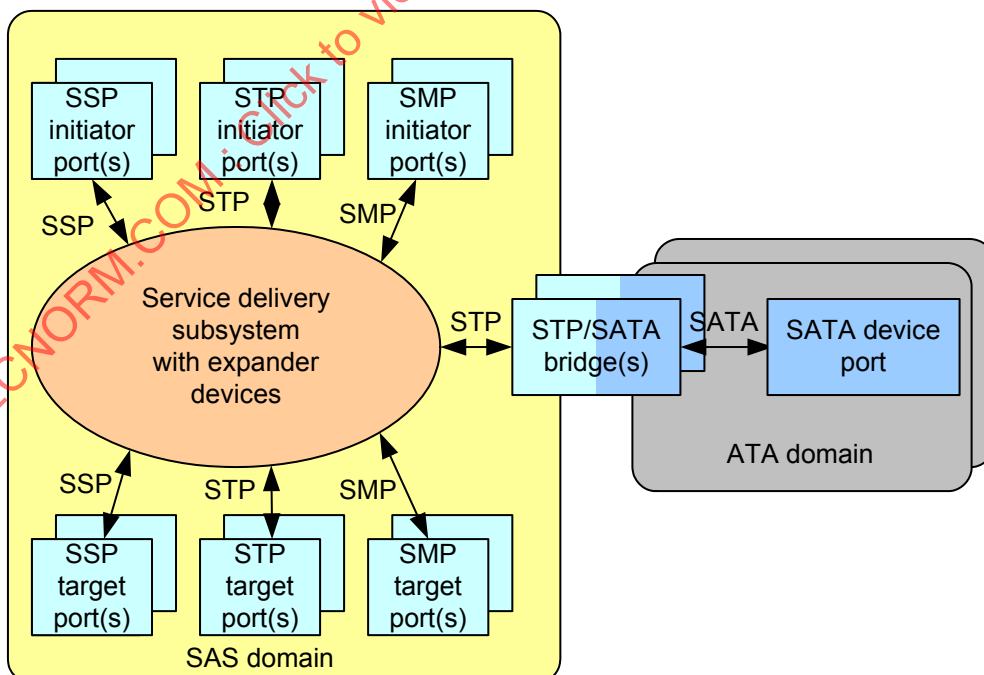


Figure 21 — SAS domain bridging to ATA domains

Figure 22 shows two SAS domains bridging to one or more ATA domains containing SATA devices with SATA port selectors.

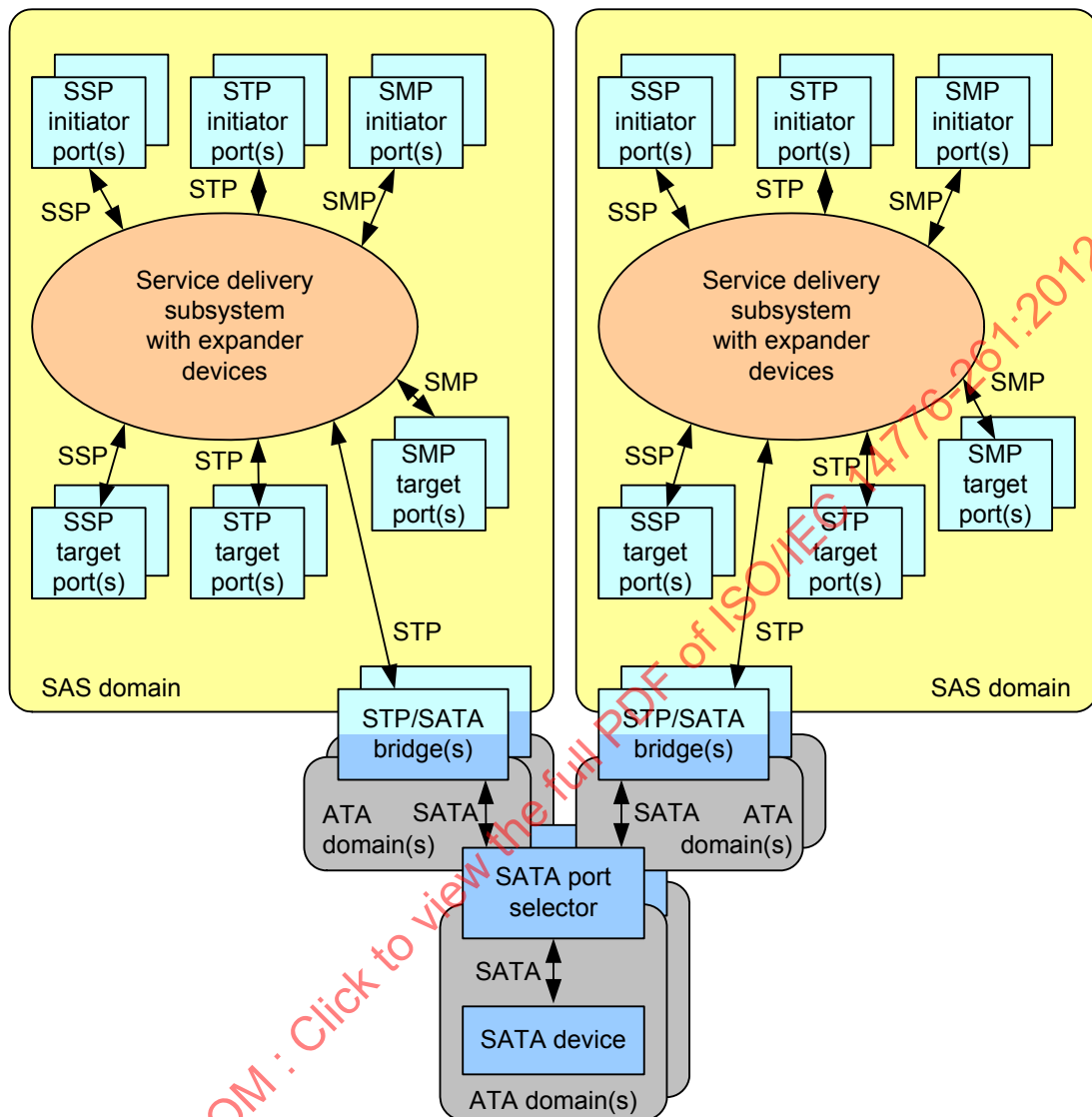


Figure 22 — SAS domains bridging to ATA domains with SATA port selectors

Figure 23 shows SAS initiator devices and SAS target devices with SAS ports in the same SAS domains and in different SAS domains. When a SAS device has ports in the same SAS domain, the ports shall have different SAS addresses. When a SAS device has ports in different SAS domains, the ports may have the same SAS address (see 4.2.4).

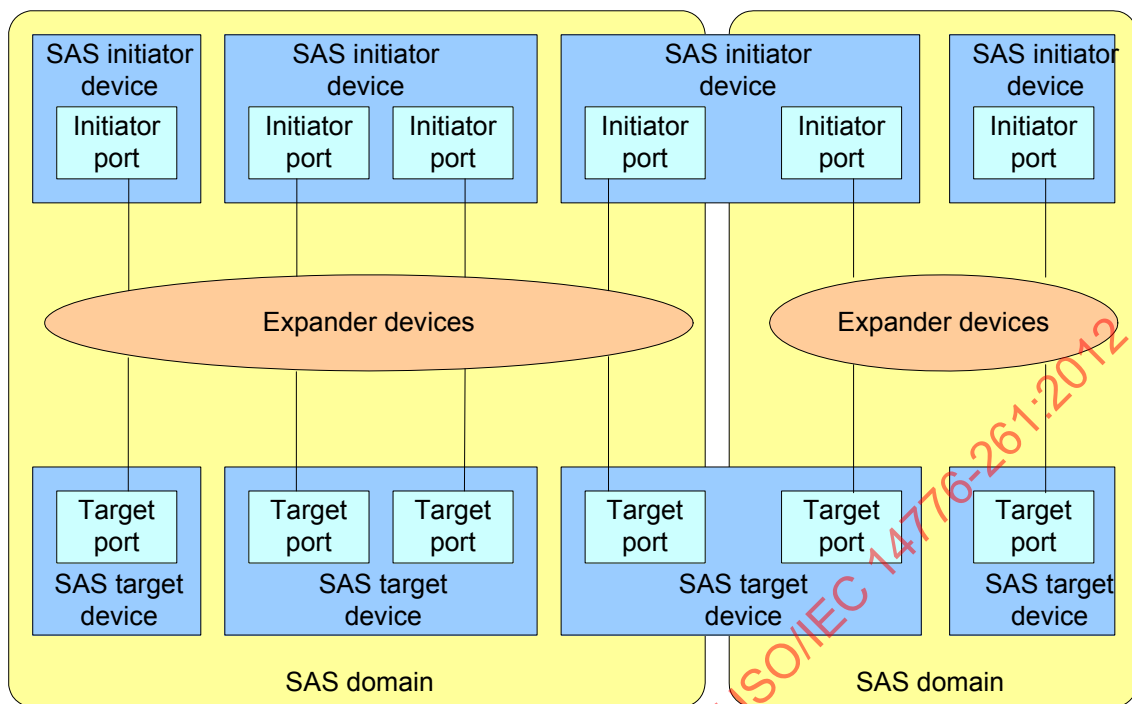


Figure 23 — Devices spanning SAS domains

4.1.10 Expander device topologies

4.1.10.1 Expander device topology overview

More than one expander device may be part of a service delivery subsystem.

To avoid an overflow of an expander route index during the configuration subprocess (see 4.8), a SAS domain containing an externally configurable expander device shall be constructed such that the number of expander route indexes available for each expander phy with the table routing attribute is greater than or equal to the number of SAS addresses addressable through that expander phy.

4.1.10.2 Expander device topologies

Figure 24 shows an example of an expander topology with one expander device.

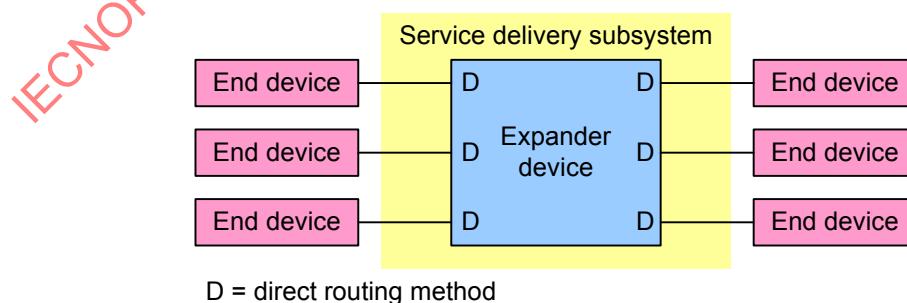


Figure 24 — Single expander device topology example

Figure 25 shows examples of expander topologies with multiple expander devices.

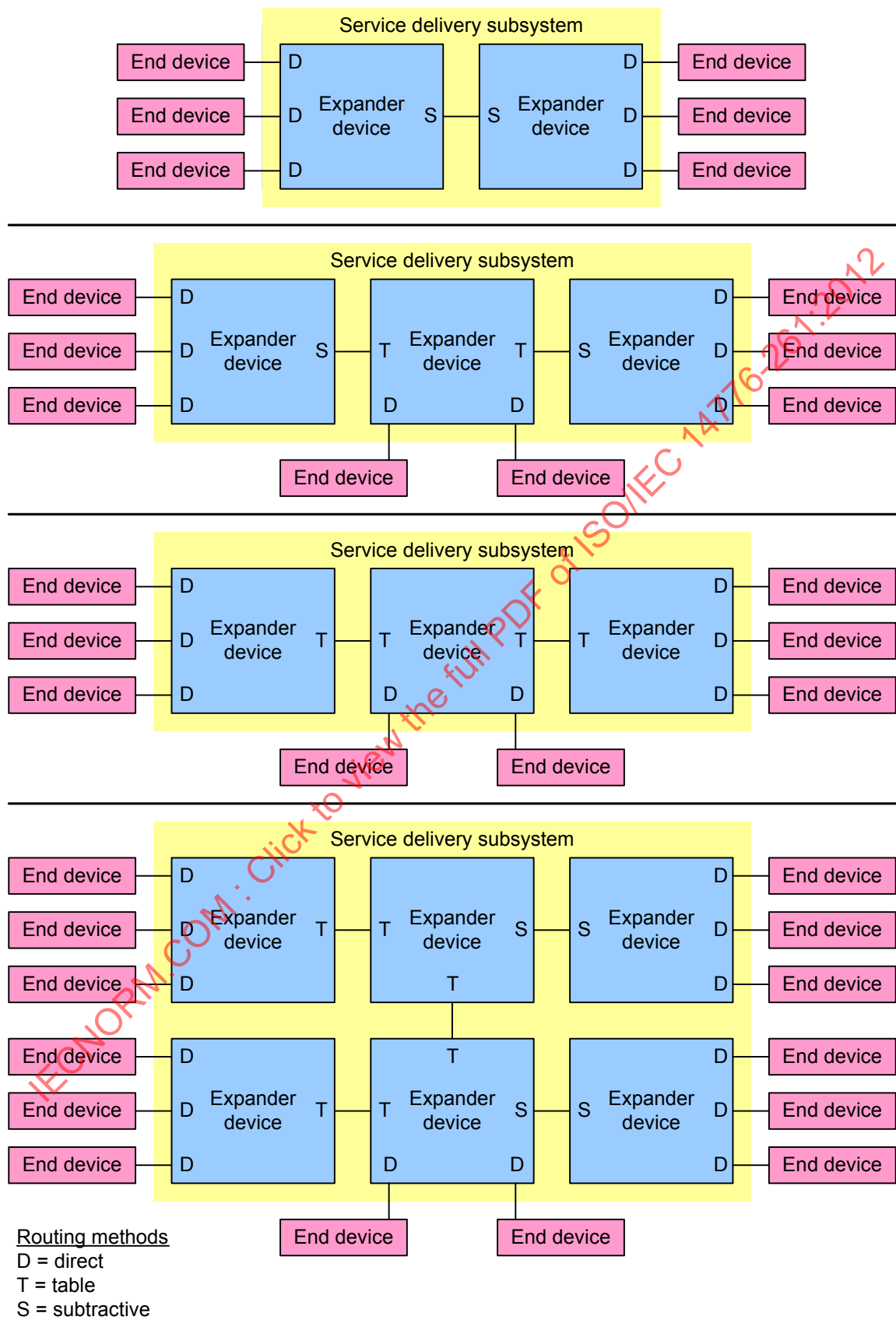


Figure 25 — Multiple expander device topologies

4.1.11 Pathways

A potential pathway is a set of logical links between a SAS initiator phy and a SAS target phy. When a SAS initiator phy is directly attached to a SAS target phy with a non-multiplexed physical link, there is one potential pathway. When the physical link is multiplexed or there are expander devices between a SAS initiator phy and a SAS target phy, it is possible that there is more than one potential pathway, each consisting of a set of logical links between the SAS initiator phy and the SAS target phy. The physical links may or may not be using the same physical link rate.

A pathway is a set of logical links between a SAS initiator phy and a SAS target phy being used by a connection (see 4.1.12).

Figure 26 shows examples of potential pathways.

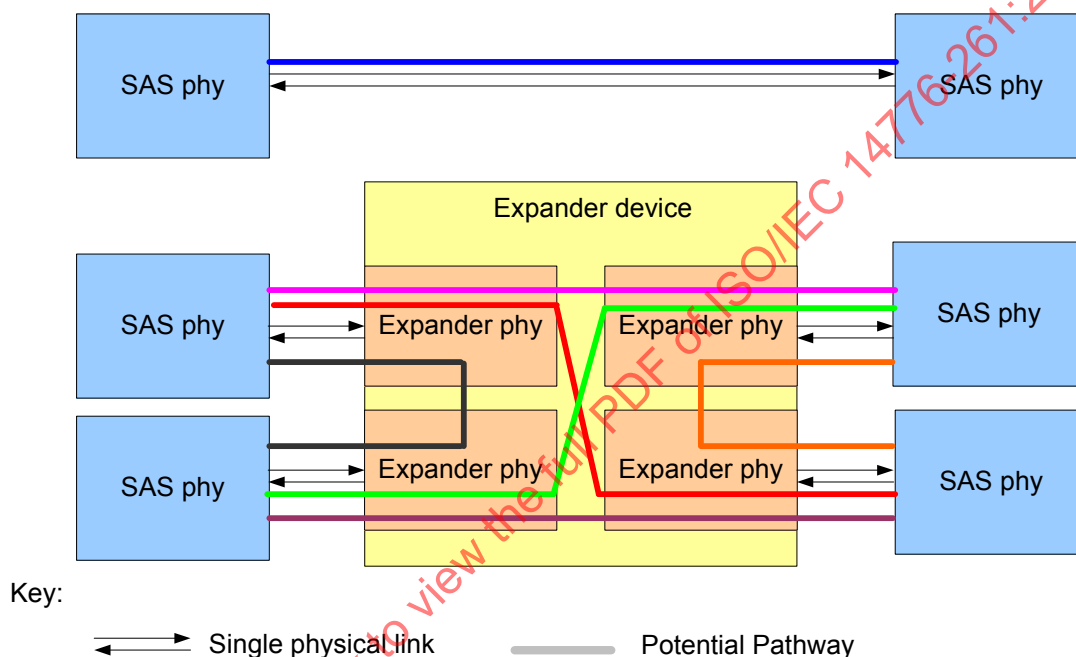


Figure 26 — Potential pathways

A partial pathway is the set of logical links participating in a connection request that have not yet conveyed a connection response (see 6.13).

A partial pathway is blocked when path resources that it requires are held by another partial pathway (see 6.13).

4.1.12 Connections

A connection is a temporary association between a SAS initiator phy and a SAS target phy. During a connection, all dwords from the SAS initiator phy that are not deletable primitives are forwarded to the SAS target phy via a pathway, and all dwords from the SAS target phy that are not deletable primitives are forwarded to the SAS initiator phy via the same pathway. A source phy transmits an OPEN address frame (see 6.8.3) specifying the SAS address of a destination phy to attempt to establish a connection.

A connection is pending when an OPEN address frame has been delivered along a completed pathway to the destination phy but the destination phy has not yet responded to the connection request. A connection is established when the source phy receives an OPEN_ACCEPT (see 6.13) from the destination phy.

A connection enables communication for one protocol: SSP, STP, or SMP. For SSP and STP, connections may be opened and closed multiple times during the processing of a command (see 6.13).

The connection rate is the effective rate of dwords through the pathway between a SAS initiator phy and a SAS target phy, established through the connection request. Every logical phy shall support a 1.5 Gbps connection rate regardless of its logical link rate.

No more than one connection is active on a logical link at a time. If the connection is an SSP or SMP connection and there are no dwords to transmit associated with that connection, idle dwords are transmitted. If the connection is an STP connection and there are no dwords to transmit associated with that connection, then SATA_SYNCs, SATA_CONTs, or vendor specific scrambled data dwords are transmitted as defined in SATA. If there is no connection on a logical link, then idle dwords are transmitted.

The number of connections established by a SAS port shall not exceed the number of SAS logical phys within the SAS port (i.e., only one connection per SAS logical phy is allowed). There shall be a separate connection on each logical link.

If multiple potential pathways exist between the SAS initiator port(s) and the SAS target port(s), multiple connections may be established by a SAS port between the following:

- a) one SAS initiator port to multiple SAS target ports;
- b) one SAS target port to multiple SAS initiator ports; or
- c) one SAS initiator port to one SAS target port.

Once a connection is established, the pathway used for that connection shall not be changed (i.e., all the logical links that make up the pathway remain dedicated to the connection until the connection is closed).

Figure 27 shows examples of connections between wide and narrow ports. All the connections shown may occur simultaneously. For the connections shown in figure 27:

- a) those labeled A, B, and C are an example of one SAS initiator port with connections to multiple SAS target ports;
- b) those labeled E and F are an example of multiple connections between one SAS initiator port and multiple SAS target ports; and
- c) those labeled C, D, E, and F are an example of one SAS initiator port with connections to multiple SAS target ports with one of those SAS target ports having multiple connections with that SAS initiator port.

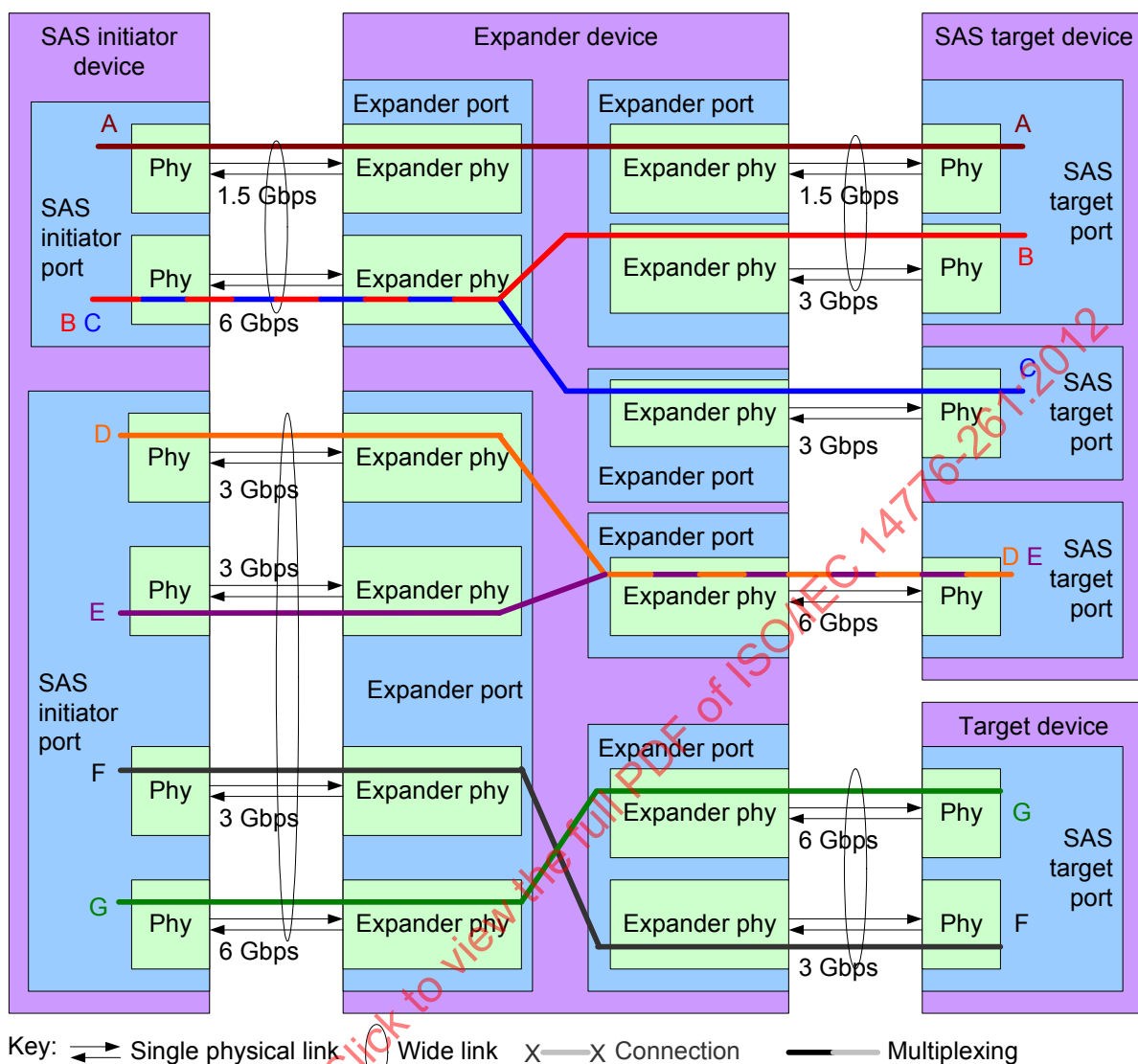


Figure 27 — Multiple connections on wide ports

4.1.13 Broadcasts

Broadcasts are used to notify SAS ports and expander devices in the SAS domain about certain events. Broadcasts are transmitted using BROADCAST (see 6.2.6.4) and/or the SMP ZONED BROADCAST function (see 9.4.3.20).

Table 7 defines the Broadcast types.

Table 7 — Broadcast types (part 1 of 2)

Broadcast	Primitive ^a	Description
Broadcast (Change)	yes	Originated by an expander device to notify SAS initiator ports that a SAS domain change has occurred (see 6.12). May also be originated by SAS initiator ports. SAS target ports shall ignore this Broadcast. See 4.7.2 for application client handling of Broadcast (Change).
Broadcast (Reserved Change 0)	yes	Reserved. SAS ports (i.e, SAS initiator ports and SAS target ports) shall process this Broadcast the same as Broadcast (Change).
Broadcast (Reserved Change 1)	yes	Reserved. SAS ports shall process this Broadcast the same as Broadcast (Change).
Broadcast (SES)	yes	Originated by a logical unit with a peripheral device type set to 0Dh (i.e., enclosure services device) (see SPC-4 and SES-3) accessible through a SAS target port in the SAS domain to notify SAS initiator ports of an asynchronous event. A SCSI application client should poll all the logical units in the SAS domain with peripheral device types set to 0Dh to determine the source. SAS target ports shall ignore this Broadcast.
Broadcast (Expander)	yes	Originated by an expander device to notify SAS initiator ports that an expander event has occurred, including: a) the expander device is going to have reduced functionality for a period of time (see 4.6.8); b) a phy event peak value detector has reached its threshold value; or c) a phy event peak value detector has been cleared by an SMP CONFIGURE PHY EVENT function (see 9.4.3.30). Expander events do not include SAS domain changes, which are communicated with Broadcast (Change). SAS target ports shall ignore this Broadcast. See 4.6.9 for application client handling of Broadcast (Expander).
Broadcast (Asynchronous Event)	yes	Originated by an SSP target port when an event occurs (e.g., a hard reset) that causes one or more unit attention conditions to be established for one or more logical units accessible through the SSP target port. An SSP target port shall only originate one Broadcast (Asynchronous Event) for each event that affects multiple logical units accessible through the SSP target port (e.g., only one Broadcast (Asynchronous Event) is originated when a hard reset occurs). SAS ports other than SSP initiator ports shall ignore this Broadcast.
^a All Broadcasts are supported by the SMP ZONED BROADCAST function (see 9.4.3.20), which defines additional reserved Broadcast types. Broadcasts labeled “yes” are also transmitted via BROADCAST primitive sequences (see 6.2.6.4).		

Table 7 — Broadcast types (part 2 of 2)

Broadcast	Primitive ^a	Description
Broadcast (Reserved 3)	yes	Reserved. SAS ports shall ignore this Broadcast.
Broadcast (Reserved 4)	yes	Reserved. SAS ports shall ignore this Broadcast.
Broadcast (Zone Activate)	no	Initiates the zone activate step (see 4.9.6.4). Devices that are not locked zoning expander devices shall ignore this Broadcast.
^a All Broadcasts are supported by the SMP ZONED BROADCAST function (see 9.4.3.20), which defines additional reserved Broadcast types. Broadcasts labeled “yes” are also transmitted via BROADCAST primitive sequences (see 6.2.6.4).		

When an expander port receives a Broadcast, the BPP (see 4.6.5) shall forward the Broadcast on at least one phy in each other expander port if zoning is disabled, or forward the Broadcast as described in 4.9.5 if zoning is enabled.

An expander device is not required to queue multiple identical Broadcasts for the same expander port. If a second identical Broadcast is requested before the first Broadcast has been transmitted, then the second Broadcast may be ignored.

A SAS device or expander device may implement counters for Broadcasts it originates and report them in the REPORT BROADCAST response (see 9.4.3.9). If counters are supported, then the SAS device or expander device shall, for each combination of Broadcast type and Broadcast reason that the SAS device or expander device supports:

- a) if the Broadcast is related to a phy, then maintain a separate Broadcast counter for each phy; or
- b) if the Broadcast is not related to a phy, then maintain one originated Broadcast counter.

Broadcast (Change)s originated by an expander device are counted and reported in the REPORT GENERAL response (see 9.4.3.4) and other SMP response frames containing an EXPANDER CHANGE COUNT field.

An expander device may implement counters for Broadcasts received from attached end devices and report them in the REPORT BROADCAST response (see 9.4.3.9).

A SAS device or an expander device is not required to maintain originated Broadcast count information in non-volatile storage or across reset events.

See 4.13 for details on phy events.

4.2 Names and identifiers

4.2.1 Names and identifiers overview

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

Table 8 describes the definitions of names and identifiers for SAS.

Table 8 — Names and identifiers

Attribute	Format	SAS usage	Reference(s)
Device name	SAS address (see 4.2.4) for SAS devices and expander devices. NAA IEEE Registered format (see 4.2.2) for SATA devices with worldwide names. ^a	Reported in: a) the IDENTIFY address frame (see 6.8.2) DEVICE NAME field; b) the Device Identification VPD page (see 9.2.11.2); and c) the DISCOVER response (see 9.4.3.10) ATTACHED DEVICE NAME field.	4.2.6 and 4.2.7
Port name	Not defined		4.2.8
Port identifier	SAS address (see 4.2.4)	Reported in: a) the IDENTIFY address frame (see 6.8.2) for SAS ports; and b) the Device Identification VPD page (see 9.2.11.2).	4.2.9
Phy identifier	8-bit value	Phy identifier	4.2.10

^a For SATA devices without worldwide names, a device name may be set in the PHY CONTROL request (see 9.4.3.28) ATTACHED DEVICE NAME field.

Table 9 describes how various SAM-5 attributes are implemented in SSP.

Table 9 — SAM-5 attribute mapping

SAM-5 attribute	SSP implementation
Initiator port identifier	SAS address of an SSP initiator port
Initiator port name	Not defined
Target port identifier	SAS address of an SSP target port
Target port name	Not defined
SCSI device name	Device name of SAS device containing an SSP port

4.2.2 NAA IEEE Registered format identifier

Table 10 defines the NAA IEEE Registered format identifier used by device names and port identifiers. This format is the same as that defined in SPC-4.

Table 10 — NAA IEEE Registered format

Byte\Bit	7	6	5	4	3	2	1	0
0	NAA (5h)				(MSB)			
1								
2	IEEE COMPANY ID							

Table 10 — NAA IEEE Registered format

Byte\Bit	7	6	5	4	3	2	1	0
3				(LSB)				
4								
5								
6								
7								

The NAA field is set to the value defined in table 10.

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

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

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

NOTE 9 Information about IEEE company identifiers may be obtained from the IEEE Registration Authority web site (see <http://standards.ieee.org/regauth/oui>).

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

4.2.3 NAA Locally Assigned format identifier

Table 11 defines the NAA Locally format identifier used by device names and port identifiers. This format is the same as that defined in SPC-4.

Table 11 — NAA Locally Assigned format

Byte\Bit	7	6	5	4	3	2	1	0
0								
1								
2								
3								
4								
5								
6								
7								

The NAA field is set to the value defined in table 11.

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

4.2.4 SAS address

A SAS address is an identifier using either:

- a) the NAA IEEE Registered format (see 4.2.2); or
- b) the NAA Locally Assigned format (see 4.2.3).

A SAS address should use the NAA IEEE Registered format (see 4.2.2).

A SAS address of 00000000 00000000h indicates an invalid identifier.

4.2.5 Hashed SAS addresses

SSP frames include hashed versions of SAS addresses of SAS ports to provide an additional level of verification of proper frame routing.

The code used for the hashing algorithm is a cyclic binary Bose, Chaudhuri, and Hocquenghem (BCH) (63, 39, 9) code. Table 12 lists the parameters for the code.

Table 12 — Hashed SAS address code parameters

Parameter	Value
Number of bits per codeword	63
Number of data bits	39
Number of redundant bits	24
Minimum distance of the code	9

The generator polynomial for this code is:

$$G(x) = (x^6 + x + 1) (x^6 + x^4 + x^2 + x + 1) (x^6 + x^5 + x^2 + x + 1) (x^6 + x^3 + 1)$$

After multiplication of the factors, the generator polynomial is:

$$G(x) = x^{24} + x^{23} + x^{22} + x^{20} + x^{19} + x^{17} + x^{16} + x^{13} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1$$

Annex D contains additional information on SAS address hashing.

4.2.6 Device names and expander device SAS addresses

Each expander device and SAS device shall include a SAS address (see 4.2.4) as its device name.

A SAS address used as a device name shall not be used as any other name or identifier (e.g., a device name, port name, port identifier, or logical unit name (see SAM-5)), except:

- a) the SAS address of an expander device is the same as the SAS address of the SMP port in that expander device.

SAS devices and expander devices report their device names in the IDENTIFY address frame (see 6.8.2).

Logical units accessed through SSP target ports report SAS target device names through SCSI vital product data (see 9.2.11).

See Annex I for more information on SAS addresses.

4.2.7 Device name for SATA devices with world wide names

Table 13 defines the NAA IEEE Registered format identifier (see 4.2.2) used by device names for SATA devices that provide world wide names in their IDENTIFY (PACKET) DEVICE data.

Table 13 — Device name created from the IDENTIFY (PACKET) DEVICE world wide name

Subformat field name ^a	Specific bits in	Contents ^b
NAA	Byte 0 bits 7:4	IDENTIFY (PACKET) DEVICE data word 108 bits 15:12 ^c
IEEE COMPANY_ID	Byte 0 bits 3:0	IDENTIFY (PACKET) DEVICE data word 108 bits 11:8
	Byte 1	IDENTIFY (PACKET) DEVICE data word 108 bits 7:0
	Byte 2	IDENTIFY (PACKET) DEVICE data word 109 bits 15:8
	Byte 3 bits 7:4	IDENTIFY (PACKET) DEVICE data word 109 bits 7:4
VENDOR SPECIFIC IDENTIFIER	Byte 3 bits 3:0	IDENTIFY (PACKET) DEVICE data word 109 bits 3:0
	Byte 4	IDENTIFY (PACKET) DEVICE data word 110 bits 15:8
	Byte 5	IDENTIFY (PACKET) DEVICE data word 110 bits 7:0
	Byte 6	IDENTIFY (PACKET) DEVICE data word 111 bits 15:8
	Byte 7	IDENTIFY (PACKET) DEVICE data word 111 bits 7:0
^a See table 10 in 4.2.2. ^b IDENTIFY (PACKET) DEVICE data words 108-111 contain the world wide name field (see ACS-2). ^c This 4-bit field is required to be set to 5h (i.e., IEEE Registered) by ACS-2.		

4.2.8 Port names

Port names are not defined in SAS.

See Annex I for more information on SAS addresses.

4.2.9 Port identifiers and SAS port SAS addresses

Each SAS port (e.g., including the STP target port in each STP/SATA bridge) shall include a SAS address (see 4.2.4) as its port identifier.

A SAS address used as a port identifier shall not be used as any other name or identifier (e.g., a device name, port name, or logical unit name (see SAM-5)) except:

- a) a SAS address may be used as a port identifier in one or more other SAS domains (see 4.1.4); and
- b) the SAS address of an SMP port in an expander device is the same as the SAS address of the expander device containing that SMP port.

Expander ports do not have port identifiers.

SAS ports in end devices report their port identifiers in the IDENTIFY address frame (see 6.8.2). Expander devices containing SAS ports (e.g., SAS ports attached to virtual phys, or STP target ports in STP/SATA bridges) report the port identifiers of those SAS ports in the SMP DISCOVER response (see 9.4.3.10).

Port identifiers are used as source and destination SAS addresses in OPEN address frames (see 6.8.3).

Logical units accessed through SSP target ports report SAS target port identifiers through SCSI vital product data (see 9.2.11).

See Annex I for more information on SAS addresses.

4.2.10 Phy identifiers

Each SAS phy and expander phy shall be assigned an identifier called a phy identifier that is unique within the SAS device and/or expander device. Each SAS logical phy within a SAS phy shall use the same phy identifier. Each expander logical phy within an expander phy shall use the same phy identifier. The phy identifier is used for management functions (see 9.4).

Phy identifiers shall be greater than or equal to 00h and less than or equal to FEh (i.e., 254) and should be numbered starting with 00h. In an expander device or in a SAS device containing an SMP target port, phy identifiers shall be less than the value of the NUMBER OF PHYS field in the SMP REPORT GENERAL response (see 9.4.3.4). In a SAS device containing an SSP target port, phy identifiers shall be less than the value of the NUMBER OF PHYS field in the Phy Control And Discover mode page (see 9.2.7.5).

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

4.3 State machines

4.3.1 State machine overview

Figure 28 shows the state machines for SAS devices and their relationships to each other and to the SAS Device, SAS Port, and SAS Phy classes.

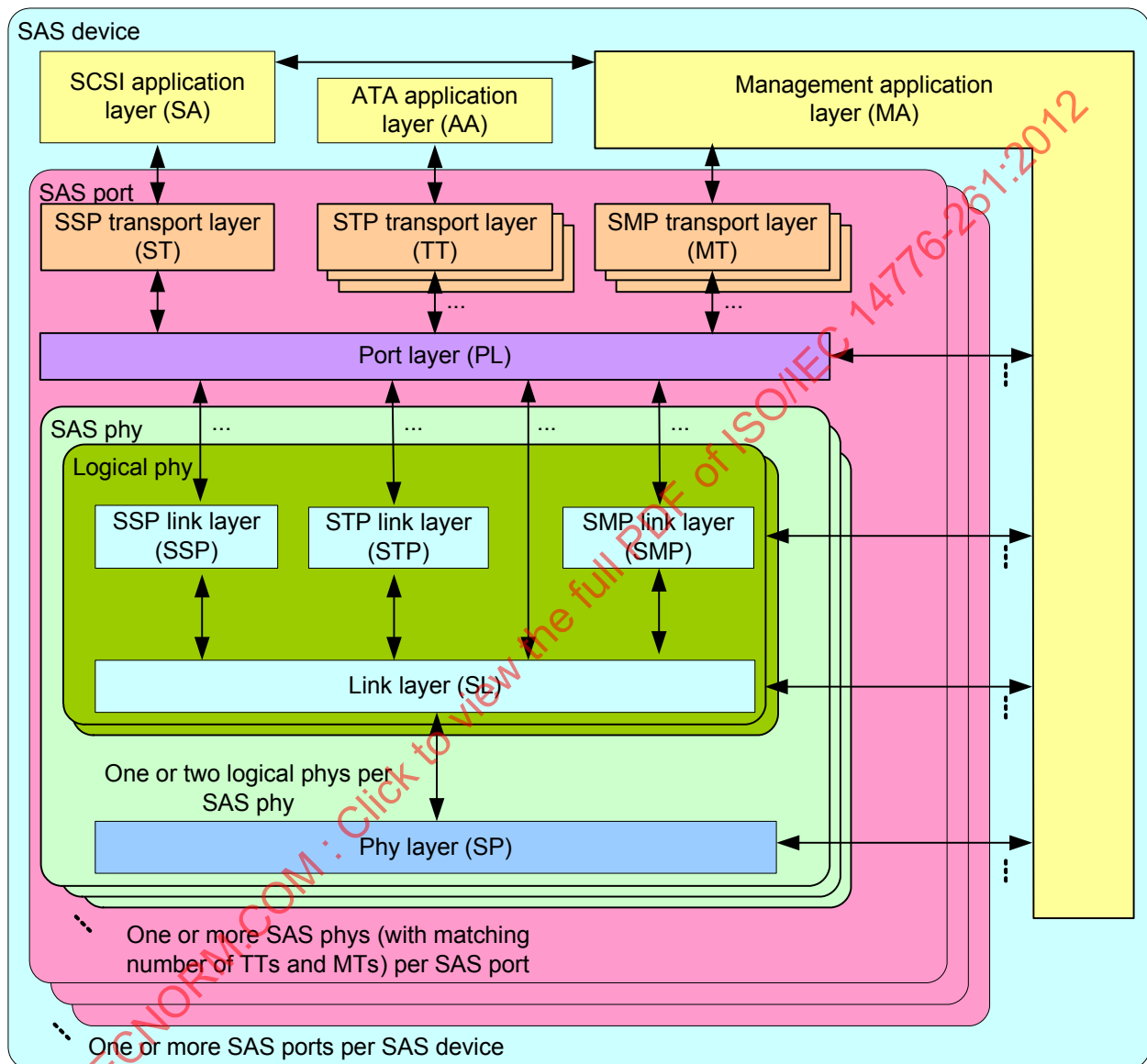
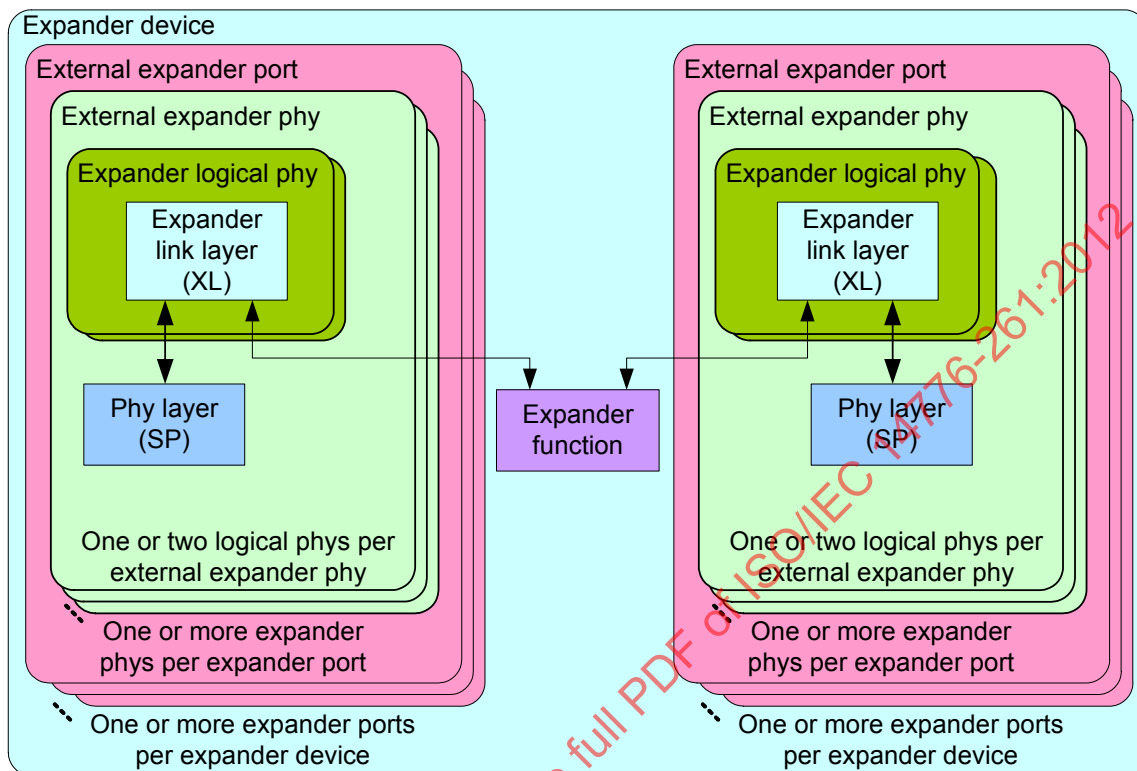


Figure 28 — State machines for SAS devices

Figure 29 shows the state machines for expander devices and their relationships to each other and to the Expander Device, Expander Port, and Expander Phy classes. Expander function state machines are not defined in this standard, but the interface to the expander function is defined in 4.6.6.



Note: The expander link layer includes the SL_IR state machines.

Figure 29 — State machines for expander devices

4.3.2 Transmit data path

Figure 30 shows the transmit data path in a SAS phy, showing the relationship between:

- the SP state machine (see 5.9) and the SP transmitter (see 5.9.2);
- multiplexing (see 5.11);
- the SL_IR state machines (see 6.10) and the SL_IR transmitter (see 6.10.2);
- physical link rate tolerance management (see 6.3);
- the SL state machines (see 6.15) and the SL transmitter (see 6.15.2);
- rate matching (see 6.14); and
- the SSP transmit data path (see figure 31), SMP transmit data path (see figure 32), and STP transmit data path (see figure 33).

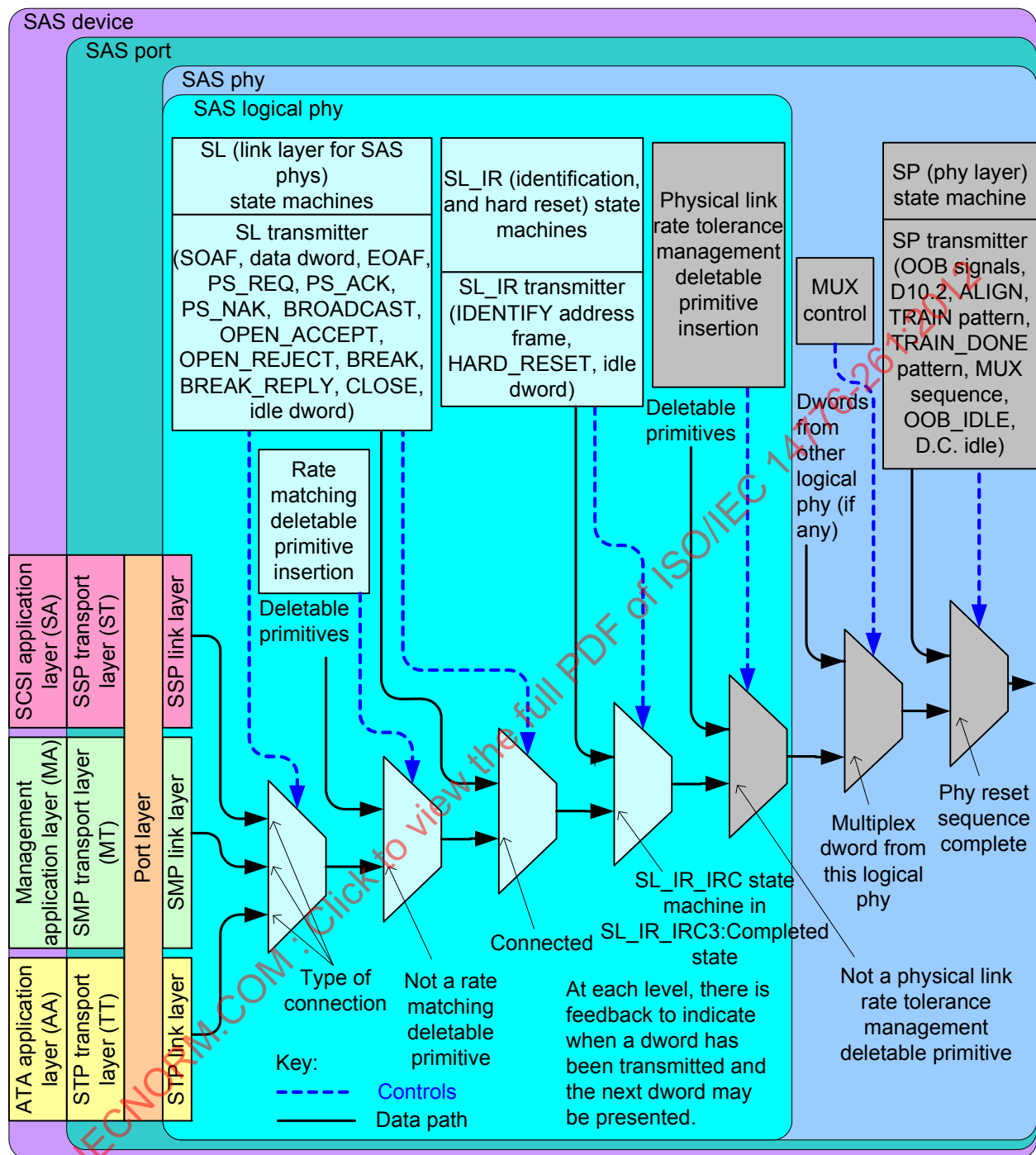


Figure 30 — Transmit data path in a SAS phy

Figure 31 shows the transmit data path for the SSP link layer, including the SSP state machines and the SSP transmitter (see 6.17.8 and 6.17.8.2), and the communication to the port layer, SSP transport layer, and SCSI application layer. Only the SSP link layer (i.e., not the port, transport, or application layer) transmits dwords.

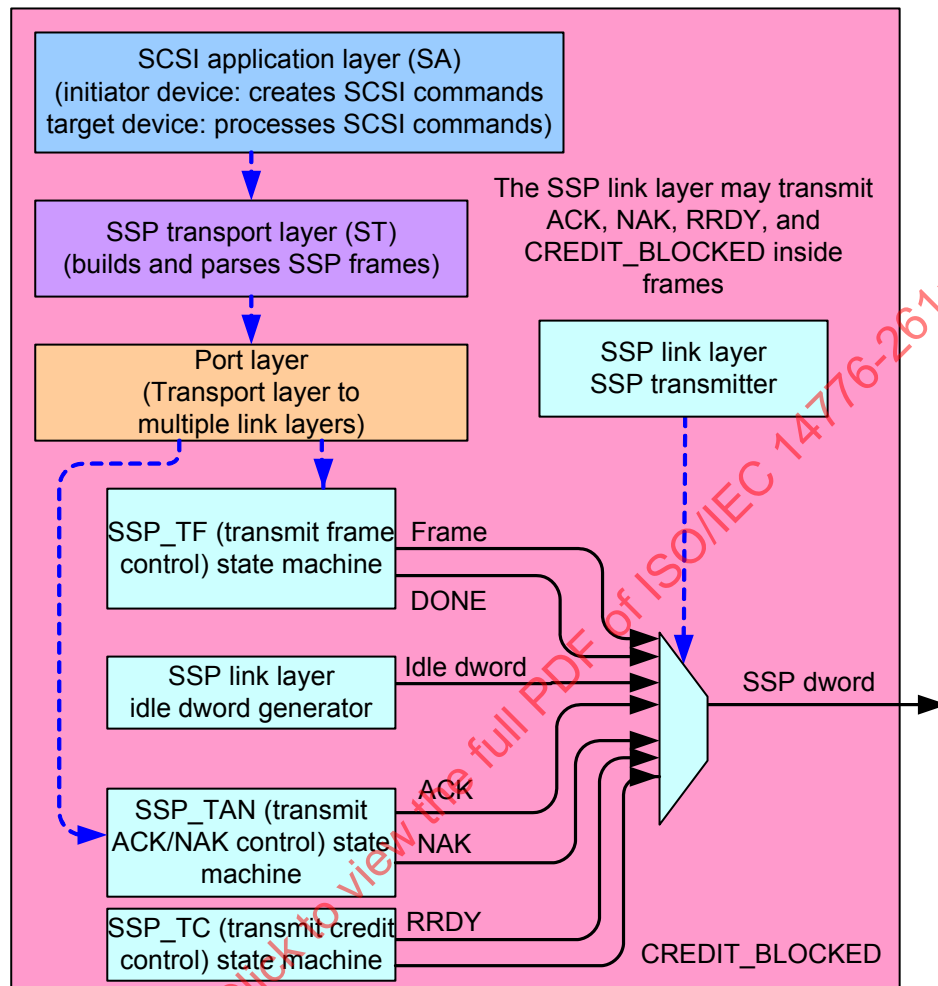


Figure 31 — SSP link, port, SSP transport, and SCSI application layer state machines

Figure 32 shows the transmit data path for the SMP link layer, including the SMP state machines and the SMP transmitter (see 6.19.5 and 6.19.5.2), and the communication to the port layer, SMP transport layer, and management application layer. Only the SMP link layer (i.e., not the port, transport, or application layer) transmits dwords.

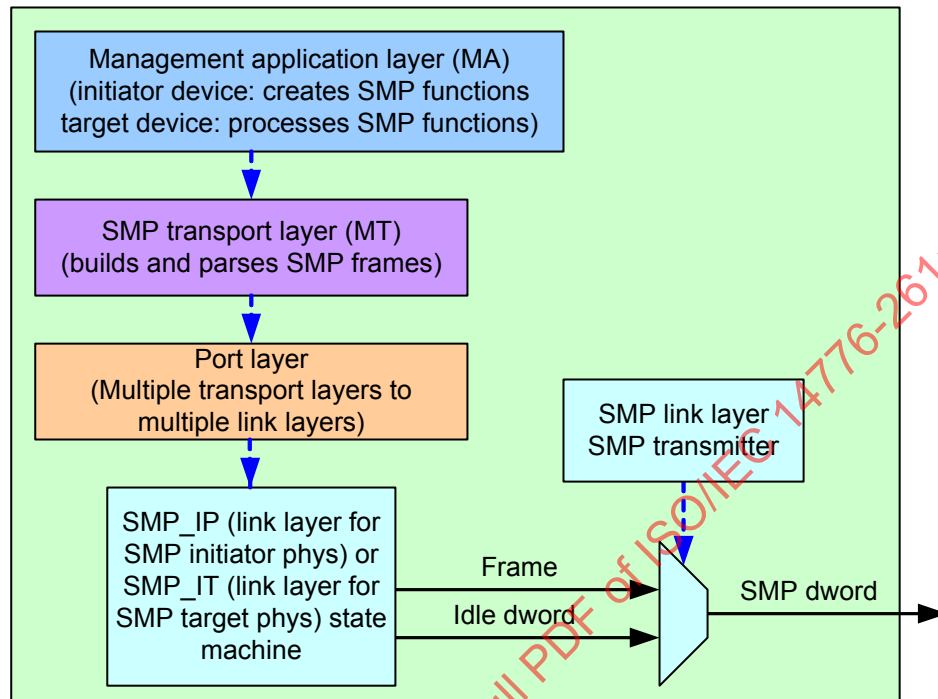


Figure 32 — SMP link, port, SMP transport, and management application layer state machines

Figure 33 shows the transmit data path for the STP link layer, including the SSP state machines and the SSP transmitter (see 6.18.8), and communication to the port layer, STP transport layer, and ATA application layer. Only the STP link layer (i.e., not the port, transport, or application layer) transmits dwords.

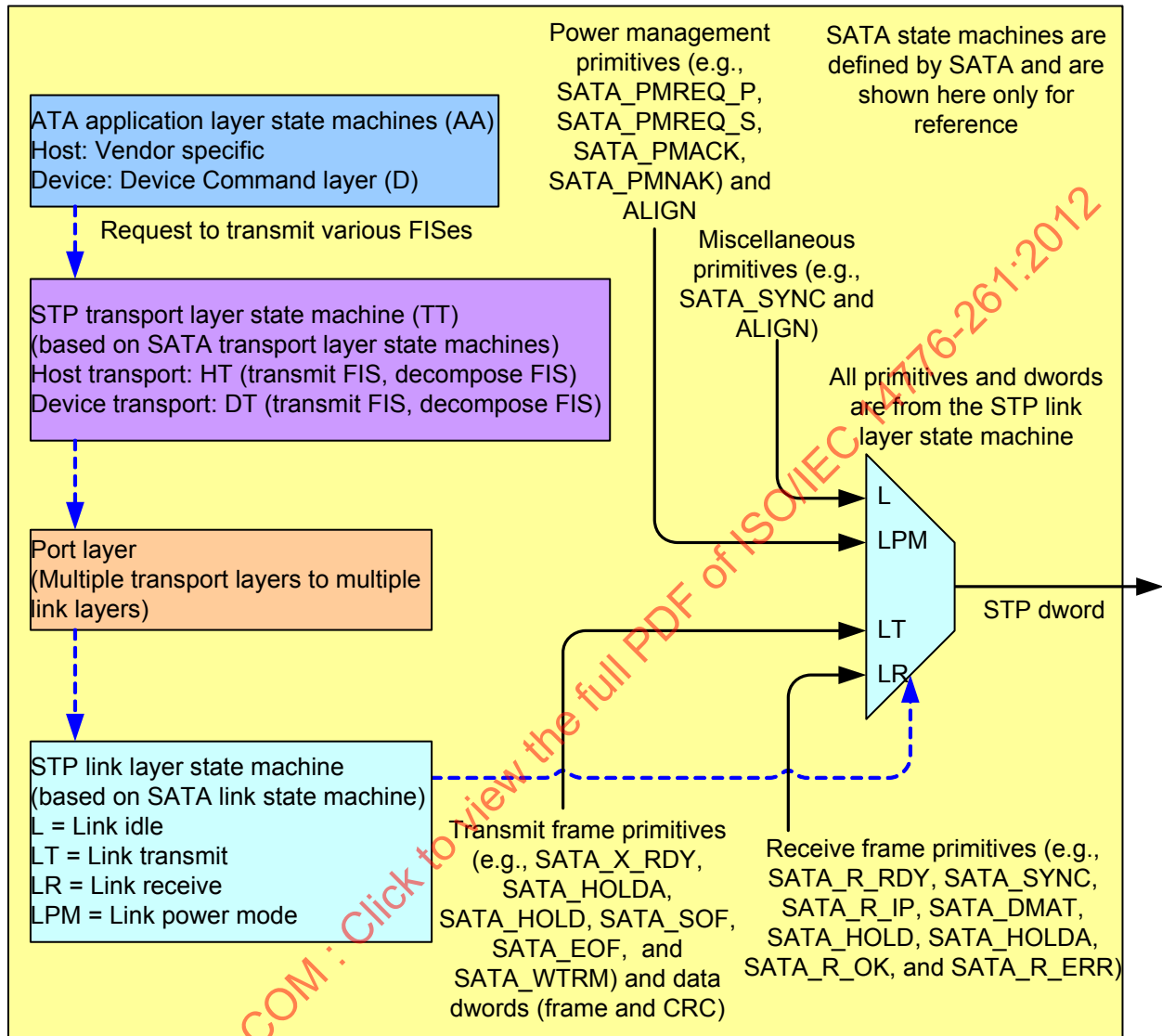


Figure 33 — STP link, port, STP transport, and ATA application layer state machines

Figure 34 shows the transmit data path in an expander phy, showing the relationship between:

- the SP state machine (see 5.9) and the SP transmitter (see 5.9.2);
- multiplexing (see 5.11);
- the SL_IR state machines (see 6.10) and the SL_IR transmitter (see 6.10.2);
- physical link rate tolerance management (see 6.3); and
- the XL state machines (see 6.16) and the XL transmitter (see 6.16.2).

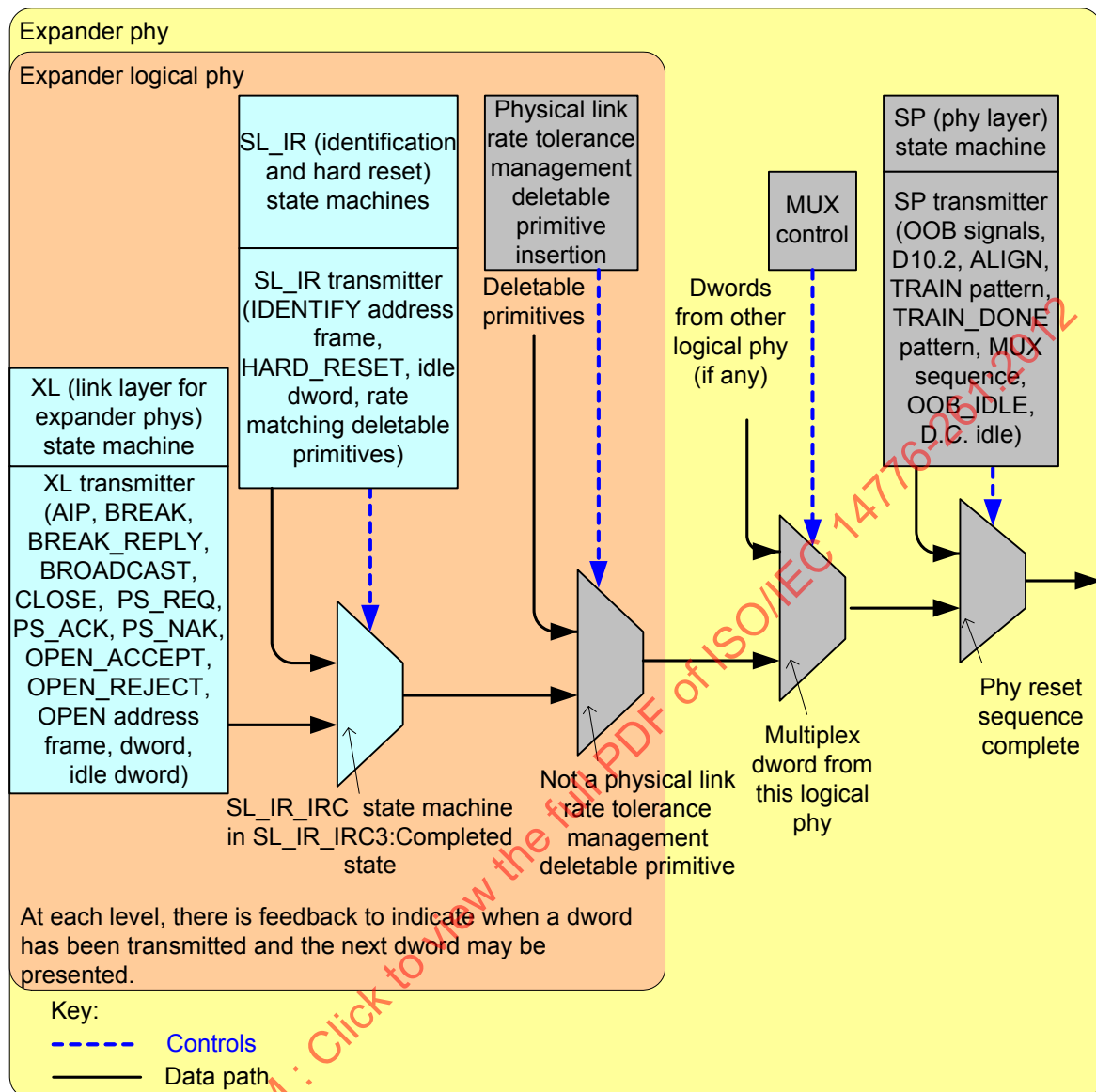


Figure 34 — Transmit data path and state machines in an expander phy

4.3.3 Receive data path

The SP_DWS receiver (see 5.10.2) establishes dword synchronization and sends dwords to the SP_DWS state machine (see 5.10) and to the link layer state machine receivers.

If multiplexing (see 5.11) is enabled (see table 51 in 5.7.4.2.3.3), then the SP_DWS receiver uses incoming MUXs to determine the logical link numbers and route the dwords to the appropriate link layer receivers.

Figure 35 shows the receive data path in a SAS phy, showing the relationship between:

- the SP_DWS state machine (see 5.10) and the SP_DWS receiver (see 5.10.2);
- the SL_IR state machines (see 6.10) and the SL_IR receiver (see 6.10.2);
- the SL state machines (see 6.15) and the SL receiver (see 6.15.2); and
- the SSP receiver (see 6.17.8.2), SMP receiver (see 6.19.5.2), and STP receiver (see 6.18.8).

See figure 94 in 6.3.1 for more information about the elasticity buffer, which is not shown in figure 35.

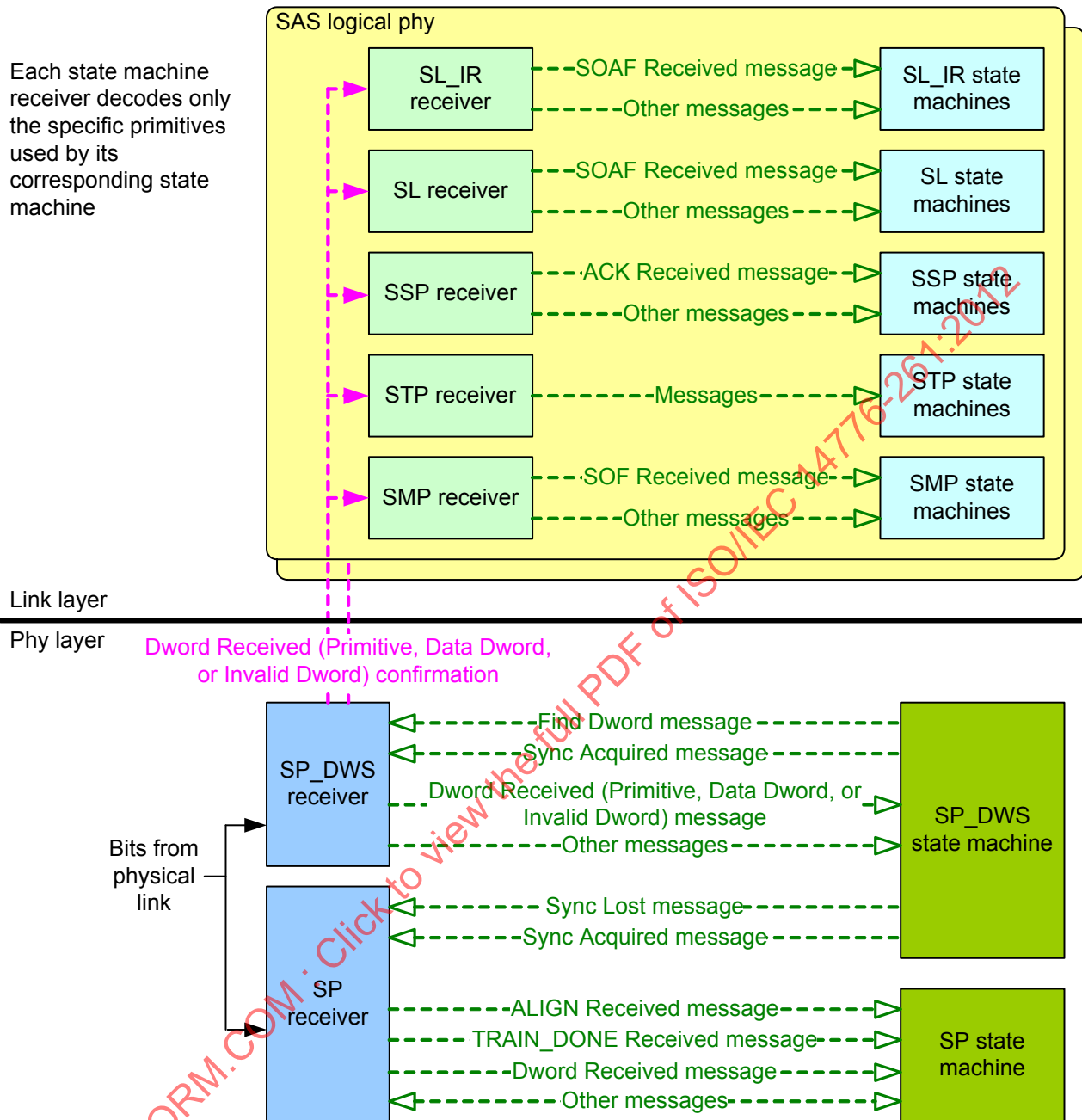


Figure 35 — Receive data path in a SAS phy

Figure 36 shows the receive data path in an expander phy showing the relationship between:

- the SP_DWS state machine (see 5.10) and the SP_DWS receiver (see 5.10.2);
- the SL_IR state machines (see 6.10) and the SL_IR receiver (see 6.10.2); and
- the XL state machines (see 6.16) and the XL receiver (see 6.16.2).

See figure 94 in 6.3.1 for more information about the elasticity buffer, which is not shown in figure 36.

Each state machine receiver decodes only the specific primitives used by its corresponding state machine

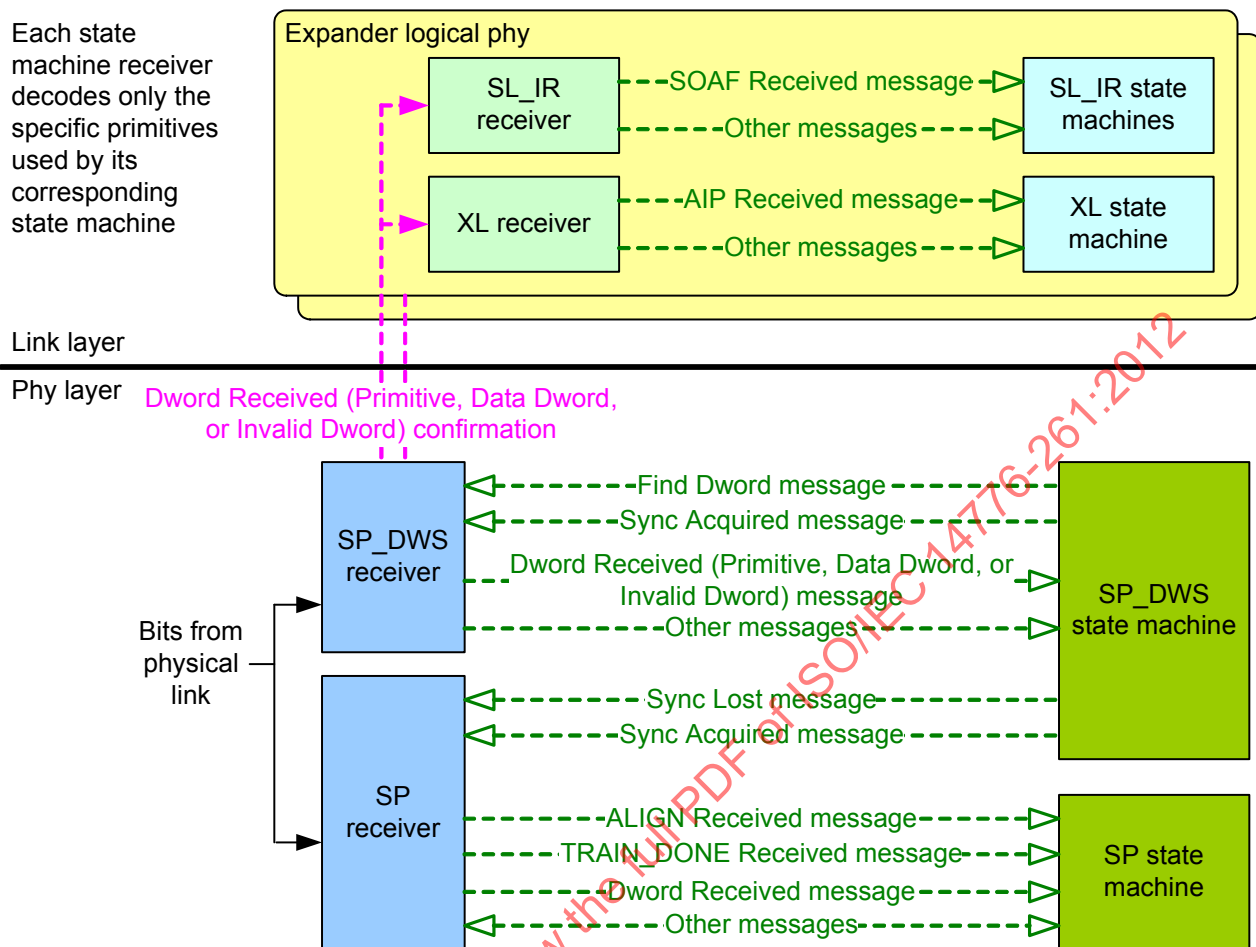


Figure 36 — Receive data path in an expander phy

4.3.4 State machines and SAS Device, SAS Port, and SAS Phy classes

Figure 37 shows which state machines are contained within the SAS Device, SAS Port, SAS Phy, and SAS Logical Phy classes.

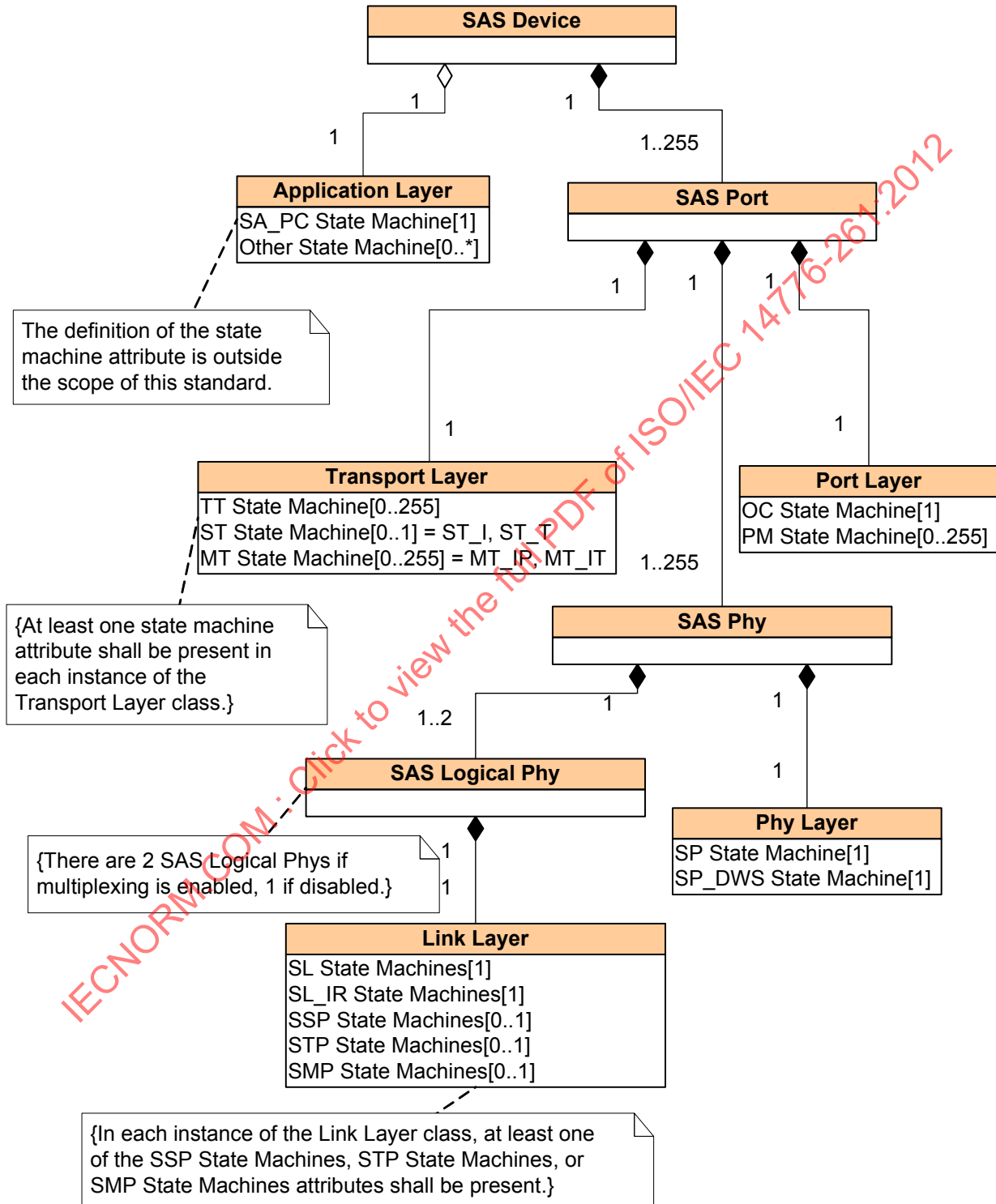


Figure 37 — State machines and SAS Device classes

Figure 38 shows which state machines are contained within the Expander Device, Expander Port, Expander Phy, Expander Logical Phy, SMP Port, SMP Phy, and SMP Logical Phy classes.

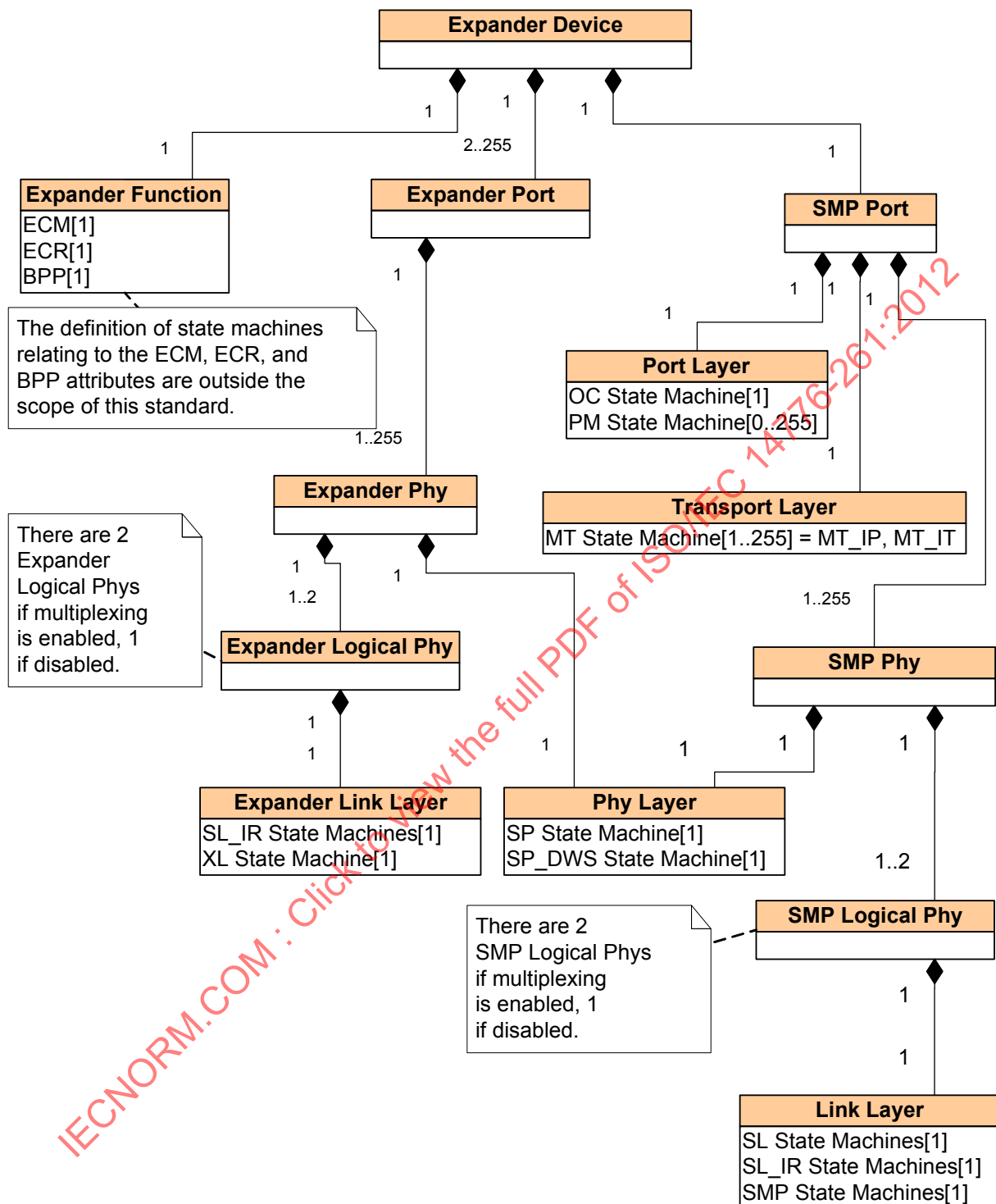


Figure 38 — State machines and Expander Device classes

4.4 Resets

4.4.1 Reset overview

Figure 39 shows the reset terminology used in this standard:

- a) link reset sequence (see 6.9);
- b) phy reset sequence (see 5.7);
- c) SATA OOB sequence (see 5.7.2.1);
- d) SATA speed negotiation sequence (see 5.7.2.2);
- e) SAS OOB sequence (see 5.7.4.1);
- f) SAS speed negotiation sequence (see 5.7.4.2);
- g) identification sequence (see 6.9); and
- h) hard reset sequence (see 6.9).

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

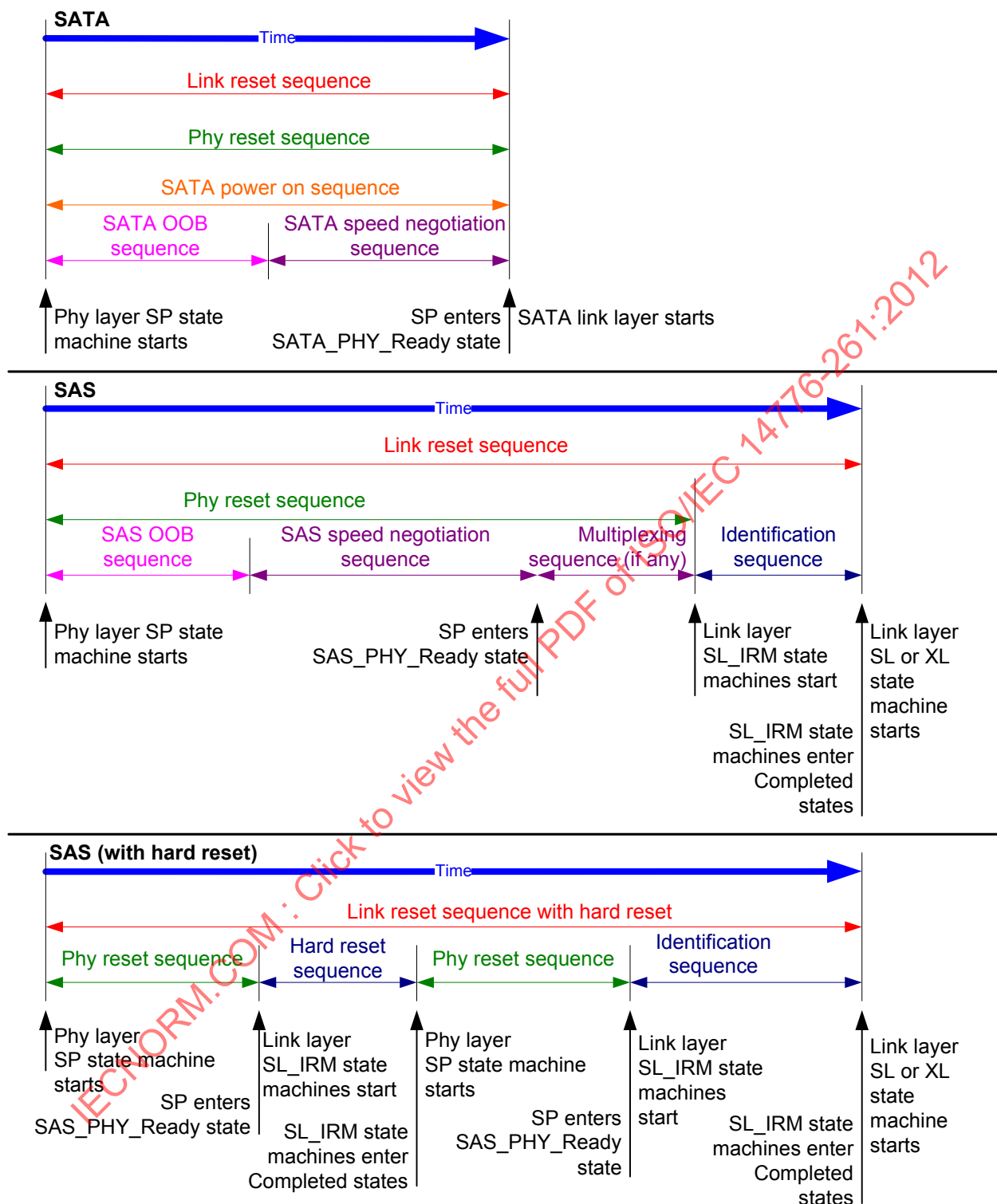


Figure 39 — Reset terminology

The phy reset sequences, including the OOB sequence, the speed negotiation sequence, and the multiplexing sequence, if any, are implemented by the SP state machine and the SP_DWS state machine and are described in 5.7, 5.9, and 5.10. The identification sequence and hard reset sequence are implemented by the SL_IR state machines and are described in 6.9 and 6.10.

The link reset sequence has no effect on the transport layer and application layer unless the link reset sequence disrupts frame transmission. A hard reset sequence replaces the identification sequence to initiate a hard reset. The link reset sequence serves as a hard reset for SATA devices (see SATA).

4.4.2 Hard reset

4.4.2.1 Hard reset overview

If, after the phy reset sequence, a phy receives a HARD_RESET primitive sequence before an IDENTIFY address frame, then the phy shall consider this to be a reset event, and the port containing the phy shall process a hard reset.

When a port processes a hard reset, the port shall stop transmitting valid dwords on each of the phys contained in that port. Each phy may then participate in new phy reset sequences (e.g., respond to incoming COMINITs) and shall originate a new link reset sequence if one is not detected. The hard reset shall not affect any other ports in the device.

If a SAS device is contained in an expander device, its SSP ports, STP ports, and/or SATA ports shall initiate a hard reset when an SMP PHY CONTROL function with a phy operation of HARD_RESET and phy identifier specifying a virtual expander phy attached to such a SAS port is processed (see 9.4.3.28).

4.4.2.2 Additional hard reset processing by SAS ports

If the port processing the hard reset is an SSP port, then the hard reset causes a Transport Reset event notification to be sent to the SCSI application layer (see 9.2.5), and the SCSI device shall perform the actions defined for hard reset in SAM-5. After processing the hard reset, each logical unit to which the SSP target port has access shall establish a unit attention condition for all SSP initiator ports with the additional sense code set to SCSI BUS RESET OCCURRED (see SAM-5 and SPC-4).

If the port processing the hard reset is an STP port in an STP/SATA bridge, then the SATA host port shall originate a link reset sequence.

If the port processing the hard reset is an STP port that is not in an STP/SATA bridge, then the STP target device shall perform the actions defined for power on or hardware reset in ATA8-AAM.

4.4.2.3 Additional hard reset processing by expander ports

If the port processing a hard reset is an expander port, then the expander device shall not originate a hard reset sequence on any of its other phys.

If the port processing a hard reset is an expander port, then the expander function and other expander ports in the expander device shall not be affected by hard reset. SAS devices contained in the expander device shall not be affected by hard resets received by external expander ports in the expander device.

4.5 I_T nexus loss

When a SAS port receives OPEN_REJECT (NO DESTINATION), OPEN_REJECT (PATHWAY BLOCKED), OPEN_REJECT (RESERVED INITIALIZE 0), OPEN_REJECT (RESERVED INITIALIZE 1), OPEN_REJECT (RESERVED STOP 0), OPEN_REJECT (RESERVED STOP 1), or an Open Timeout timer expires (see 6.13.2) in response to a connection request, the SAS port shall retry the connection request until:

- a) the connection is established; or
- b) the I_T nexus loss timer, if any, expires (see 7.2.2.1, 9.2.7.4, and 9.4.3.18).

An I_T nexus loss occurs in a SAS port when:

- a) the I_T nexus loss timer expires;
- b) the SAS port receives an abandon-class OPEN_REJECT (see table 74 in 6.2.6.10); or
- c) the SAS port receives a vendor specific indication (e.g., SGPIO presence detection) that the destination port is no longer in the SAS domain.

I_T nexus loss is handled by the port layer state machines (see 7.2.2.3). In some cases, the I_T nexus loss timer is overridden for connection requests through self-configuring expander devices as described in 4.7.1.

If an I_T nexus loss occurs in an SSP port, then the port shall send a Nexus Loss event notification to the SCSI application layer (see 9.2.5) and the SCSI device shall perform the actions defined for I_T nexus loss in SAM-5. If an I_T nexus loss occurs in an SSP initiator port, a SCSI application client should send an I_T NEXUS RESET task management function to the SSP target port during the next connection between that SSP initiator port and that SSP target port.

If an I_T nexus loss occurs in an STP initiator port, then the STP initiator port shall send a Transport Event Notification (Nexus Loss, [Device]) indication to the ATA application client (i.e., create a nexus loss event) (see ATA8-AAM). The ATA application client shall consider any commands for the lost STP target port to be completed with an error.

If an I_T nexus loss occurs in an STP target port, then the ATA device server shall abort all outstanding commands for the lost STP initiator port. If the STP target port is in an STP/SATA bridge, then the STP/SATA bridge shall originate a link reset sequence to the SATA device so the ATA device server in the SATA device aborts all outstanding commands.

If an I_T nexus loss occurs in an SMP initiator port, then the port shall stop attempting to establish connections to the lost SMP target port.

If an I_T nexus loss occurs in an initiator port due to I_T nexus loss timer expiration, then a management application client should cause a link reset sequence on the phy(s) attached to the lost target port (e.g, if directly attached, then the phys in the initiator port; if attached via expander device(s), then the phys in the expander device attached to the target port).

4.6 Expander device model

4.6.1 Expander device model overview

An expander device shall contain the following:

- a) an expander function containing:
 - A) an expander connection manager (ECM) (see 4.6.3);
 - B) an expander connection router (ECR) (see 4.6.4); and
 - C) a Broadcast propagation processor (BPP) (see 4.6.5);
- b) two or more physical expander phys;
- c) an expander port available per phy; and
- d) an SMP target port and a management device server.

An expander device may contain:

- a) an SMP initiator port and a management application client; and/or
- b) SAS devices with SSP ports, STP ports, and/or SMP ports and their associated device servers and/or application clients.

Figure 40 shows a model of an expander device showing the state machines in each expander port. The internal SMP ports are not shown.

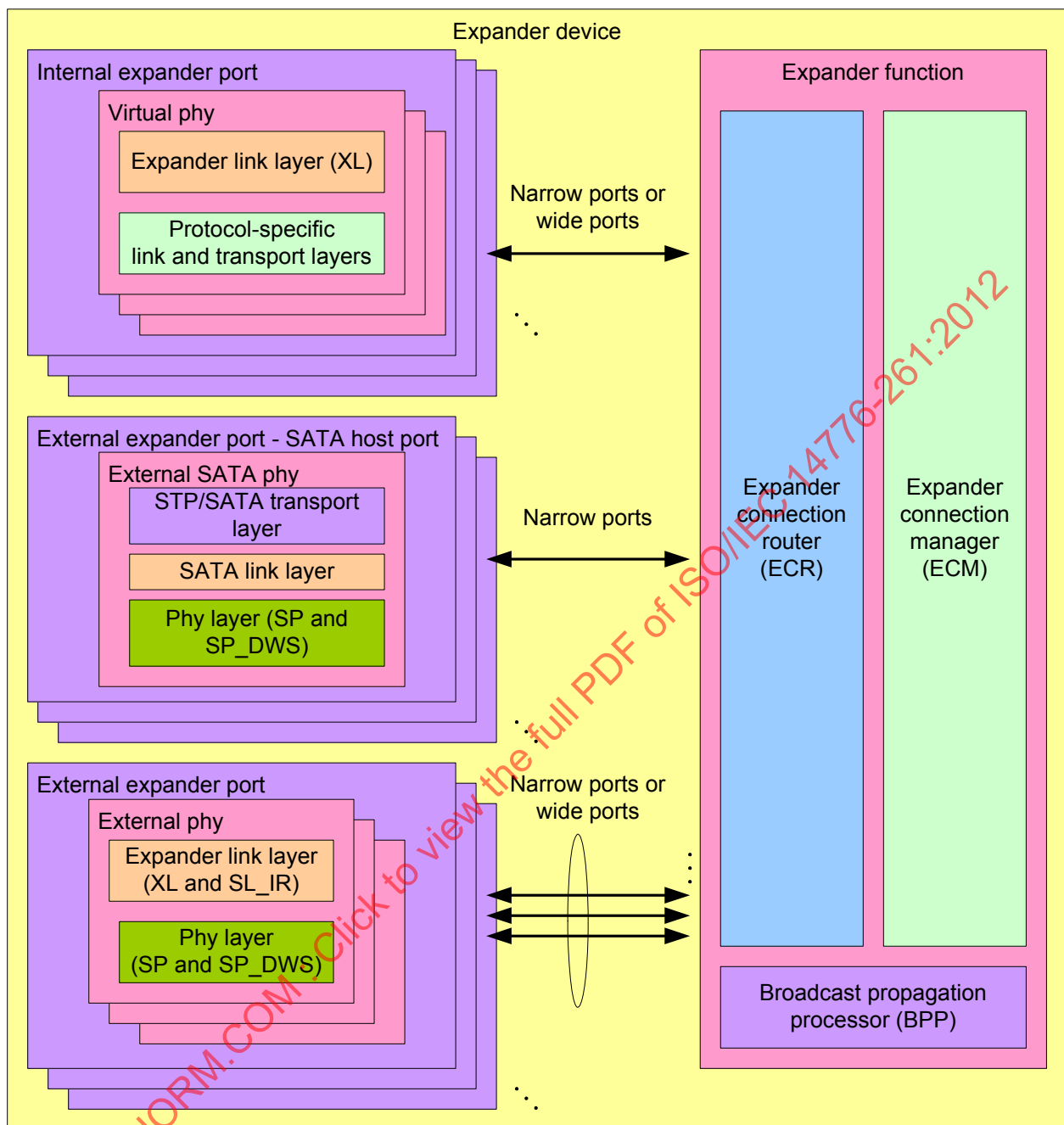


Figure 40 — Expander device model

4.6.2 Expander ports

An external expander port contains one or more physical phys (see 4.1.2). Since each phy in the expander device has the same SAS address, expander ports are created based on the attached SAS addresses (see 4.1.4).

Each phy in an expander port shall have the same routing attribute (see 4.6.7.1). The management device server shall return the same value in the ROUTING ATTRIBUTE field for each phy in an expander port in the SMP DISCOVER response (see 9.4.3.10).

Each phy in an expander port containing phys with table routing attributes in an externally configurable expander device shall have the same number of routing table entries (see 4.6.7.4).

A set of expander phys with table routing attributes in an expander device not supporting table-to-table attachments using the same external connector is called an enclosure out port (see SAS-2.1). A set of expander phys with subtractive routing attributes using the same external connector is called an enclosure in port (see SAS-2.1). A set of expander phys with table routing attributes in an expander device supporting table-to-table attachments using the same external connector is called an enclosure universal port (see SAS-2.1).

Each phy in an expander port shall have the same zone phy information (see 4.9.3.1). The zone phy information associated with each of the phys in an expander port is treated as the zoning properties of the expander port.

Each expander logical phy contains an expander link layer with an XL state machine (see 6.16) and one set of SL_IR state machines (see 6.10). The XL state machine in each expander logical phy within an expander port processes connection requests independently of the XL state machines in other expander logical phys.

An internal expander port contains a virtual phy with an expander link layer and a protocol-specific transport layer (e.g., to provide access as an SSP target port to a logical unit with a peripheral device type set to 0Dh (i.e., enclosure services device) (see SPC-4 and SES-3)).

Each expander device shall include one internal SMP port using the expander device's SAS address.

Any additional internal SAS ports shall be inside SAS devices contained in the expander device, and thus have SAS addresses different from that of the expander device. These SAS ports shall be attached to internal expander ports with virtual phys.

Each STP/SATA bridge shall have a unique SAS address. This SAS address is reported in the ATTACHED SAS ADDRESS field in the SMP DISCOVER response (see 9.4.3.10) for the expander phy containing the STP/SATA bridge (i.e., the expander phy attached to the SATA device or SATA port selector).

4.6.3 Expander connection manager (ECM)

The ECM performs the following functions:

- a) maps a destination SAS address in a connection request to a destination phy using direct, subtractive, or table routing methods;
- b) arbitrates and assigns or denies path resources for connection requests following SAS arbitration and pathway recovery rules;
- c) configures the ECR; and
- d) manages the phy power conditions (see 4.10).

4.6.4 Expander connection router (ECR)

The ECR routes messages between pairs of expander logical phys as configured by the ECM. Enough routing resources shall be provided to support at least one connection.

When forwarding dwords during a connection from a phy with a higher logical link rate to a phy with a lower logical link rate, rate matching (see 6.14) ensures the dwords are at a connection rate equal to or less than the lower logical link rate. However, the ECR may be requested to forward more dwords than the receiving phy is able to accept if:

- a) an invalid dword occurs during a deletable primitive;
- b) an invalid dword occurs during a CLOSE primitive sequence; or
- c) multiple invalid dwords occur during a BREAK primitive sequence.

The ECR may discard dwords if needed and count them as elasticity buffer overflows (see 4.13).

When forwarding dwords from a SATA physical link with a higher physical link rate to a SAS logical link with a lower logical link rate, the SATA host port in the STP/SATA bridge shall throttle incoming FISes with SATA_HOLD (see 6.18.2).

NOTE 10 The receiver may send up to 20 data dwords after detection of SATA_HOLD. Therefore, the transmitter of SATA_HOLD may receive up to 21+n data dwords (see 6.18.2). SATA_HOLD does not affect primi-

tives (see SATA). The STP/SATA bridge may expand or contract repeated and continued primitives without changing their functional meaning.

When forwarding dwords from a SAS logical link with a lower logical link rate to a SATA physical link with a higher physical link rate, the SATA host port in the STP/SATA bridge shall perform a process similar to rate matching (see 6.14) by inserting ALIGN (0) and/or SATA_HOLD on the SATA physical link whenever it underflows.

NOTE 11 SATA requires that ALIGN (0) be transmitted in pairs (see SATA).

4.6.5 Broadcast propagation processor (BPP)

The BPP receives Broadcasts from each expander logical phy or from the management device server on behalf of an expander logical phy and requests transmission of those Broadcasts on all expander ports except the expander port from which the Broadcast was received.

In a zoning expander device with zoning enabled (see 4.9.2), Broadcasts are forwarded as described in 4.9.5.

4.6.6 Expander device interfaces

4.6.6.1 Expander device interface overview

The expander device arbitrates and routes between expander logical phys. All routing occurs between expander logical phys, not expander ports. The interaction between an XL state machine and the expander function consists of requests, confirmations, indications, and responses. This interaction is called the expander device interface.

Figure 41 describes the interfaces present within an expander device.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

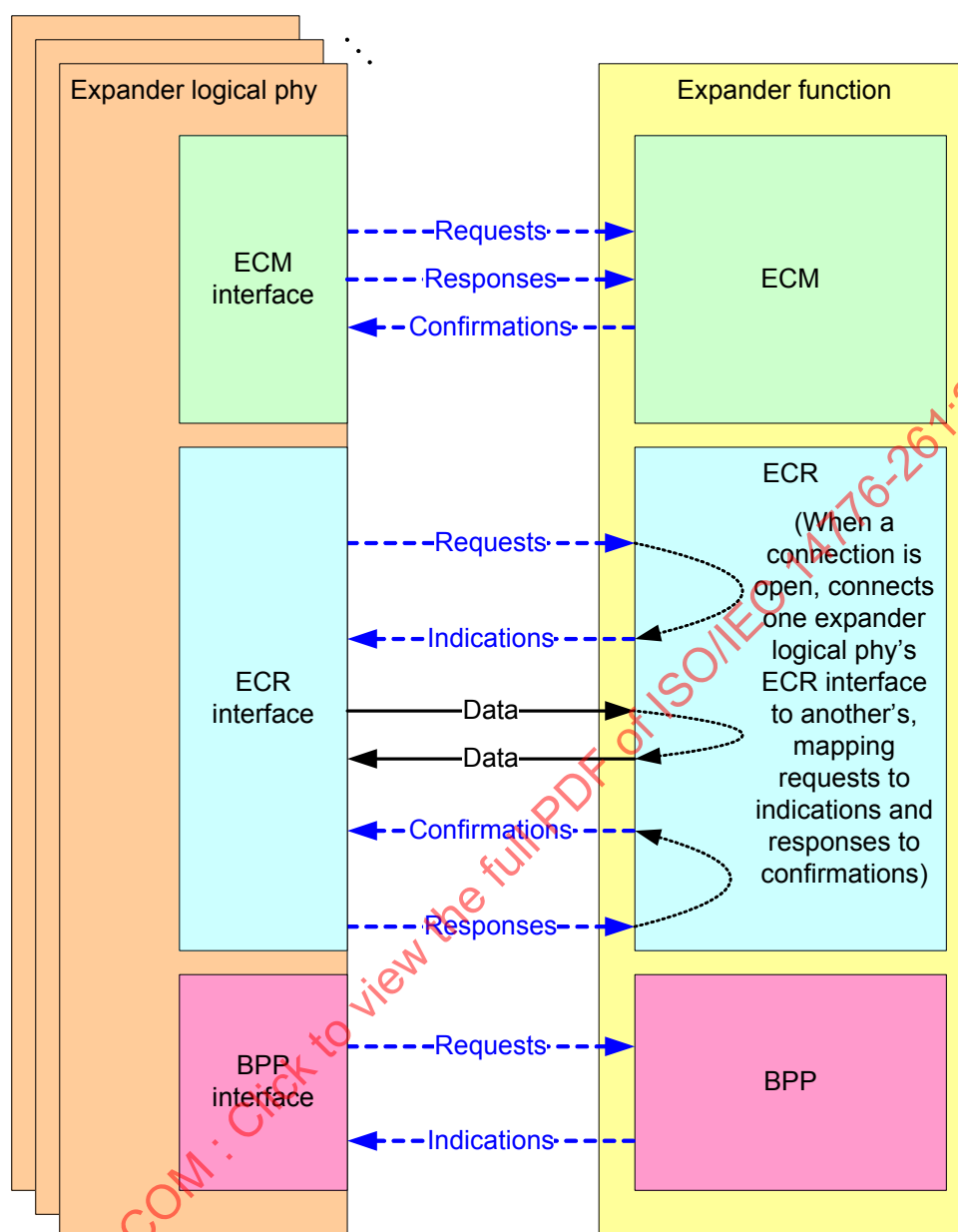


Figure 41 — Expander device interfaces

4.6.6.2 Expander device interfaces detail

Figure 42 shows the interface requests, confirmations, indications, and responses used by an expander device to manage connections.

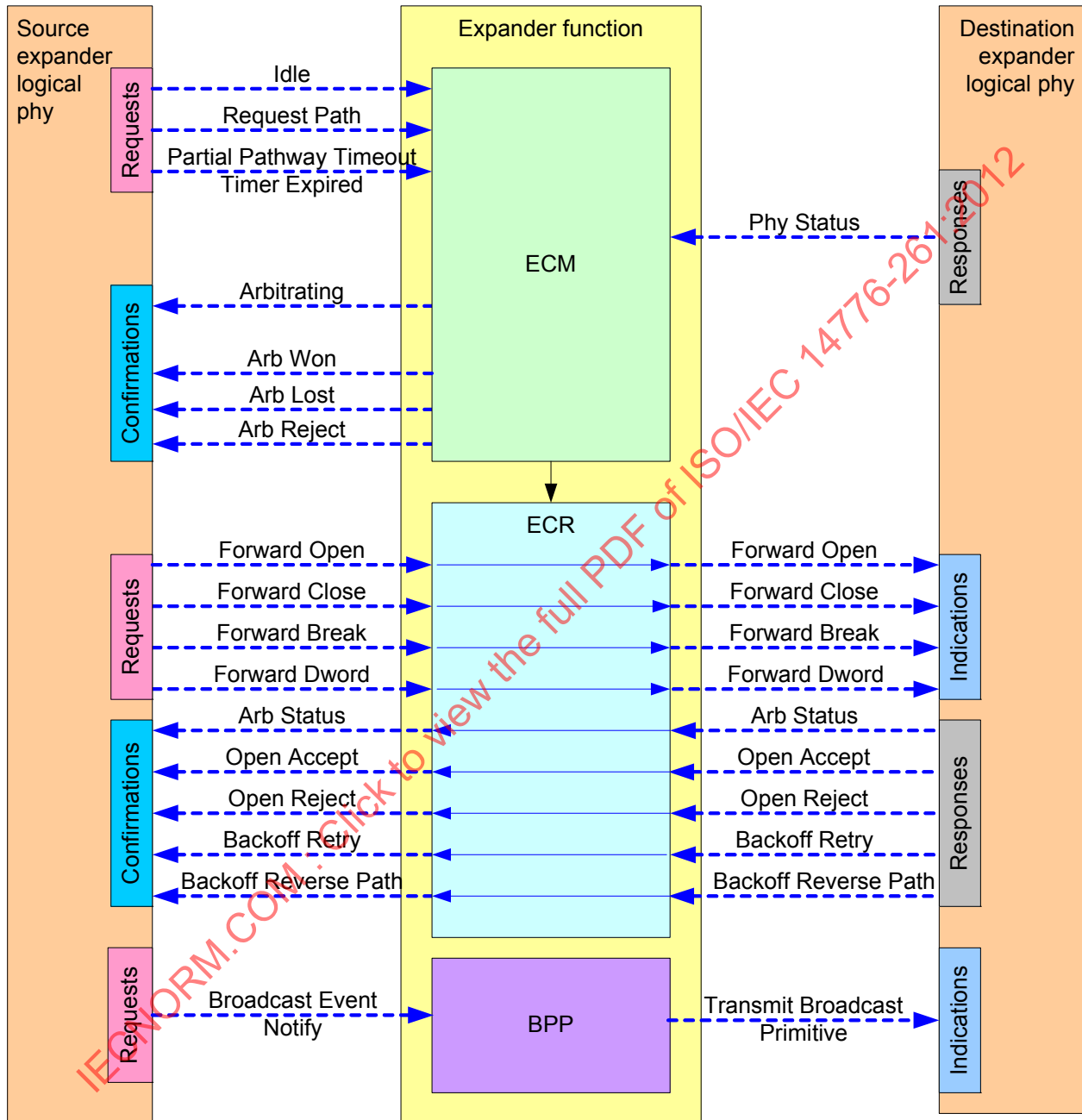


Figure 42 — Expander device interface detail

4.6.6.3 ECM interface

Table 14 describes the requests from an expander logical phy to the ECM. The XL state machine (see 6.16) defines when each request is sent.

Table 14 — Expander logical phy to ECM requests

Request	Description
Idle	The XL state machine is in the XL0:Idle state (e.g., after receiving an Enable Disable SAS Link (Disable) message).
Request Path (arguments)	Request for a connection.
Partial Pathway Timeout Timer Expired	The Partial Pathway Timeout Timer expired.

Table 15 describes the responses from an expander logical phy to the ECM. The XL state machine (see 6.16) defines when each response is sent.

Table 15 — Expander logical phy to ECM responses

Response	Description
Phy Status (Partial Pathway)	Response meaning that an expander logical phy: a) is being used for an unblocked partial pathway; or b) is waiting on another expander logical phy being used for a partial pathway.
Phy Status (Blocked Partial Pathway)	Response meaning that an expander logical phy: a) is being used for a blocked partial pathway; or b) is waiting on another expander logical phy being used for a blocked partial pathway.
Phy Status (Connection)	Response meaning that an expander logical phy: a) is being used for a connection; or b) is waiting on another expander logical phy being used for a connection.
Phy Status (Breaking Connection)	Response meaning that an expander logical phy is breaking a connection.

Table 16 describes the confirmations from the ECM to an expander logical phy. These confirmations are sent in confirmation of a Request Path request. See 6.13.4 for specific definitions for when each confirmation is sent.

Table 16 — ECM to expander logical phy confirmations

Confirmation	Description
Arbitrating (Normal)	Confirmation that the ECM has received the Request Path request.
Arbitrating (Waiting On Partial)	Confirmation that the ECM is waiting on a partial pathway (see 4.1.11).
Arbitrating (Blocked On Partial)	Confirmation that the ECM is waiting on a blocked partial pathway (see 4.1.11).
Arbitrating (Waiting On Connection)	Confirmation that the ECM is waiting for a connection to complete (see 4.1.12).
Arb Won	Confirmation that an expander logical phy has won path arbitration.
Arb Lost	Confirmation that an expander logical phy has lost path arbitration.
Arb Reject (No Destination)	Confirmation that the request is rejected because the expander device is not configuring (see 4.7.4) and there is no path to the destination.
Arb Reject (Bad Destination)	Confirmation that the request is rejected because the path to the destination maps back to the requesting expander port.
Arb Reject (Connection Rate Not Supported)	Confirmation that the request is rejected because there is a destination port capable of routing to the requested destination SAS address but no phys within the destination port are configured to support the requested connection rate.
Arb Reject (Zone Violation)	Confirmation that the request is rejected because the expander device is not locked and there is a zoning violation (see 4.9.3).
Arb Reject (Pathway Blocked)	Confirmation that the request is rejected because the requesting expander logical phy needs to back off according to SAS pathway recovery rules.
Arb Reject (Retry)	Confirmation that the request is rejected because: a) the expander device is configuring (see 4.7.4) and the ECM would otherwise have returned Arb Reject (No Destination); b) the expander device is locked (see 4.9.6.2) and the ECM would otherwise have returned Arb Reject (Zone Violation); - or c) the expander device has reduced functionality (see 4.6.8 and 6.13.4.2.5).

4.6.6.4 ECR interface

Table 17 describes the requests from an expander logical phy to the ECR and the corresponding indications from the ECR to another expander logical phy. The XL state machine (see 6.16) defines when each request is sent.

Table 17 — Expander logical phy to ECR to expander logical phy requests and indications

Request/indication	Description
Forward Open (arguments)	Request/indication to forward an OPEN address frame.
Forward Close	Request/indication to forward a CLOSE.
Forward Break	Request/indication to forward a BREAK.
Forward Dword	Request/indication to forward a dword.

Table 18 describes the responses from an expander logical phy to the ECR and the corresponding confirmations from the ECR to another expander logical phy. These responses are sent in response to a Forward Open indication. The XL state machine (see 6.16) defines when each response is sent.

Table 18 — Expander logical phy to ECR to expander logical phy responses and confirmations

Response/confirmation	Description
Arb Status (Normal)	Confirmation/response that AIP (NORMAL) has been received.
Arb Status (Waiting On Partial)	Confirmation/response that AIP (WAITING ON PARTIAL) has been received.
Arb Status (Waiting On Connection)	Confirmation/response that AIP (WAITING ON CONNECTION) has been received.
Arb Status (Waiting On Device)	Confirmation/response that either: a) AIP (WAITING ON DEVICE) has been received; or b) the expander logical phy has completed the forwarding of an OPEN address frame and has entered the XL6:Open_Response_Wait state.
Open Accept	Confirmation/response that OPEN_ACCEPT has been received.

Table 18 — Expander logical phy to ECR to expander logical phy responses and confirmations

Response/confirmation	Description
Open Reject	Confirmation/response that either: a) OPEN_REJECT has been received; or b) that the phy is in the slumber phy power condition (see 4.10.1.4).
Backoff Retry	Confirmation/response that the phy is in the partial phy power condition (see 4.10.1.3) or: a) a higher priority OPEN address frame has been received (see 6.13.3); and b) the source SAS address and connection rate of the received OPEN address frame are not equal to the destination SAS address and connection rate of the transmitted OPEN address frame.
Backoff Reverse Path	Confirmation/response that: a) a higher priority OPEN address frame has been received (see 6.13.3); and b) the source SAS address and connection rate of the received OPEN address frame are equal to the destination SAS address and connection rate of the transmitted OPEN address frame.

4.6.6.5 BPP interface

Table 19 describes the requests from an expander logical phy to the BPP. Requests from the management device server about SMP ZONED BROADCAST requests (see table 267) received from the SMP target port in zoning expander devices with zoning enabled are not described by this standard. See 4.9.5 for more information on how zoning expander devices with zoning enabled handle Broadcasts.

Table 19 — Expander logical phy to BPP requests

Request	Description
Broadcast Event Notify (Phy Not Ready)	Request to originate a Broadcast (Change) because the expander logical phy's SP state machine transitioned from the SP15:SAS_PHY_Ready state, SP22:SATA_PHY_Ready state, SP31:SAS_PS_Low_Phy_Power state, SP32:SAS_PS_ALIGN0 state, or SP33:SAS_PS_ALIGN1 state to the SP0:OOB_COMINIT state (see 5.9).
Broadcast Event Notify (SATA Spinup Hold)	Request to originate a Broadcast (Change) because the SATA spinup hold state has been reached (see 5.9 and 5.12) by the expander phy.
Broadcast Event Notify (Identification Sequence Complete)	Request to originate a Broadcast (Change) because the expander logical phy has completed the identification sequence (see 6.9).
Broadcast Event Notify (SATA Port Selector Change)	Request to originate a Broadcast (Change) because the expander phy detected that a SATA port selector appeared or disappeared.
Broadcast Event Notify (Change Received)	Request to forward a Broadcast (Change) because the expander logical phy received a Broadcast (Change). See 6.12 and 6.16.

Table 19 — Expander logical phy to BPP requests

Request	Description
Broadcast Event Notify (Reserved Change 0 Received)	Request to forward a Broadcast (Reserved Change 0) because the expander logical phy received a Broadcast (Reserved Change 0). See 6.12 and 6.16.
Broadcast Event Notify (Reserved Change 1 Received)	Request to forward a Broadcast (Reserved Change 1) because the expander logical phy received a Broadcast (Reserved Change 1). See 6.12 and 6.16.
Broadcast Event Notify (SES Received)	Request to forward a Broadcast (SES) because the expander logical phy received a Broadcast (SES). See 6.16.
Broadcast Event Notify (Expander Received)	Request to forward a Broadcast (Expander) because the expander logical phy received a Broadcast (Expander). See 6.16.
Broadcast Event Notify (Asynchronous Event Received)	Request to forward a Broadcast (Asynchronous Event) because the expander logical phy received a Broadcast (Asynchronous Event). See 6.16.
Broadcast Event Notify (Reserved 3 Received)	Request to forward a Broadcast (Reserved 3) because the expander logical phy received a Broadcast (Reserved 3). See 6.16.
Broadcast Event Notify (Reserved 4 Received)	Request to forward a Broadcast (Reserved 4) because the expander logical phy received a Broadcast (Reserved 4). See 6.16.

Table 20 describes the indications from the BPP to an expander logical phy. Indications to the management application client to generate SMP ZONED BROADCAST functions from the SMP initiator port in a zoning expander device with zoning enabled are not described. See 4.9.5 for more information on how zoning expander devices with zoning enabled handle Broadcasts.

Table 20 — BPP to expander logical phy indications

Message	Description
Transmit Broadcast (type)	Indication to transmit a BROADCAST primitive sequence with the specified type.

4.6.7 Expander device routing

4.6.7.1 Routing attributes and routing methods

Each expander phy in an expander device shall support one of the following routing attributes:

- a direct routing attribute;
- a table routing attribute; or
- a subtractive routing attribute.

The routing attributes allow the ECM to determine which routing method to use when routing connection requests to the expander logical phys in the expander phy:

- the table routing method routes connection requests to attached expander devices using an expander route table;
- the subtractive routing method routes unresolved connection requests to an attached expander device; or
- the direct routing method routes connection requests to attached end devices, the SMP port of an attached expander device, or SAS devices contained in the expander device.

Table 21 describes the routing methods that the ECM uses based on the routing attributes of an expander phy.

Table 21 — Routing attributes and routing methods

Routing attribute of an expander phy	Routing method used by ECM for the expander phy	
	If attached to an end device	If attached to an expander device
Direct	Direct ^a	
Table	Direct	Direct for the SAS address of the expander device. Table for SAS addresses beyond the expander device.
Subtractive	Direct	Subtractive
^a If attached to an expander device, then the ECM is only able to route to the expander device itself through a phy with the direct routing attribute.		

An expander device may have zero or more phys with the table routing attribute. A self-configuring expander device may support table-to-table attachment (i.e., having its table routing phys attached to the table routing phys of other expander devices). An externally configurable expander device shall not support table-to-table attachment.

An expander device shall have at most one defined port containing phys with the subtractive routing attribute.

The SMP REPORT GENERAL function (see 9.4.3.4) reports whether or not the expander device is self-configuring and supports table-to-table attachment. The SMP DISCOVER function (see 9.4.3.10) reports the routing attribute of each expander phy (see 9.4.3.4).

4.6.7.2 Expander device topology routing attribute restrictions

If an expander device does not support table-to-table attachment, then its table-routing phys shall not be attached to table routing phys in other expander devices (e.g., they may be attached to subtractive routing phys).

If multiple phys within an expander device have subtractive routing attributes and are attached to expander devices, then they shall be attached to phys with identical SAS addresses (i.e., the same expander device).

If multiple phys within an expander device have subtractive routing attributes and are attached to expander devices that do not have identical SAS addresses, then the application client that is performing the discover process (see 4.7) shall report an error in a vendor specific manner.

4.6.7.3 Connection request routing

The ECM shall determine how to route a connection request from a source expander logical phy to a destination expander logical phy in a different expander port if the destination expander logical phy is enabled, operating at a valid logical link rate (e.g., the SMP DISCOVER response reports a NEGOTIATED LOGICAL LINK RATE field set to G1 (i.e., 8h), G2 (i.e., 9h), or G3 (i.e., Ah)), and not excluded because of zoning (see 4.9.2) using the following precedence:

- 1) route to an expander logical phy with the direct routing attribute or table routing attribute when the destination SAS address matches the attached SAS address;
- 2) route to an expander logical phy with the table routing attribute when the destination SAS address matches an enabled SAS address in the expander route table;
- 3) route to an expander logical phy with the subtractive routing attribute; and
- 4) return an Arb Reject confirmation (see 4.6.6.3) to the source expander logical phy.

If the destination expander logical phy only matches an expander logical phy in the same expander port from which the connection request originated, then the ECM shall return an Arb Reject confirmation (see 4.6.6.3).

If the destination SAS address of a connection request matches a disabled SAS address in an expander route table, then the ECM shall ignore the match.

If low phy power conditions (see 4.10.1) are enabled in the SAS expander device, then the ECM should use the following precedence in selecting the destination expander logical phy when the expander port contains multiple expander logical phys:

- 1) expander logical phys in the active phy power condition (see 4.10.2);
- 2) expander logical phys in the partial phy power condition (see 4.10.1.3); and
- 3) expander logical phys in the slumber phy power condition (see 4.10.1.4).

4.6.7.4 Expander route table

4.6.7.4.1 Expander route table overview

An expander device that supports the table routing method shall contain an expander route table. The expander route table is a structure that provides an association between destination SAS addresses (i.e., routed SAS addresses) and the expander phys to which connection requests to those destination SAS addresses are forwarded.

Zoning expander devices include additional fields in their expander route tables (see 4.9.3.4).

Table 22 defines the types of expander route tables.

Table 22 — Expander route table types

Type	SMP functions to access	Reference
Phy-based	REPORT ROUTE INFORMATION (see 9.4.3.13) and CONFIGURE ROUTE INFORMATION (see 9.4.3.27)	4.6.7.4.2
Expander-based	REPORT EXPANDER ROUTE TABLE LIST (see 9.4.3.17)	4.6.7.4.3

4.6.7.4.2 Phy-based expander route table

Figure 43 shows a representation of a phy-based expander route table.

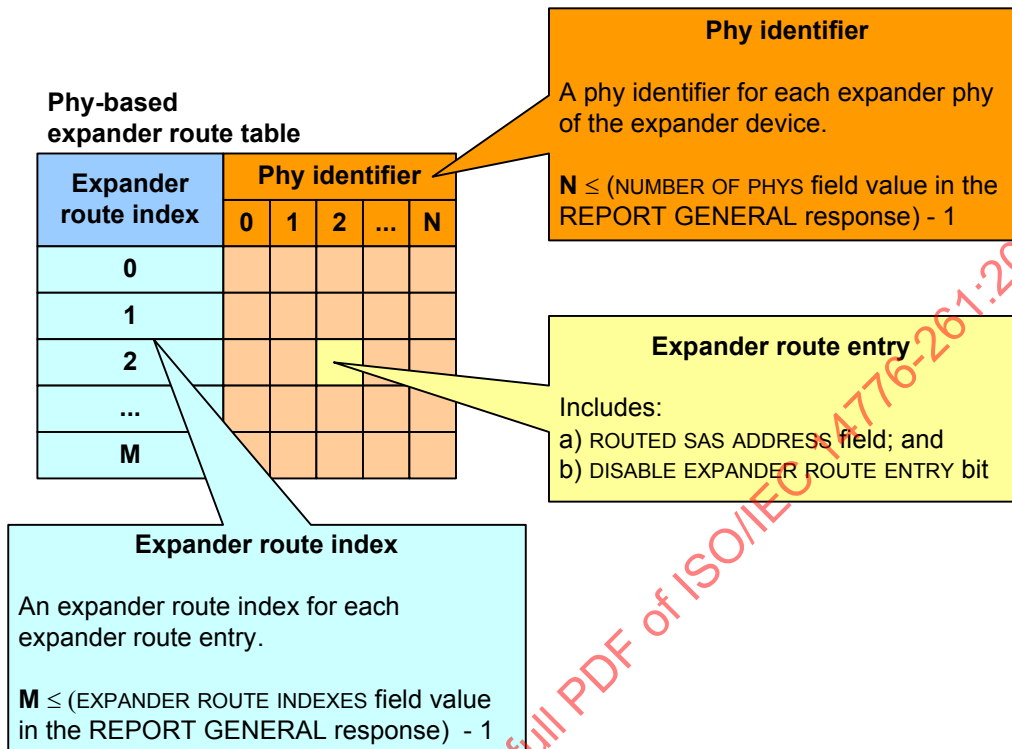


Figure 43 — Phy-based expander route table

For each expander route index and phy identifier combination, the phy-based expander route table contains an expander route entry containing a ROUTED SAS ADDRESS field and a DISABLE EXPANDER ROUTE ENTRY bit.

A management application client may access a specific expander route entry within a phy-based expander route table with the SMP REPORT ROUTE INFORMATION function (see 9.4.3.13) and the SMP CONFIGURE ROUTE INFORMATION function (see 9.4.3.27).

An expander device reports the maximum expander route index in the EXPANDER ROUTE INDEXES field and indicates if the phy-based expander route table is configurable in the EXTERNALLY CONFIGURABLE ROUTE TABLE bit in the SMP REPORT GENERAL response (see 9.4.3.4).

Each expander route entry shall be disabled after power on.

4.6.7.4.3 Expander-based expander route table

Figure 44 shows a representation of an expander-based expander route table.

Expander-based expander route table

Routed SAS address	Phy bit map				
	0	1	2	...	N
SAS address A					
SAS address B					
SAS address C					
...					

Indicates the expander phy(s) to which connection requests to the routed SAS address may be forwarded.

1 indicates the expander phy is used for forwarding connection requests, 0 indicates it is not

Figure 44 — Expander-based expander route table

Routed SAS addresses are not required to be sorted in any particular order.

For each routed SAS address, the expander-based expander route table contains a phy bit map.

A management application client may access an expander-based expander route table with the SMP REPORT EXPANDER ROUTE TABLE LIST function (see 9.4.3.17).

An expander device reports the size of its expander-based expander route table in the MAXIMUM NUMBER OF ROUTED SAS ADDRESSES field in the SMP REPORT GENERAL response (see 9.4.3.4).

4.6.8 Expander device reduced functionality

An expander device shall originate a Broadcast (Expander) to indicate that it is going to have reduced functionality for a period of time (e.g., if, during a microcode update, the expander device disables the ECM and ECR access to its SMP target port or to one or more expander phys, or if it experiences reduced performance). The maximum period of time that the expander device is going to have reduced functionality is indicated:

- in the REPORT SUPPORTED OPERATION CODES command parameter data (see SPC-4) for the WRITE BUFFER command reported by a logical unit with a peripheral device type set to 0Dh (i.e., enclosure services device) accessed via an SSP target port contained in the expander device; and
- from the contents of the MAXIMUM REDUCED FUNCTIONALITY TIME field in the REPORT GENERAL response (see 9.4.3.4).

After the expander device originates a Broadcast (Expander) to indicate that it is going to have reduced functionality for a period of time, it shall:

- set the REDUCED FUNCTIONALITY bit to one in the REPORT GENERAL response (see 9.4.3.4);
- initialize the reduced functionality delay timer to the value indicated by the INITIAL REDUCED FUNCTIONALITY DELAY field in the REPORT GENERAL response (see 9.4.3.4) and start the reduced functionality delay timer;
- wait for the reduced functionality delay timer to expire before reducing any expander functionality; and
- not stop or restart the reduced functionality delay timer until after the expander device enters the reduced functionality condition.

If the expander device receives a connection request that maps to an expander phy or its SMP target port that is not accessible because of the reduced functionality, then the expander device shall respond with an OPEN

REJECT (RETRY) until the operation that caused the expander device to have reduced functionality is complete.

After the operation that caused the expander device to have reduced functionality is complete, the expander device shall:

- 1) set the REDUCED FUNCTIONALITY bit to zero in the REPORT GENERAL response (see 9.4.3.4); and
- 2) originate a Broadcast (Change) or a link reset sequence on each expander phy.

4.6.9 Broadcast (Expander) handling

After receiving a Broadcast (Expander), a management application client behind an SMP initiator port should issue a REPORT GENERAL function (see 9.4.3.4) to all expander devices to determine:

- a) the expander devices, if any, that are reducing their functionality (i.e., the REDUCED FUNCTIONALITY bit is set to one in the REPORT GENERAL response) (see 4.6.8); and
- b) the amount of time remaining until the reduced functionality occurs (i.e., the contents of the TIME TO REDUCED FUNCTIONALITY field in the REPORT GENERAL response).

If an expander device is found that is reducing its functionality, then the management application client should:

- a) terminate any outstanding I_T_L_Q nexuses whose connections are through that expander device; and
- b) not create any new I_T_L_Q nexuses whose connections go through that expander device.

4.7 Discover process

4.7.1 Discover process overview

Management application clients direct an SMP initiator port to request SMP functions from an SMP target port. Management application clients are located in every SAS initiator device and every self-configuring expander device. A management application client performs a discover process to discover all the SAS devices and expander devices in the SAS domain (i.e., determining their device types, SAS addresses, and supported protocols). A SAS initiator device uses this information to determine SAS addresses with which it is able to establish connections and to select connection rates for connection requests (see 6.8.3). A self-configuring expander device uses this information to fill in its expander route table.

4.7.2 Starting the discover process (Broadcast (Change) handling)

In a SAS initiator device, a management application client behind an SMP initiator port should perform a discover process after a link reset sequence or after receiving a Broadcast (Change).

In a self-configuring expander device, the management application client behind an SMP initiator port in a self-configuring expander device shall perform a discover process after a link reset sequence or after receiving a Broadcast (Change).

When a discover process is performed after a link reset sequence, the management application client discovers all the devices in the SAS domain. When a discover process is performed after a Broadcast (Change), the management application client determines which devices have been added to or removed from the SAS domain.

4.7.3 Discover process traversal

A management application client performing the discover process shall perform a level-order (i.e., breadth-first) traversal of the SAS domain. The management application client shall discover devices in the following order:

- 1) the device(s) to which the device containing the management application client is attached;
- 2) for each expander phy with the subtractive routing attribute or the table routing attribute, if the attached device is an expander device, then every device attached to that expander device; and

- 3) repeat step 2) for each additional expander device found attached to that expander device.

The discover process completes when all expander devices have been traversed. If the management application client discovers an externally configurable expander device that is not located beyond a self-configuring expander device with the CONFIGURES OTHERS bit set to one in the REPORT GENERAL response (see 9.4.3.4), then the management application client shall perform the configuration subprocess (see 4.8) to configure the expander route table before attempting to establish connections with devices attached two or more levels (see 4.8.4) beyond that externally configurable expander device.

If a management application client is in an end device or a self-configuring expander device that is directly attached to a self-configuring expander device with the CONFIGURES OTHERS bit set to one in the REPORT GENERAL response (see 9.4.3.4), then it is not required to perform the configuration subprocess. If all the expander devices in the SAS domain are self-configuring expander devices, then management application clients in end devices are not required to perform the configuration subprocess.

If the management application client is inside a self-configuring expander device, then the discover process shall be repeated on each expander port.

Figure 45 shows an example of level-order traversal.

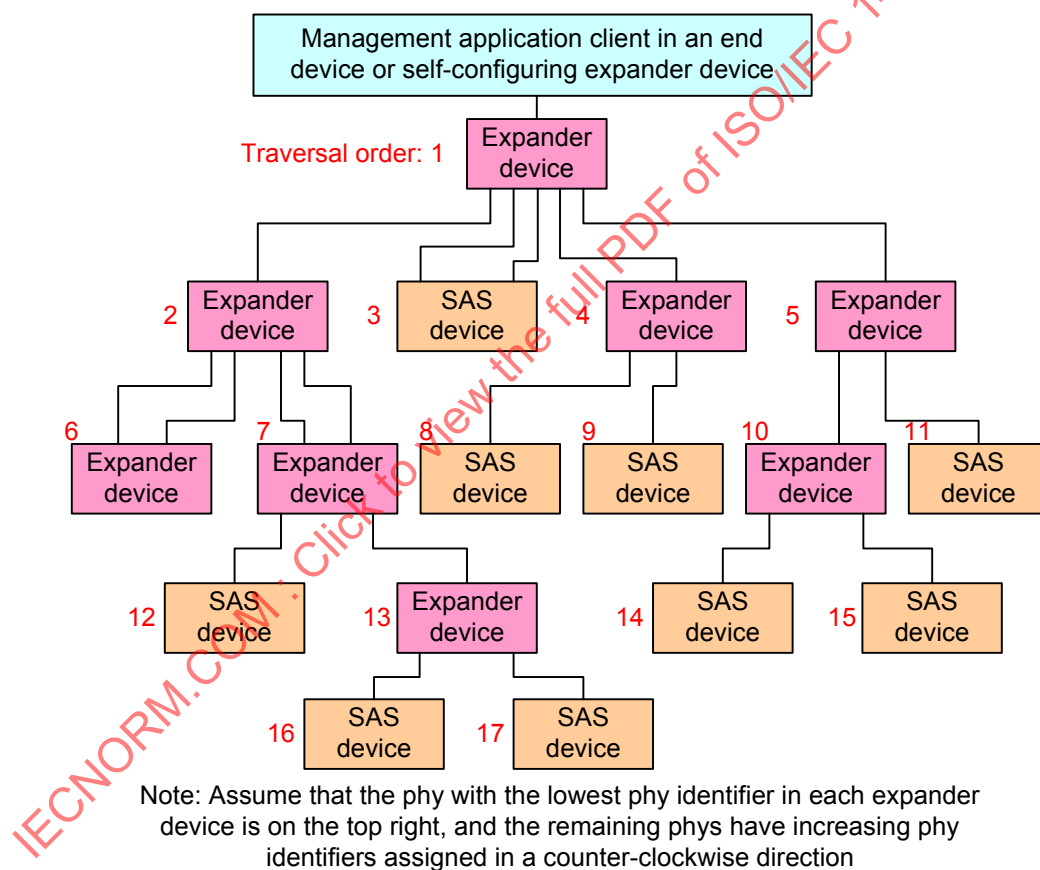


Figure 45 — Level-order traversal example

The management application client determines whether an expander device or SAS device is attached at each point in the traversal. For the first device (i.e., the device that is directly attached), this is determined from the DEVICE TYPE field in the IDENTIFY address frame (see 6.8.2) information received by the phy that the management application client is using. For other devices (i.e., devices that are not directly attached), this is determined from ATTACHED DEVICE TYPE field the SMP DISCOVER response (see 9.4.3.10).

If an expander device is attached, then the management application client shall use the SMP REPORT GENERAL function (see 9.4.3.4) to determine how many phys are in the expander device and then use the SMP DISCOVER function (see 9.4.3.10) and/or the SMP DISCOVER LIST function (see 9.4.3.15) to

determine what is attached to each expander phy (e.g., the device type, SAS address, and supported protocol(s)).

NOTE 12 Expander devices compliant with SAS-1.1 do not implement the SMP DISCOVER LIST function.

If the expander device's EXTERNALLY CONFIGURABLE ROUTE TABLE bit is set to zero in the SMP REPORT GENERAL response, its own management application client shall configure its own expander route table as described in 4.8.

While a self-configuring expander device's CONFIGURING bit is set to one in the SMP REPORT GENERAL response, connection requests for destination ports two or more levels beyond the self-configuring expander device that would otherwise have returned OPEN_REJECT (NO DESTINATION) return OPEN_REJECT (RETRY) instead (see 4.6.6.3, 4.7.4, and 4.9.6.3).

If a SAS device is attached, then the discover process is not required to obtain any more information about the SAS device. Additional discovery software may access that SAS device, however, as follows:

- a) if the SAS device supports an SMP target port, then the management application client may use SMP functions (e.g., REPORT GENERAL and REPORT MANUFACTURER INFORMATION) to determine additional information about the SAS target device;
- b) if the SAS device supports an SSP target port, then a SCSI application client may transmit SCSI commands (e.g., INQUIRY and REPORT LUNS) to determine additional information about the SCSI target device; and
- c) if the end device supports an STP target port, then an ATA application client may transmit ATA commands (e.g., IDENTIFY DEVICE and IDENTIFY PACKET DEVICE (see ACS-2)) to determine additional information about the ATA device.

The result of the discover process is that the management application client has the necessary information (e.g., the device type, SAS address, and supported protocol(s)) to communicate with each SAS device and expander device in the SAS domain and each externally configurable expander device is configured with the necessary expander route entries to allow routing of connection requests through the SAS domain.

The discover process may be aborted prior to completion and restarted if there is an indication that it may be using incorrect information (e.g., reception of a Broadcast (Change) or a change in the EXPANDER CHANGE COUNT field returned in an SMP response).

Annex I contains an example implementation of how a management application client may perform the discover process.

4.7.4 Discover process in a self-configuring expander device

The management application client of a self-configuring expander device shall configure:

- a) the expander routing table in that expander device; and
- b) the expander routing table in each externally configurable expander device in the SAS domain that is not located behind another self-configuring expander device with the CONFIGURES OTHERS bit set to one in the REPORT GENERAL response (see 9.4.3.4).

When a self-configuring expander device receives a Broadcast (Change) the management application client shall start the discover process using the expander port that received the Broadcast (Change). If a change to the expander route table is identified, then the management device server shall set its SELF CONFIGURING bit to one in the SMP REPORT GENERAL response (see 9.4.3.4).

If zoning is enabled, then the management application client in a self-configuring expander device shall use the SMP DISCOVER response (see 9.4.3.10) or SMP DISCOVER LIST response (see 9.4.3.15) values to set the zone group values in the zoning expander route table for all SAS addresses in the zoning expander route table (see 4.9.3.4).

The management device server shall set the SELF CONFIGURING bit to zero when the discover process is complete. When the SELF CONFIGURING bit changes from one to zero:

- a) a zoning expander device with zoning enabled shall originate a Broadcast (Change) on each expander port that has access to the expander port through which the discover process was performed based on the zone permission table; and
- b) an expander device with zoning disabled shall originate a Broadcast (Change) on each expander port other than the one through which the discover process was performed.

After receiving a Broadcast (Change), a self-configuring expander device shall continue to route connection requests for each previously valid SAS address until it determines that it is no longer valid. After determining that a SAS address is no longer valid, the self-configuring expander device shall continue to route connection requests for other SAS addresses.

While the SELF CONFIGURING bit is set to one, the expander device shall return OPEN_REJECT (RETRY) for any connection requests that would otherwise have resulted in OPEN_REJECT (NO DESTINATION) (see 4.6.6.3).

The management application client in a self-configuring expander device shall maintain self-configuration status for the last vendor specific number of errors encountered during self-configuration and should maintain at least one self-configuration status per phy. The management device server shall assign descriptors to the statuses sequentially starting at 0001h and shall return the descriptors in the SMP REPORT SELF-CONFIGURATION STATUS response (see 9.4.3.6). The management device server shall return the index of the last self-configuration status descriptor in the SMP REPORT GENERAL response (see 9.4.3.4), the SMP REPORT SELF-CONFIGURATION STATUS response (see 9.4.3.6), and the SMP DISCOVER LIST response (see 9.4.3.15). The management device server shall wrap the index to 0001h when the highest supported descriptor index has been used.

The management device server shall support self-configuration status descriptor indexes from 0001h to FFFFh. The actual number of self-configuration status descriptors that the management device server maintains for retrieval with the REPORT SELF-CONFIGURATION STATUS request is vendor specific and is indicated by the MAXIMUM NUMBER OF STORED SELF-CONFIGURATION STATUS DESCRIPTORS field defined in the REPORT GENERAL response (see 9.4.3.4). The volatility of these stored descriptors is vendor specific. The management device server shall replace the oldest self-configuration status descriptor with a new one once the number of recorded descriptors exceeds the value indicated by the MAXIMUM NUMBER OF STORED SELF-CONFIGURATION STATUS DESCRIPTORS field.

4.7.5 Enabling multiplexing

A management application client may configure multiplexing (see 5.11) in expander devices. Self-configuring expander devices may configure multiplexing for their own phys. The choice of whether or not to enable multiplexing on a physical link is vendor specific.

If the SAS domain contains all 6 Gbps SAS phys, then the management application clients should disable multiplexing on every phy.

If the SAS domain contains all 3 Gbps SAS phys, then the management application clients should:

- a) multiplex each 6 Gbps physical link into two 3 Gbps logical links; and
- b) not multiplex 3 Gbps physical links.

If the SAS domain contains all 1.5 Gbps SAS phys, then the management application client should:

- a) multiplex each 6 Gbps physical link into two 3 Gbps logical links; and
- b) multiplex each 3 Gbps physical link into two 1.5 Gbps logical links.

NOTE 13 Rate matching is used for 1.5 Gbps connections carried on 3 Gbps logical links.

4.8 Configuration subprocess

4.8.1 Configuration subprocess overview

As part of the discover process (see 4.7), when the management application client discovers an externally configurable expander device, it performs the configuration subprocess to configure the expander routing table in that externally configurable expander device with SAS addresses discovered two or more levels beyond each table routing phy in that externally configurable expander device. A single discover process performs the configuration subprocess at least once per externally configurable expander device.

The routing table in an expander device needs to be configured before connections are able to be established with devices attached two or more levels beyond that expander device.

4.8.2 Allowed expander device topologies

If the management application client detects:

- a) an expander phy with the table routing attribute in an externally configurable expander device; or
- b) an expander phy with the table routing attribute in a self-configuring expander device with the TABLE TO TABLE bit set to zero in the SMP REPORT GENERAL response,

attached to:

- a) an expander phy with either the direct routing attribute or the table routing attribute in either an externally configurable expander device or a self-configuring expander device,

then it shall report an error in a vendor specific manner.

If the management application client detects an overflow of an expander route index, then it shall report an error in a vendor specific manner.

If the route table optimization (see 4.8.3) is disabled and the management application client detects an expander route entry that references the SAS address of the expander device itself (i.e., self-reference), then the management application client shall disable the expander route entry by setting the DISABLE EXPANDER ROUTE ENTRY bit to one in the SMP CONFIGURE ROUTE INFORMATION request (see 9.4.3.27). The management application client shall disable each expander route entry in the expander route table by setting the DISABLE EXPANDER ROUTE ENTRY bit to one in the SMP CONFIGURE ROUTE INFORMATION request (see 9.4.3.27) for each expander phy that has its attached device type set to 000b (i.e., no device attached).

If the management application client detects a port that:

- a) the configuration subprocess has not already configured with a SAS address; and
- b) it has already found attached to another expander device,

then it should use the SMP PHY CONTROL function (see 9.4.3.28) to disable all the expander phys attached to that SAS address except for phys in the expander device with the lowest SAS address. Figure 46 shows some illegal topologies.

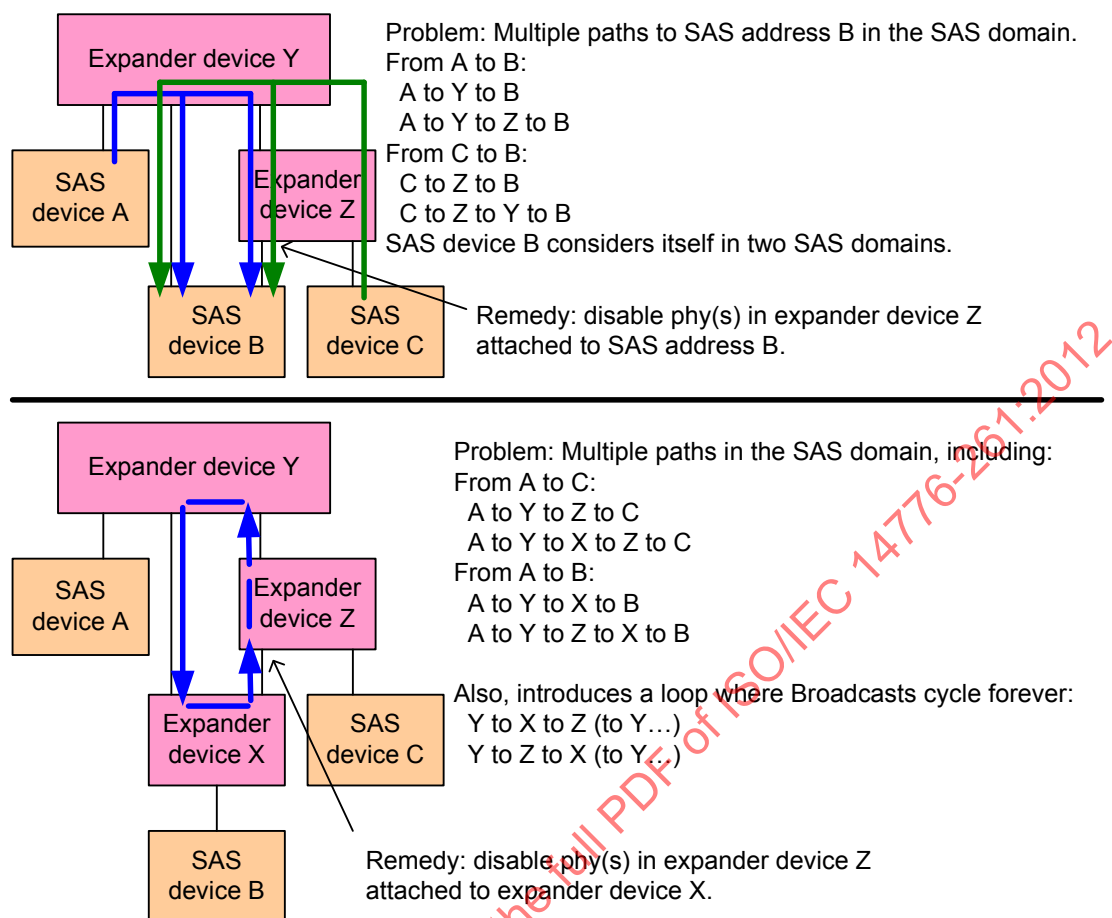


Figure 46 — Examples of invalid topologies

Annex I contains an example implementation of how a management application client may perform the discover process and configuration subprocess.

4.8.3 Externally configurable expander device route table optimization

The management application client shall support a route table optimization that reduces the number of entries required in an expander route table in an externally configurable expander device. The method used to enable and disable the route table optimization is vendor specific.

If the route table optimization is enabled, then the management application client shall exclude discovered SAS addresses from the expander route table when any of the following conditions are met:

- a) in the SMP DISCOVER response (see 9.4.3.10) for the discovered phy:
 - A) the FUNCTION RESULT field is set to a non-zero value (i.e., not SMP FUNCTION ACCEPTED);
- b) in the SMP DISCOVER response for the discovered phy:
 - A) the FUNCTION RESULT field is set to zero (i.e., SMP FUNCTION ACCEPTED);
 - B) the ROUTING ATTRIBUTE field is set to 1h (i.e., subtractive) or 2h (i.e., table); and
 - C) the ATTACHED DEVICE TYPE field is set to zero (i.e., no device attached);
- c) in the SMP DISCOVER response for the discovered phy:
 - A) the FUNCTION RESULT field is set to zero (i.e., SMP FUNCTION ACCEPTED);
 - B) the ROUTING ATTRIBUTE field is set to 1h (i.e., subtractive) or 2h (i.e., table);
 - C) the ATTACHED DEVICE TYPE field is set to a non-zero value (e.g., end device or expander device); and
 - D) the ATTACHED SAS ADDRESS field contains the SAS address of the expander device being configured (i.e., a self-referencing address);

- d) in the SMP DISCOVER response for the discovered phy:
 - A) the FUNCTION RESULT field is set to zero (i.e., SMP FUNCTION ACCEPTED);
 - B) the ROUTING ATTRIBUTE field is set to 1h (i.e., subtractive) or 2h (i.e., table);
 - C) the ATTACHED DEVICE TYPE field is set to a non-zero value (e.g., end device or expander device); and
 - D) the ATTACHED SAS ADDRESS field contains the SAS address of a device directly attached to the expander device being configured;
 or
- e) in the SMP DISCOVER response for the discovered phy:
 - A) the FUNCTION RESULT field is set to zero (i.e., SMP FUNCTION ACCEPTED);
 - B) the ROUTING ATTRIBUTE field is set to 1h (i.e., subtractive) or 2h (i.e., table);
 - C) the ATTACHED DEVICE TYPE field is set to a non-zero value (e.g., end device or expander device); and
 - D) the ATTACHED SAS ADDRESS field contains a SAS address that already exists in the expander route table.

If the discovered SAS address being included in the expander route table is for a device that is not attached (i.e., the ATTACHED DEVICE TYPE field is set to zero (i.e., no device attached) and the ROUTING ATTRIBUTE field is set to 0h (i.e., direct)), then the entry shall be inserted with the ROUTED SAS ADDRESS field set to 00000000 00000000h and the DISABLE EXPANDER ROUTE ENTRY bit set to one (see 9.4.3.27).

If route table optimization is disabled, then all SAS addresses shall be qualified for insertion in the expander route table.

If the management application client supports route table optimization, then the management application client should provide a vendor specific method for initiating a check of the resulting expander route tables. The check should be performed under the following situations:

- a) when an I_T nexus loss occurs for a destination port that is expected to be present;
- b) when a discover process has been completed;
- c) when another SMP initiator port is discovered in the SAS domain; or
- d) when a self-configuring expander device is discovered in the SAS domain.

If the management application client detects an inconsistency in the expander route tables when route table optimization is enabled (e.g., detects entries that appear to have been filled in by a discover process with route table optimization disabled), then the management application client shall report an error in a vendor specific manner and shall disable route table optimization. The management application client should then reinitiate a discover process with route table optimization disabled.

4.8.4 Externally configurable expander device expander route index order

The expander route table in an externally configurable expander device shall be configured for each expander phy that has a table routing attribute.

If the expander phy is not attached to an expander device, then every expander route entry for that expander phy shall be disabled (i.e., the ROUTED SAS ADDRESS field shall be set to 00000000 00000000h and the DISABLE EXPANDER ROUTE ENTRY bit shall be set to one).

If the expander phy is attached to an expander device, then the expander route table shall be configured for that expander phy as follows. For purposes of configuring the expander route table for that phy, the expander devices attached to the expander phy are assigned levels:

- 1) the expander device in which the expander route table is being configured is level 0;
- 2) the attached expander device is considered level 1;
- 3) devices attached to the level 1 expander device, except for the level 0 expander device, are considered level 2;
- 4) devices attached to level 2 expander devices, except for level 1 expander devices, are considered level 3; and
- 5) for each n greater than 3, devices attached to level n-1 expander devices, except for level n-2 expander devices, are considered level n.

The expander route table for each expander phy shall be configured starting from expander route index 0 by level (e.g., if there are three levels, then all level 1 entries first, then all level 2 entries, then all level 3 entries) up to the value of the EXPANDER ROUTE INDEXES field reported by the SMP REPORT GENERAL function (see 9.4.3.4).

If the level 1 expander device has expander phys attached to N phys with qualified SAS addresses (see 4.8.3), then the first N entries shall be used for those SAS addresses in expander phy order (i.e., the addresses attached to lower expander phy numbers first).

For each of the level 2 devices that:

- a) is an expander device attached to M phys with qualified SAS addresses; and
- b) is attached to an expander phy in the level 1 expander device with the table routing attribute,

the next M entries shall be used for the level 2 expander device's qualified SAS addresses in expander phy order (i.e., lower phy numbers first).

This process shall be repeated for all levels of expander devices.

SAS addresses of devices attached beyond expander phys that are attached table-to-table shall not be included in the expander route table. The SAS address of the first expander device that is attached table-to-table shall be included and the SAS address of every device attached beyond that expander device shall not be included.

NOTE 14 Not including those SAS addresses provides compatibility with management application clients compliant with SAS-1.1. End devices in SAS domains containing externally configurable expander devices and table-to-table attachments may not be able to establish connections to each other. End devices in SAS domains containing only self-configuring expander devices supporting table-to-table attachments are able to establish connections to any other end device.

Figure 47 shows an example of a route table that does not include SAS addresses beyond a table-to-table attachment.

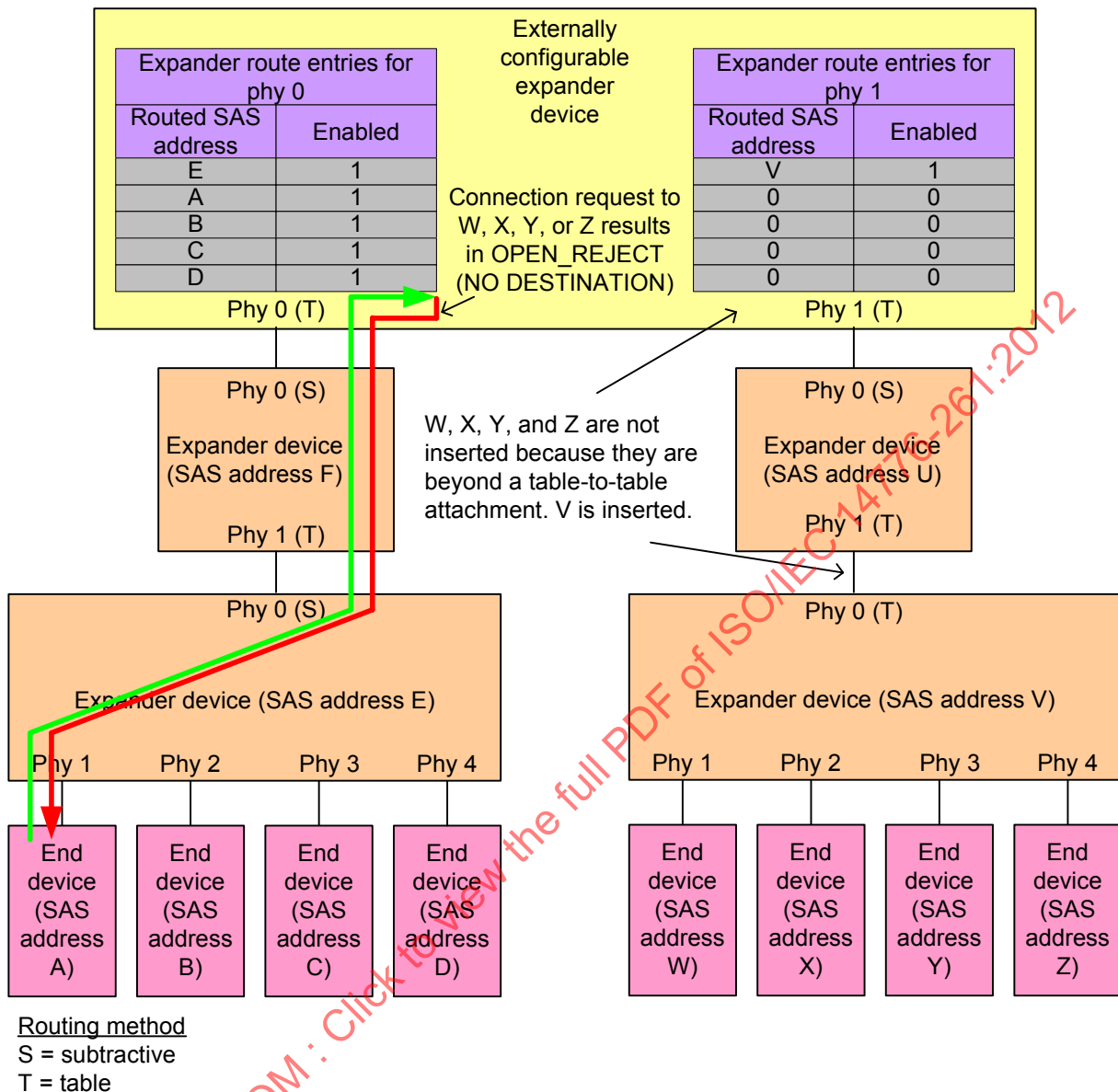


Figure 47 — Externally configurable expander device and table-to-table attachment

After the expander route table has been configured with entries for all levels of expander devices, all remaining expander route entries, if any, shall be disabled (i.e., the ROUTED SAS ADDRESS field shall be set to 00000000 00000000h and the DISABLE EXPANDER ROUTE ENTRY bit shall be set to one). The management application client is not required to disable entries if the topology of expander devices has not changed.

Figure 48 shows a portion of a SAS domain, where phy A in expander device R is being configured.

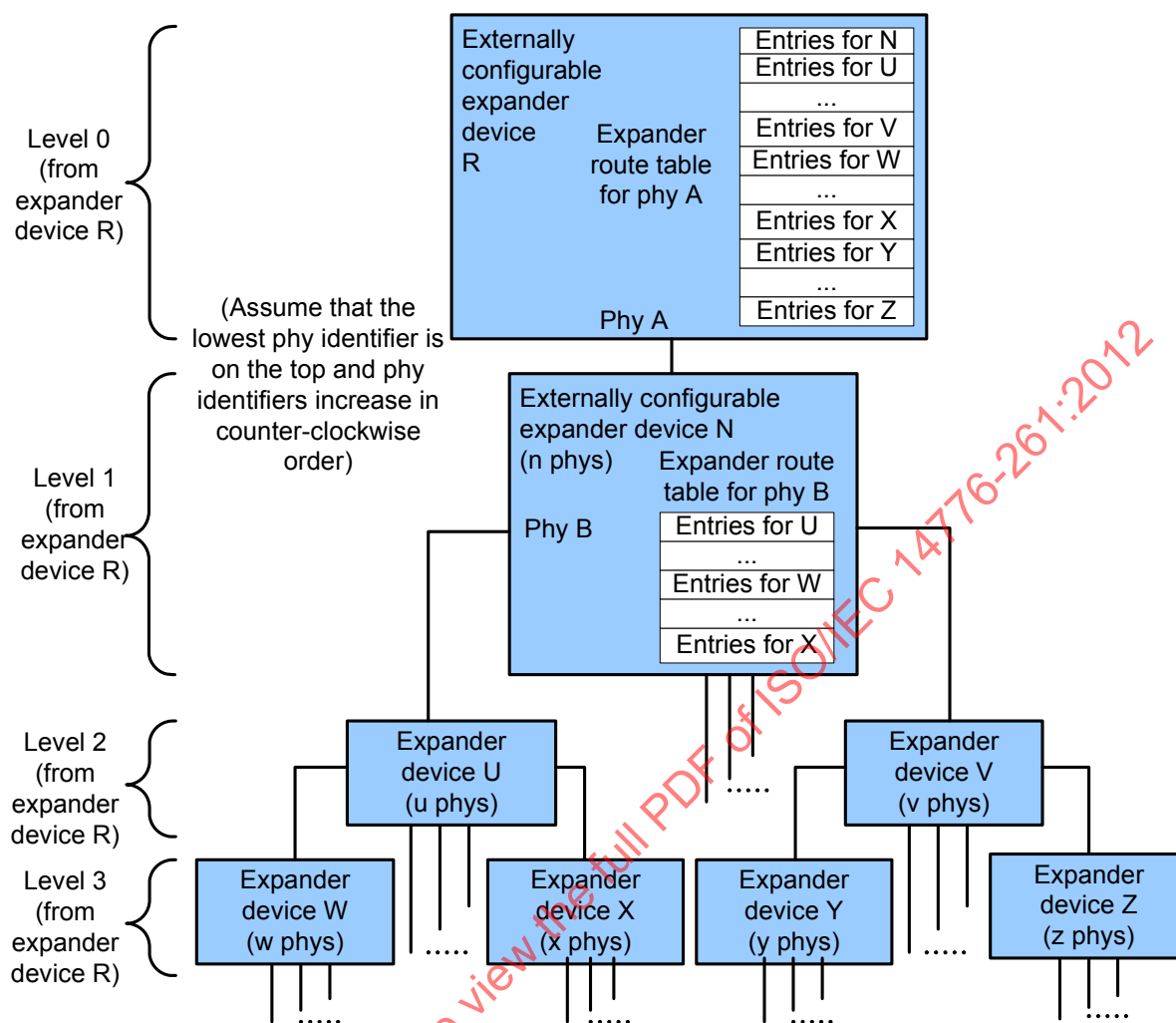


Figure 48 — Expander route index levels example

Table 23 shows how the expander route table is configured for externally configurable expander device R phy A in figure 48.

Table 23 — Expander route table levels for externally configurable expander device R phy A

Expander route index	Expander route entry contents
Level 1 (from expanded device R) entries	
0 .. ($\leq n$ entries)	Qualified SAS addresses attached to expander device N
Level 2 (from expanded device R) entries	
($\leq u$ entries)	Qualified SAS addresses attached to expander device U
...	...additional qualified SAS addresses for expander devices at level 2...
($\leq v$ entries)	Qualified SAS addresses attached to expander device V
Level 3 (from expanded device R) entries	
($\leq w$ entries)	Qualified SAS addresses attached to expander device W

Table 23 — Expander route table levels for externally configurable expander device R phy A

Expander route index	Expander route entry contents
...	...additional qualified SAS addresses for expander devices at level 3...
(≤ x entries)	Qualified SAS addresses attached to expander device X
(≤ y entries)	Qualified SAS addresses attached to expander device Y
...	...additional qualified SAS addresses for expander devices at level 3...
(≤ z entries)	Qualified SAS addresses attached to expander device Z
Entries for additional levels	
...	...
Disabled entries	
...	...

Table 24 shows how the expander route table is configured for externally configurable expander device N phy B in figure 48.

Table 24 — Expander route table levels for externally configurable expander device N

Expander route index	Expander route entry contents
Level 1 (from expanded device N) entries	
(0 .. ≤ u entries)	Qualified SAS addresses attached to expander device U
Level 2 (from expanded device N) entries	
(≤ w entries)	Qualified SAS addresses attached to expander device W
...	...additional qualified SAS addresses for expander devices at level 2...
(≤ x entries)	Qualified SAS addresses attached to expander device X
Entries for additional levels	
...	...
Disabled entries	
...	...

Figure 49 shows an example topology.

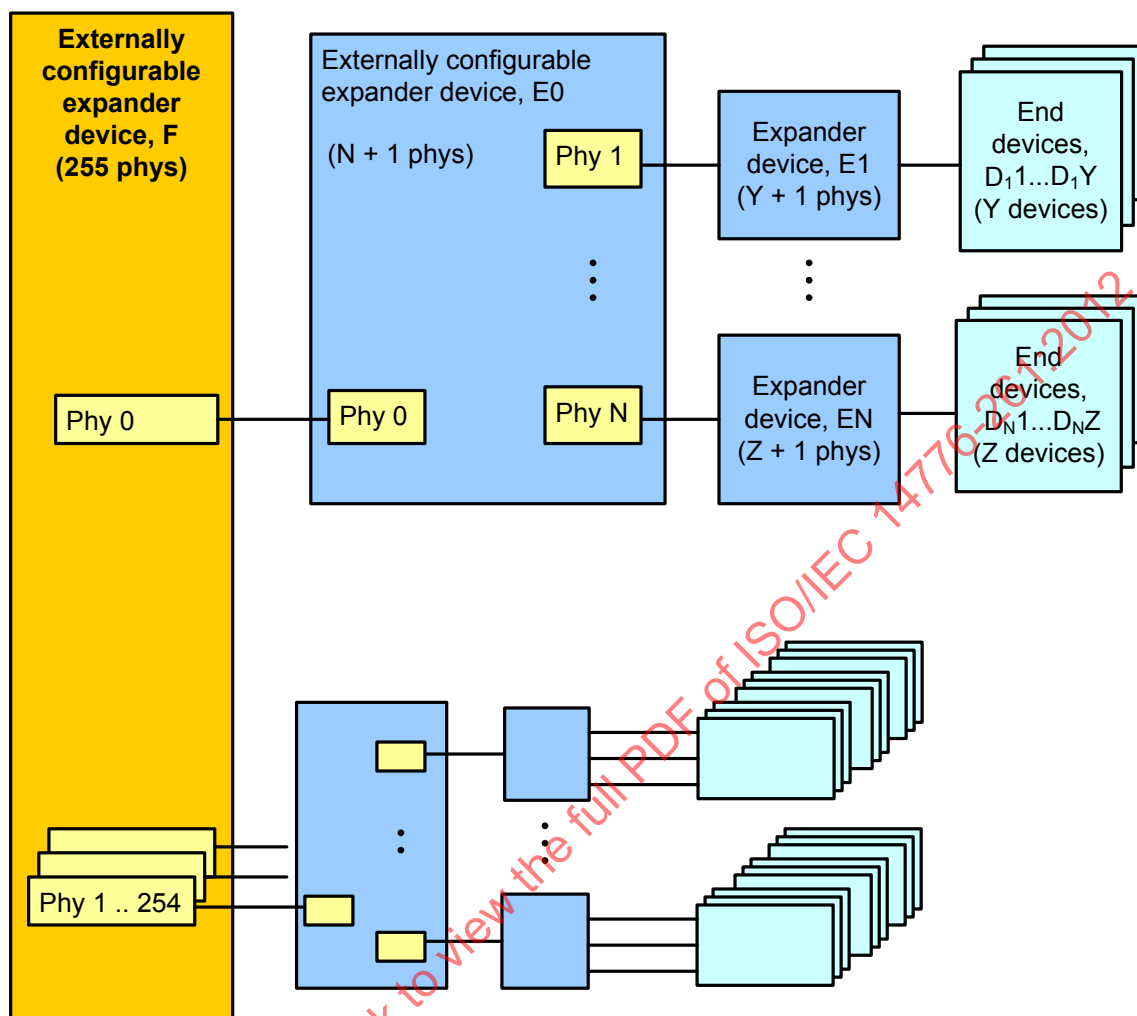


Figure 49 — Expander route index order example

Table 25 shows the expander route index order for externally configurable expander device E0 phy 1 in figure 49, assuming that all phys are present and not subject to exclusion by route table optimization (see 4.8.3).

Table 25 — Expander route entries for externally configurable expander device E0 phy 1

Expander route index	Expander route entry contents
Level 1 entries	
0	SAS address (e.g., D_{11}) of the device attached to phy 1 of expander device E1
1	SAS address (e.g., D_{12}) of the device attached to phy 2 of expander device E1
...	...
$Y - 1$	SAS address (e.g., D_{1Y}) of the device attached to phy Y of expander device E1
Level 2 and beyond	

Table 25 — Expander route entries for externally configurable expander device E0 phy 1

Expander route index	Expander route entry contents
	No entries
Disabled entries	
	Any remaining entries are disabled

Table 26 shows the expander route index order for externally configurable expander device F phy 0 in figure 49, assuming that all phys are present and not subject to exclusion by route table optimization (see 4.8.3).

Table 26 — Expander route entries for externally configurable expander device F phy 0

Expander route index	Expander route entry contents
Level 1 entries	
0	SAS address (e.g., E1) of the device attached to phy 1 of expander device E0
...	...additional qualified SAS addresses for expander device E0...
N - 1	SAS address (e.g., EN) of the device attached to phy N of expander device E0
Level 2 entries	
N	SAS address (e.g., D ₁ 1) of the device attached to phy 1 of expander device E1
...	...additional qualified SAS addresses for expander device E1...
	SAS address (e.g., D ₁ Y) of the device attached to phy Y of expander device E1
...	...additional qualified SAS addresses for expander devices E2 through EN-1...
	SAS address (e.g., D _N 1) of the device attached to phy 1 of expander device EN
...	...additional qualified SAS addresses for expander device EN...
	SAS address (e.g., D _N Z) of the device attached to phy Z of expander device EN
Level 3 and beyond	
	No entries since all devices attached to E1 through EN, except for E0, are end devices
Disabled entries	
	Any remaining entries are disabled

4.9 Zoning

4.9.1 Zoning overview

SAS zoning is implemented by a set of zoning expander devices with zoning enabled that define a zoned portion of a service delivery subsystem (ZPSDS). The zoning expander devices control whether a phy is permitted to participate in a connection to another phy.

Figure 50 shows an example of zoning.

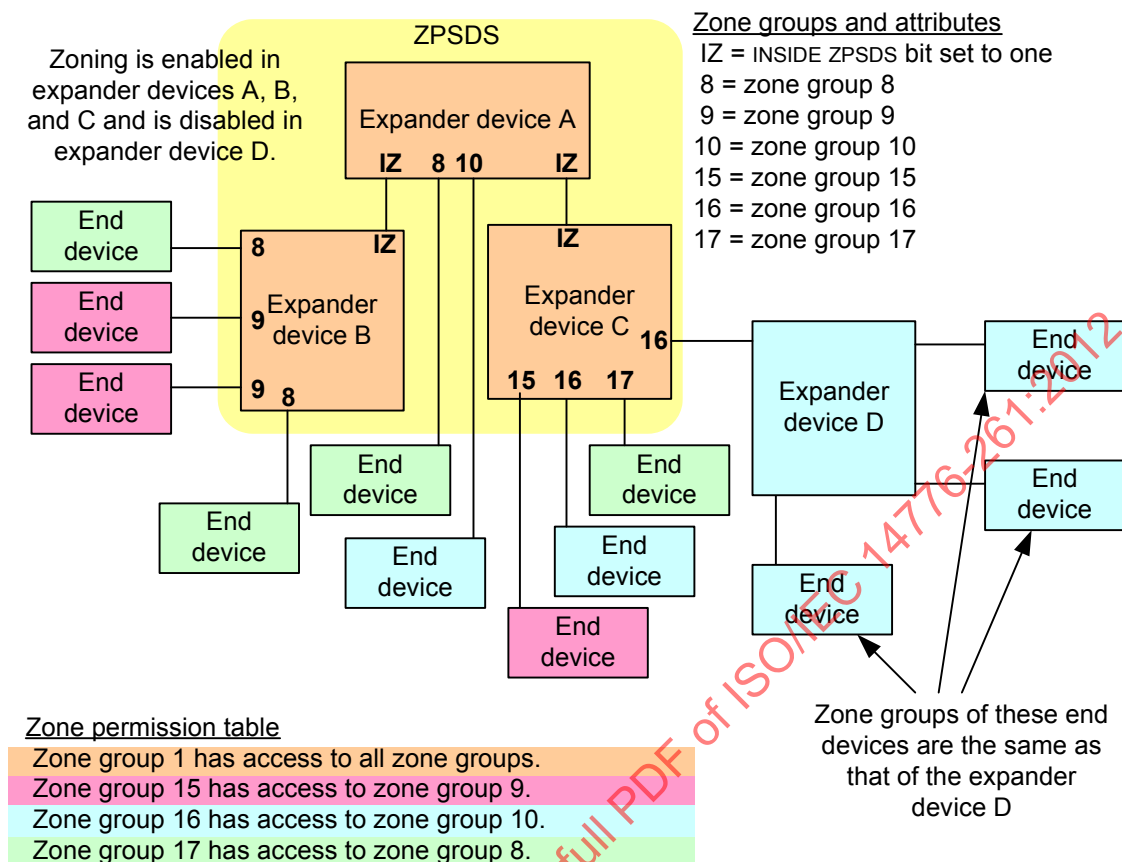
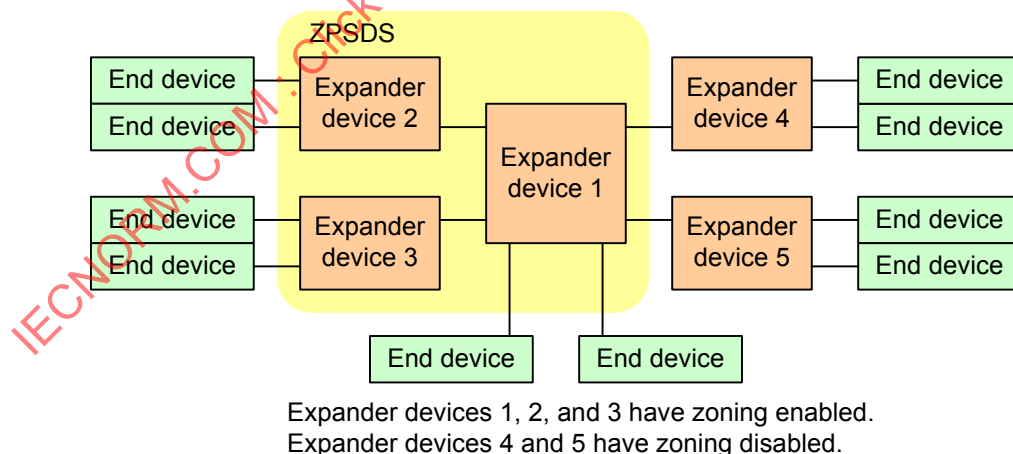
**Figure 50 — Zoning example**

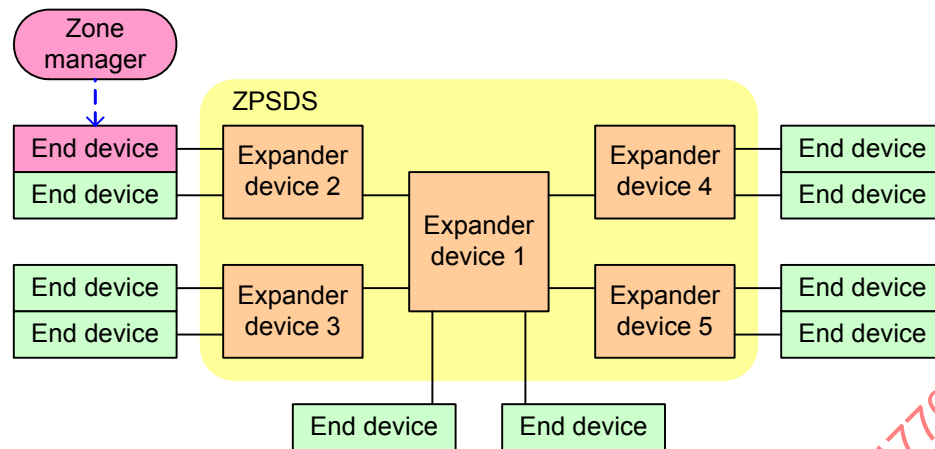
Figure 51 shows an example of one ZPSDS in a SAS domain.

**Figure 51 — One ZPSDS example**

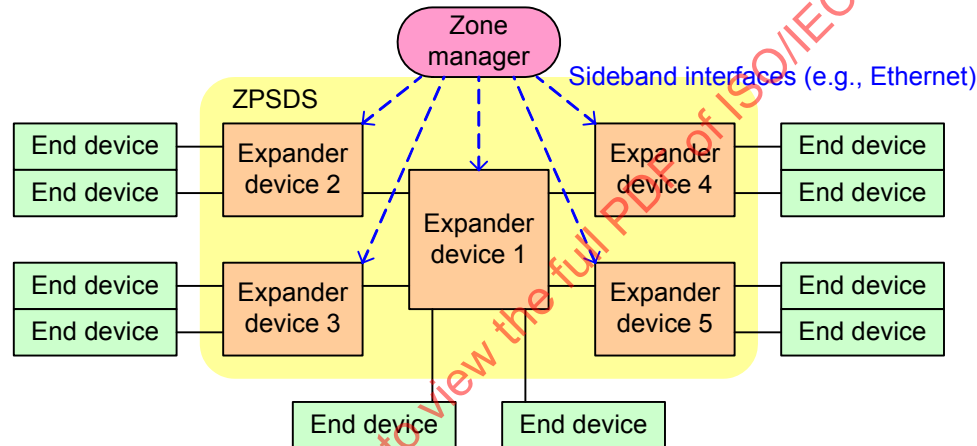
A ZPSDS has a zone manager responsible for its configuration. The zone manager may have access to an end device with a SAS port whose zone group (see 4.9.3.2) has access to zone group 2, or may have access to one or more zoning expander devices through a sideband interface (e.g., Ethernet) outside the scope of this standard. The SAS address reported for a sideband zone manager is 00000000 00000000h.

Figure 52 shows examples of zone manager locations in a SAS domain.

Zone manager attached to an end device with a SAS port whose zone group has access to zone group 2



Zone manager attached directly to the expander devices in the ZPSDS



Zone manager attached directly to one expander device in the ZPSDS

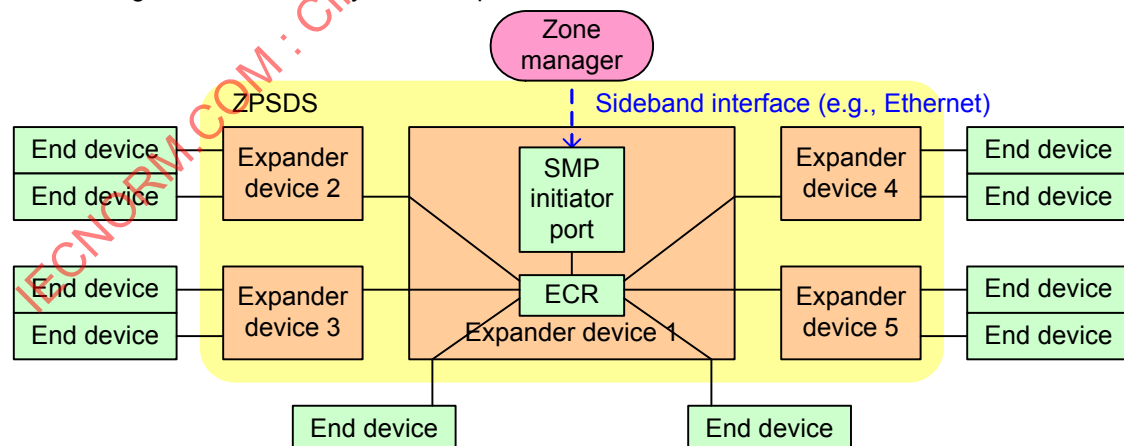


Figure 52 — Zone manager location examples

There may be any number of non-overlapping ZPSDSes in a service delivery subsystem, particularly as a SAS domain is being reconfigured (e.g., as a user is attaching enclosures together). A SAS domain with more than one ZPSDS should be transitory. A ZPSDS may encompass some or all of a service delivery subsystem.

Figure 53 shows an example of three ZPSDSes in a SAS domain.

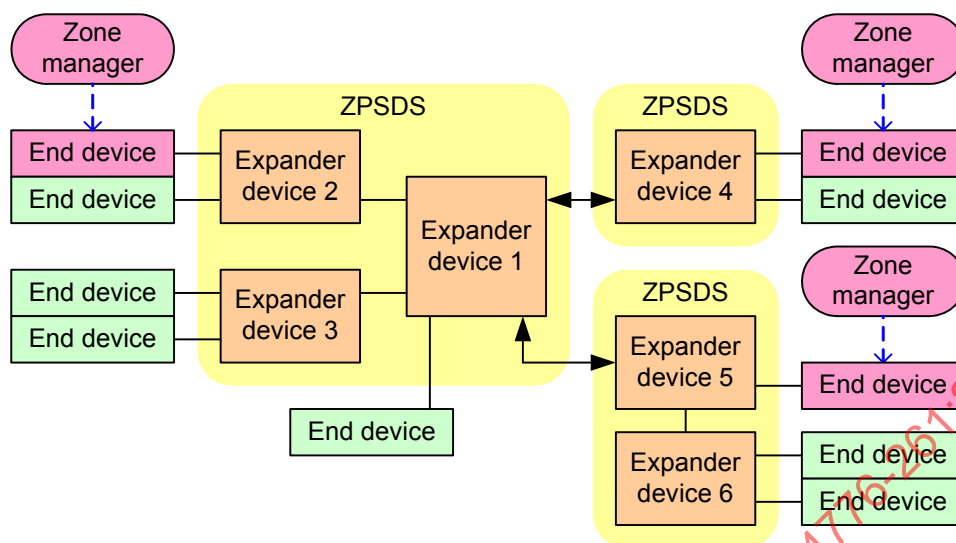


Figure 53 — Three ZPSDSes example

The zone manager assigns zone groups (see 4.9.3.2) to all zoning expander phys inside the ZPSDS. There are 128 or 256 zone groups numbered 0 through 127 or 0 through 255. All phys in a wide port shall be assigned to the same zone group. Zone groups are assigned to zoning expander phys as part of the zone phy information (see 4.9.3.1) and are stored along with SAS addresses in the zoning expander route table (see 4.9.3.4). The zone groups assigned in one ZPSDS have no relationship to the zone groups assigned in another ZPSDS.

The zone manager shall assign each zoning expander phy attached to another zoning expander phy inside a ZPSDS to zone group 1. The zone manager shall assign each zoning expander phy on the boundary of the ZPSDS (i.e., with the INSIDE ZPSDS bit set to zero) to a zone group. All phys in the SAS domain beyond that boundary zoning expander phy are considered to be in the same zone group as that zoning expander phy.

Each zoning expander device contains a zone permission table (see 4.9.3.3) that controls whether a connection is allowed between phys based on their zone groups. As defined in 4.9.3.5, a requested connection shall only be established if the zone permission table indicates that access between the zone group of the source port and the zone group of the destination port is allowed.

The zoning expander route table (see 4.9.3.4) is an extended version of the expander route table (see 4.6.7.4) that also includes the zone group of each SAS address.

Physical presence detection is a mechanism used to allow management access. The definition of physical presence detection is vendor specific (e.g., a user pressing a button or inserting a key).

The zone manager password is a value used to allow management access. The zone manager password is 32 bytes long and is specified in table 27.

Table 27 — Zone manager password

Code	Name	Description
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000h	ZERO	Well-known value that provides access to any zone manager that presents it.
FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFFh	DISABLED	Well-known value that does not provide access to any zone manager (e.g., zone manager password usage is disabled).
All others		Random value, providing access only to a zone manager that presents the correct value.

The expander device:

- a) shall maintain a current value;
- b) shall maintain a shadow value;
- c) may maintain a saved value; and
- d) shall have a default value,

for each of the following settings:

- a) zoning enabled;
- b) the zone permission table; and
- c) zone phy information.

The expander device:

- a) shall maintain a current value;
- b) may maintain a saved value; and
- c) shall have a default value,

for the zone manager password, if any.

Support or lack of support for saved values for one setting does not imply support or lack of support for saved values for any other setting (e.g., the expander device may maintain a saved value for zoning enabled but not for the zone permission table).

For each setting, after power on or expander device reduced functionality, the expander device shall set the current value to the saved value, if any, or the default value, if there is no saved value.

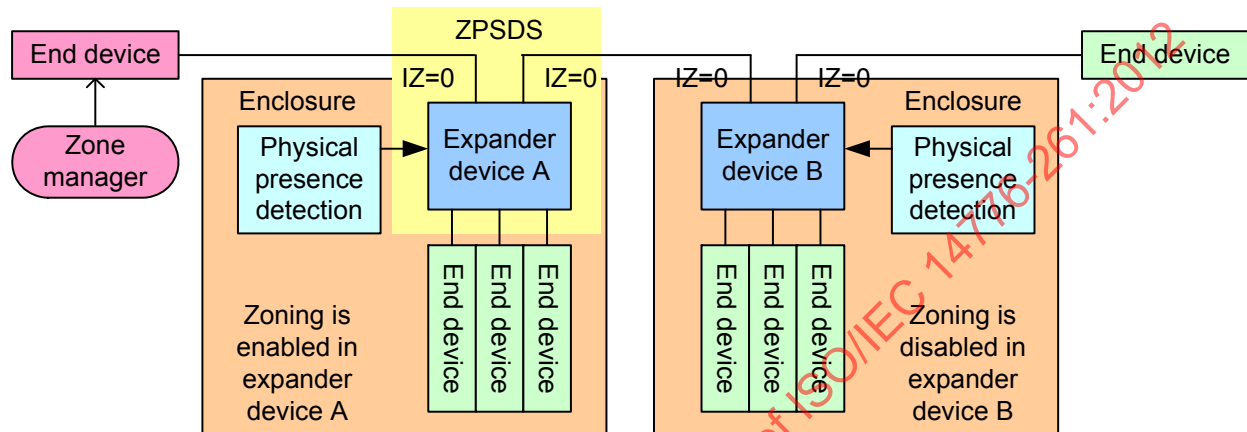
4.9.2 Zoning expander device requirements

In addition to the requirements for expander devices described in 4.6, a zoning expander device shall:

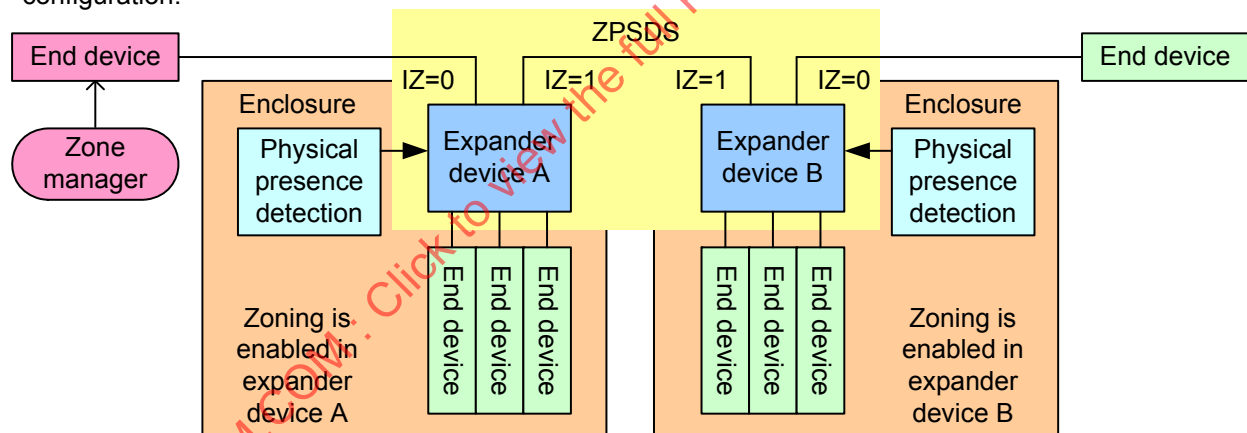
- a) contain a zoning expander route table (see 4.9.3.4);
- b) contain current and shadow zone permission tables that supports 128 or 256 zone groups (see 4.9.3.3);
- c) contain current and shadow zone phy information for each phy;
- d) if zoning is enabled, allow or deny connection requests based on the current zone permission table (see 4.9.3.5);
- e) support fields related to zoning in its SMP REPORT GENERAL response (see 9.4.3.4);
- f) support the zone lock inactivity timer;
- g) be self-configuring;
- h) contain an SMP initiator port (see 4.6.1); and
- i) support zoning-related SMP functions.

A zoning expander device may support physical presence detection and/or a zone manager password to allow management access.

Figure 54 shows an example of two enclosures with physical presence detection where zoning is enabled in the expander device in the left enclosure, but is not enabled in the expander devices in the right enclosure. The zone manager is able to configure zoning in zoning expander device A because the zone group of its SMP initiator port has access to zone group 2. However, it is not able to enable or configure zoning in expander device B unless physical presence is asserted or it presents the correct zone manager password for that expander device.



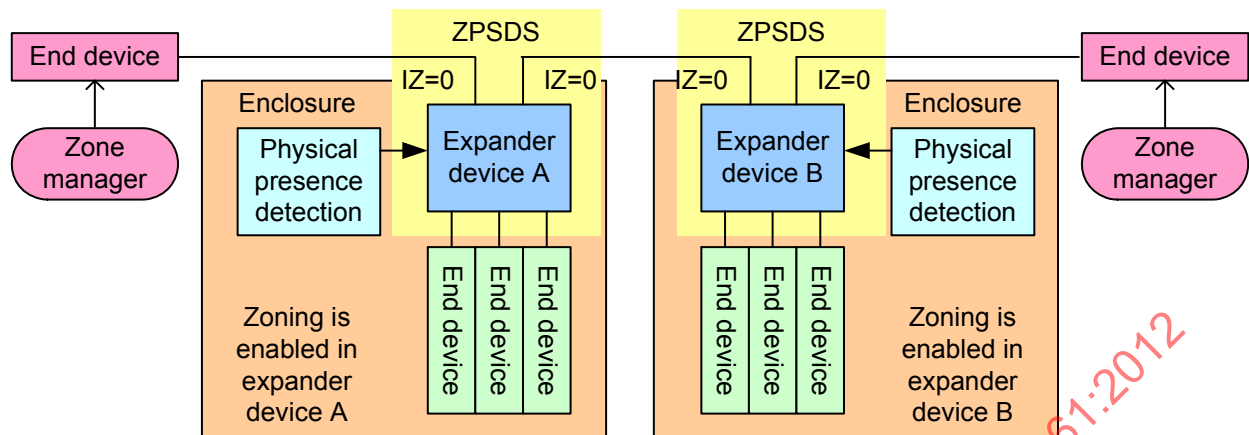
The ZPSDS is extended as shown below after the zone manager on the left completes zone configuration.



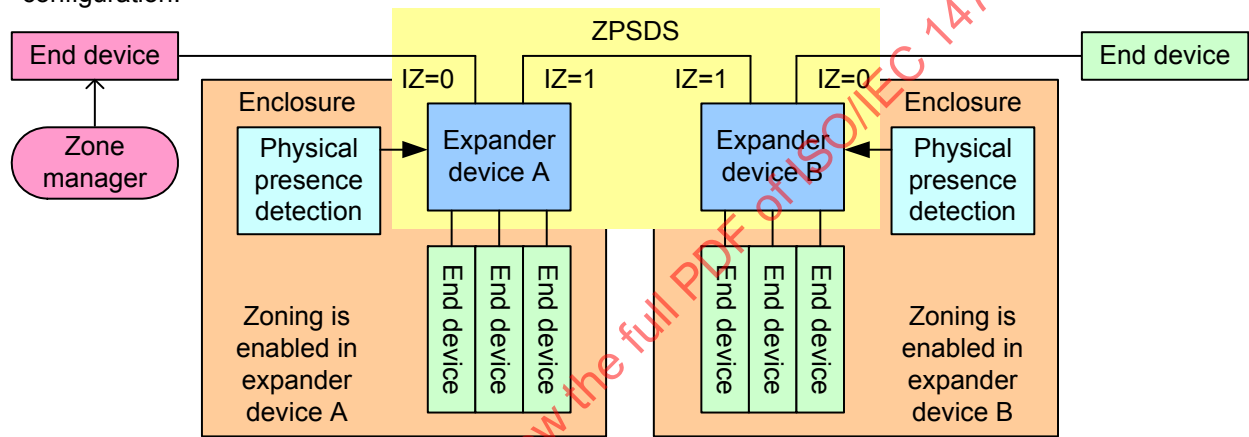
IZ = INSIDE ZPSDS bit in the zone phy information

Figure 54 — Extending a ZPSDS example

Figure 55 shows an example of two enclosures with physical presence detection where zoning is enabled in both expander devices. The zone manager is able to configure zoning in zoning expander device A because the zone group of its SMP initiator port has access to zone group 2. However, it is not able to configure zoning in expander device B unless physical presence is asserted or it presents the correct zone manager password for that expander device.



The ZPSDS is extended as shown below after the zone manager on the left completes zone configuration.



IZ = INSIDE ZPSDS bit in the zone phy information

Figure 55 — Overtaking a ZPSDS example

4.9.3 Zoning operation

4.9.3.1 Zone phy information

Each phy of a zoning expander device shall support the zone phy information fields defined in table 28.

Table 28 — Zone phy information

Field	Description	Recommended default
INSIDE ZPSDS bit	Indicates if the phy is inside or on the boundary of a ZPSDS	N/A ^a
REQUESTED INSIDE ZPSDS bit	Used to establish the boundary of the ZPSDS	0
^a The INSIDE ZPSDS bit is determined from the values exchanged during the link reset sequence.		

Table 28 — Zone phy information

Field	Description	Recommended default
INSIDE ZPSDS PERSISTENT bit	Used to determine the value of the INSIDE ZPSDS bit after a link reset sequence	0
ZONE GROUP PERSISTENT bit	Used to determine the zone group of the phy after a link reset sequence if the INSIDE ZPSDS bit is set to zero	0
ZONE GROUP field	The zone group to which the phy belongs	00h
^a The INSIDE ZPSDS bit is determined from the values exchanged during the link reset sequence.		

Table 29 lists the usage of the current values of the zone phy information fields.

Table 29 — Zone phy information usage

Field	Transmitted in IDENTIFY address frame ^a	Indicated in DISCOVER function and DISCOVER LIST function ^b	Attached value indicated in DISCOVER function ^c	Programmable with the CONFIGURE ZONE PHY INFORMATION function ^d	Changeable by the expander device after a link reset sequence ^e
INSIDE ZPSDS bit	No	Yes	No	No	Yes
REQUESTED INSIDE ZPSDS bit	Yes	Yes	Yes	Yes	Yes
INSIDE ZPSDS PERSISTENT bit	Yes	Yes	Yes	Yes	No
ZONE GROUP PERSISTENT bit	No	Yes	No	Yes	No
ZONE GROUP field	No	Yes	No	Yes	Yes ^f
^a Defined in the IDENTIFY address frame (see 6.8.2). ^b Defined in the DISCOVER response (see 9.4.3.10) and the DISCOVER LIST response SHORT FORMAT descriptor (see 9.4.3.15.4). ^c Defined in the DISCOVER response (see 9.4.3.10). ^d Defined in the zone phy configuration descriptor (see 9.4.3.25.3). Current values are not updated until the activate step (see 4.9.6.4). The saved values are also programmable with this function. ^e See 4.9.4. ^f Only changes to 00h after a link reset sequence. See 4.9.4.					

All phys in an expander port shall have the same zone phy information.

The expander device shall preserve the zone phy information while:

- a) zoning is disabled;
- b) no power loss occurs; and
- c) there is no expander device reduced functionality (see 4.6.8).

The INSIDE ZPSDS bit indicates if the phy is inside or on the boundary of a ZPSDS. An INSIDE ZPSDS bit set to zero indicates that:

- a) zoning is disabled;
- b) the phy is attached to an end device;
- c) the phy is attached to an expander device that does not support zoning;

- d) the phy is attached to an expander device that supports zoning, but zoning is disabled; or
- e) the phy is attached to an expander device that supports zoning, zoning is enabled, but is outside the ZPSDS (i.e., is in another ZPSDS).

An INSIDE ZPSDS bit set to one indicates that the phy is attached to a zoning expander device with zoning enabled and is thus inside a ZPSDS. The INSIDE ZPSDS bit only changes following a link reset sequence (see 4.9.4), based on:

- a) the REQUESTED INSIDE ZPSDS bit;
- b) the REQUESTED INSIDE ZPSDS bit received in the incoming IDENTIFY address frame (see 6.8.2);
- c) the INSIDE ZPSDS PERSISTENT bit; and
- d) the INSIDE ZPSDS PERSISTENT bit received in the incoming IDENTIFY address frame.

The REQUESTED INSIDE ZPSDS bit is used to establish the boundary of the ZPSDS. The REQUESTED INSIDE ZPSDS bit is used to determine the values of other zone phy information fields after a link reset sequence (see 4.9.4).

The INSIDE ZPSDS PERSISTENT bit is used to determine the value of the INSIDE ZPSDS bit after a link reset sequence (see 4.9.4).

The ZONE GROUP field contains the zone group to which the phy belongs. The zone group of the SMP initiator port and SMP target port in a zoning expander device shall be 01h. 4.9.3.2 defines more about zone groups.

The ZONE GROUP PERSISTENT bit is used to determine the method of determining the zone group of the phy after a link reset sequence if the INSIDE ZPSDS bit is set to zero (see 4.9.4).

4.9.3.2 Zone groups

The zone groups are defined in table 30.

Table 30 — Zone groups

Zone group	Configurable in the zone permission table ^a	Description
0	No	Phys in zone group 0 have access to phys in zone group 1 and do not have access to phys in other zone groups.
1	No	Phys in zone group 1 have access to phys in all zone groups.
2	Yes	Phys in zone group 2 have access to phys in the zone groups indicated by the zone permission table. A management device server in a zoning expander device with zoning enabled only allows management application clients using phys in zone groups with access to zone group 2 to perform the following SMP functions: a) CONFIGURE GENERAL (see 9.4.3.18); b) ZONE LOCK (see 9.4.3.21); and c) SMP zone configuration functions (see 4.9.6.1) performed while the zoning expander device is locked. A management device server in a zoning expander device with zoning enabled only allows management application clients to perform certain SMP phy-based control and configuration functions (e.g., PHY CONTROL, PHY TEST FUNCTION, and CONFIGURE PHY EVENT) if the zone group of the management application client's phy has access to zone group 2 or the zone group of the specified phy.
3	Yes	Phys in zone group 3 have access to phys in the zone groups indicated by the zone permission table. A management device server in a zoning expander device with zoning enabled only allows management application clients using a phy in a zone group with access to zone group 3 to perform certain SMP zoning-related functions (i.e., ZONED BROADCAST (see 9.4.3.20)).
4 to 7	Reserved	
8 to 255	Yes	Phys in zone groups 8 through 255 have access to phys in the zone groups indicated by the zone permission table.
^a A zone group defined as configurable is able to be changed with the SMP CONFIGURE ZONE PERMISSION TABLE function (see 9.4.3.26).		

4.9.3.3 Zone permission table

The zone permission table specifies access permission between zone groups. If a bit in the zone permission table is set to one, then connection requests shall be permitted between phys in the zone groups. If a bit in the zone permission table is set to zero, then connection requests between phys in the zone groups shall be rejected with OPEN_REJECT (ZONE VIOLATION) or OPEN_REJECT (RETRY) as described in 4.9.3.5.

The zone permission table structure is shown in table 31.

Table 31 — Zone permission table

Destination zone group (i.e., d)	Source zone group (i.e., s) ^{a b}				
	0	1	2 to 3	4 to 7	8 to (z-1) ^c
0	0	1	0	0	0
1	1	1	1	1	1
2 to 3	0	1	ZP[s = 2 to 3, d = 2 to 3]	Reserved	ZP[s = 8 to (z-1), d = 2 to 3]
4 to 7	0	1	Reserved	Reserved	Reserved
8 to (z-1) ^c	0	1	ZP[s = 2 to 3, d = 8 to (z-1)]	Reserved	ZP[s = 8 to (z-1), d = 8 to (z-1)]
^a Shading identifies configurable zone groups. ^b All reserved ZP bits shall be set to zero (e.g., bits ZP[4 to 7, 4 to (z-1)] are set to zero). ^c The number of zone groups (i.e., z) is reported in NUMBER OF ZONE GROUPS field in the REPORT GENERAL response (see 9.4.3.4).					

A ZP[s, d] bit set to one specifies that the source zone group (i.e., s) has permission to access the destination zone group (i.e., d). A ZP[s, d] bit set to zero specifies that the source zone group (i.e., s) does not have permission to access the destination zone group (i.e., d).

ZP[s, d] shall be set to the same value as ZP[d, s].

The zoning expander device:

- a) shall preserve the zone permission table while zoning is disabled; and
- b) may or may not preserve the zone permission table through power loss and expander device reduced functionality.

If the zoning expander device preserves whether or not zoning is enabled and does not preserve the zone permission table, then it shall set the current zone permission table to grant minimal permissions after power on or expander device reduced functionality as specified in table 32.

Table 32 — Zone permission table granting minimal permissions

Destination zone group (i.e., d)	Source zone group (i.e., s) ^{a b}				
	0	1	2 to 3	4 to 7	8 to (z-1) ^c
0	0	1	0	0	0
1	1	1	1	1	1
2 to 3	0	1	0	Reserved	0
4 to 7	0	1	Reserved	Reserved	Reserved
8 to (z-1) ^c	0	1	0	Reserved	0
^a Shading identifies configurable zone groups. ^b All reserved ZP bits shall be set to zero (e.g., bits ZP[4 to 7, 4 to (z-1)] are set to zero). ^c The number of zone groups (i.e., z) is reported in NUMBER OF ZONE GROUPS field in the REPORT GENERAL response (see 9.4.3.4).					

If a zone manager enables zoning on zoning capable expander devices that report different values in the NUMBER OF ZONE GROUPS field in the REPORT GENERAL response (see 9.4.3.4) (e.g., some support 128 and others support 256), then the zone manager shall:

- configure all zoning enabled expander devices contained within the ZPSDS to use the highest commonly supported number of zone groups (e.g., 128);
- configure the zone phy information in all the zoning expander devices to set each phy to a zone group less than the highest commonly supported number of zone groups; and
- configure the zone permission table in all the zoning expander devices to set each entry to zero that is higher than the highest commonly supported number of zone groups.

4.9.3.4 Zoning expander route table

A zoning expander route table is an expander-based expander route table (see 4.6.7.4) that is able to hold the zone group of each routed SAS address.

Figure 56 shows a representation of the zoning expander route table.

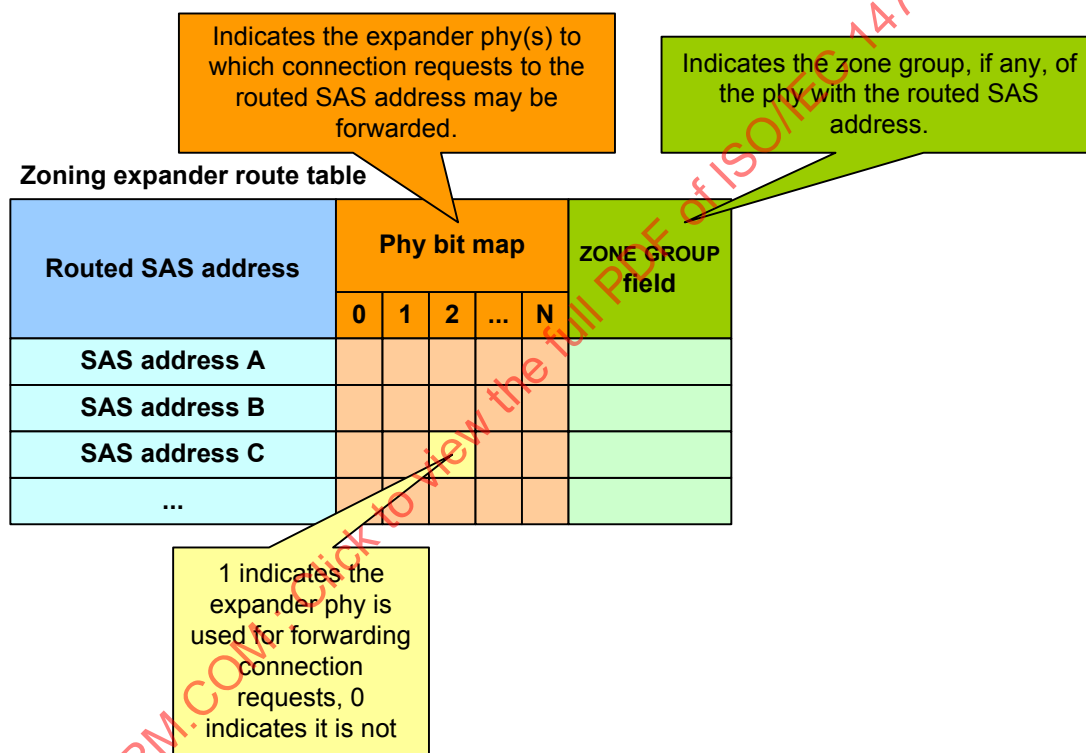


Figure 56 — Zoning expander route table

The zoning expander route table:

- shall include discovered SAS addresses discovered behind each expander phy with the ROUTING ATTRIBUTE field set to 2h (i.e., table) in the DISCOVER response; and
- may include discovered SAS addresses discovered behind expander phys with the ROUTING ATTRIBUTE field set to 1h (i.e., subtractive) in the DISCOVER response as long as they do not prevent inclusion of SAS addresses for expander phys with the ROUTING ATTRIBUTE field set to 2h (i.e., table). The determination of which such SAS addresses to include is vendor specific.

The total number of routed SAS addresses shall not exceed the value indicated in the MAXIMUM NUMBER OF ROUTED SAS ADDRESSES field in the REPORT GENERAL response.

4.9.3.5 Source zone group and destination zone group determination

When a zoning expander device with zoning enabled receives an OPEN address frame (see 6.8.3):

- a) the zone group of the source port (i.e., s) is identified as defined in table 33; and
- b) the zone group of the destination port (i.e., d) is identified as defined in table 34.

If the ZP[s, d] bit is set to one, then access between the source and destination phys shall be permitted, and the zoning expander device shall perform the ECM arbitration procedure. If the ZP[s, d] bit is set to zero, then access between the source and destination phys is not permitted and the zoning expander device shall transmit an OPEN_REJECT in response to the connection request as follows:

- a) OPEN_REJECT (RETRY) if the zoning expander device is locked; and
- b) OPEN_REJECT (ZONE VIOLATION) if the zoning expander device is unlocked.

Zoning expander devices with zoning enabled shall follow the rules in table 33 to determine the zone group of the source port.

Table 33 — Source zone group determination

INSIDE ZPSDS bit of the expander phy that received the OPEN address frame	Source zone group
0	Zone group of the receiving expander phy
1	Source zone group specified by the SOURCE ZONE GROUP field in the received OPEN address frame

Zoning expander devices with zoning enabled shall follow the rules in table 34 to determine the zone group of the destination port.

Table 34 — Destination zone group determination

Routing method of the destination expander phy	Destination zone group
Direct	Zone group of the destination expander phy
Subtractive	If the destination SAS address is in the zoning expander route table, then the zone group stored in the zoning expander route table for the destination SAS address. If the destination SAS address is not in the zoning expander route table, then the zone group of the destination expander phy (i.e., the subtractive routing phy).
Table	The zone group stored in the zoning expander route table for the destination SAS address.

4.9.4 Zone phy information and link reset sequences

At the completion of a link reset sequence (see 4.4), if a SATA device is attached to an expander phy, then the zoning expander device with zoning enabled shall set the INSIDE ZPSDS bit to zero for that expander phy.

At the completion of a link reset sequence, if a SATA device is not attached to an expander phy, then the zoning expander device with zoning enabled shall update the current zone phy information fields as defined in table 35 based on:

- the REQUESTED INSIDE ZPSDS bit and the INSIDE ZPSDS PERSISTENT bit in the zone phy information (i.e., the bits transmitted in the outgoing IDENTIFY address frame (see 6.8.2)); and
- the REQUESTED INSIDE ZPSDS bit and INSIDE ZPSDS PERSISTENT bit received in the incoming IDENTIFY address frame.

Table 35 — Zone phy information fields after a link reset sequence

REQUESTED INSIDE ZPSDS bit		INSIDE ZPSDS PERSISTENT bit		Zone phy information field changes
Transmitted	Received	Transmitted	Received	
0	0 or 1	0 or 1	0 or 1	The zoning expander device shall set the INSIDE ZPSDS bit to zero.
1	0			
1	1	0	0	If the SAS ADDRESS field or the REQUESTED INSIDE ZPSDS bit received in the IDENTIFY address frame during the identification sequence is different from the SAS ADDRESS field or REQUESTED INSIDE ZPSDS bit received prior to the completion of the link reset sequence, then the zoning expander device shall:
		0	1	
		1	0	a) set the REQUESTED INSIDE ZPSDS bit to zero; and b) set the INSIDE ZPSDS bit to zero.
		1	1	If the SAS ADDRESS field and the REQUESTED INSIDE ZPSDS bit received in the IDENTIFY address frame during the identification sequence is the same as the SAS ADDRESS field and the REQUESTED INSIDE ZPSDS bit received prior to the completion of the link reset sequence, then the zoning expander device shall: a) set the INSIDE ZPSDS bit to one; and b) set the ZONE GROUP field to 01h.
		1	1	The zoning expander device shall: a) set the INSIDE ZPSDS bit to one; and b) set the ZONE GROUP field to 01h.

If the ZONE GROUP PERSISTENT bit is set to one, then a link reset sequence (see 4.4) shall not cause the zone group of an expander phy to change unless the INSIDE ZPSDS bit changes from zero to one as specified in table 35. If the ZONE GROUP PERSISTENT bit is set to zero, then table 36 specifies events based on the initial condition of an expander phy that shall cause a zoning expander device with zoning enabled to change the ZONE GROUP field of the expander phy to its reset value (i.e., the saved value, if any, or the default value (e.g., 00h) if there is no saved value).

Table 36 — Conditions that cause the ZONE GROUP field to be reset if the ZONE GROUP PERSISTENT bit is set to zero

Initial condition	Event after the initial condition is established
Completed link reset sequence with a SAS device attached	A subsequent link reset sequence completes and: a) any of the following fields received in the IDENTIFY address frame (see 6.8.2) during the identification sequence are different from their values prior to the completion of the link reset sequence: A) SAS ADDRESS field; or B) REQUESTED INSIDE ZPSDS field; - or b) a SATA device is attached.
Completed link reset sequence with a SATA device attached	Either: a) a subsequent link reset sequence completes and: A) a hot-plug timeout (see 5.7.5) occurred between the time of the initial condition and the time the link reset sequence completed; B) the zoning expander device has detected the possibility that a new SATA device has been inserted. The method of detection is outside the scope of this standard (e.g., an enclosure Services process reports a change in the ELEMENT STATUS CODE field in the Device or Array Device element (see SES-3), or a change in the WORLD WIDE NAME field in the attached SATA device's IDENTIFY (PACKET) DEVICE data (see ACS-2); or C) a SAS phy or expander phy is attached; - or b) the expander phy is disabled with the SMP PHY CONTROL function (see 9.4.3.28) DISABLE phy operation.

4.9.5 Broadcast processing in a zoning expander device with zoning enabled

The BPP determines the source zone group(s) of the Broadcast as follows:

- a) if the BPP receives a Broadcast Event Notify request from an expander logical phy (i.e., a zoning expander logical phy received a BROADCAST primitive sequence), then the Broadcast has a single source zone group set to the zone group of that expander phy; or
- b) if the BPP receives a message from the management device server indicating that the management device server received an SMP ZONED BROADCAST request (see table 267) from an SMP initiator port that has access to zone group 3, then the Broadcast has each of the source zone groups specified in the SMP ZONED BROADCAST request.

The BPP forwards the Broadcast to each expander port other than the one on which the Broadcast was received (i.e., the expander port that received the BROADCAST primitive sequence or SMP ZONED BROADCAST request) if:

- a) the Broadcast is not a Broadcast (Zone Activate) and any of the source zone groups have access to the zone group of the expander port;
- b) the Broadcast is a Broadcast (Zone Activate), the BPP is in a locked zoning expander device, the INSIDE ZPSDS bit is set to one, and the source zone group has access to zone group 2; or
- c) the Broadcast is a Broadcast (Zone Activate), the BPP is not in a locked zoning expander device, and any of the source zone groups have access to the zone group of the expander port.

To forward a Broadcast to an expander port:

- a) if the expander port's INSIDE ZPSDS bit is set to one, then the BPP shall request that the SMP initiator port establish a connection on at least one phy in the expander port to the SMP target port of the

attached expander device and transmit an SMP ZONED BROADCAST request specifying the source zone group(s); or

- b) if the expander port's INSIDE ZPSDS bit is set to zero, then the BPP shall send a Transmit Broadcast message to at least one phy in the expander port, causing it to transmit a BROADCAST primitive sequence.

4.9.6 Zone configuration

4.9.6.1 Zone configuration overview

Zoning expander devices implement a lock to coordinate zoning configuration by zone manager(s).

There are four steps in the zone configuration process:

- 1) lock (see 4.9.6.2);
- 2) load (see 4.9.6.3);
- 3) activate (see 4.9.6.4); and
- 4) unlock (see 4.9.6.5).

The management device server in a zoning expander device only accepts SMP zone configuration function requests, SMP ZONE ACTIVATE requests, and SMP ZONE UNLOCK requests while it is locked, and only accepts SMP zone configuration function requests from the zone manager that locked the zoning expander device (i.e., the active zone manager). SMP zone configuration functions change zoning expander shadow values. When changes are complete, the zone manager activates the changes and the zoning expander device sets the zoning expander current values equal to the zoning expander shadow values. The zone manager then unlocks the zoning expander devices.

For a ZPSDS to function correctly, all zoning expander devices within the ZPSDS are required to have identical values in their zone permission tables. To change zone permission tables, a zone manager device locks all zoning expander devices in a ZPSDS.

To change zone phy information, a zone manager locks only the zoning expander devices containing the phys to be changed.

When a zoning expander device with zoning disabled is being added to a ZPSDS (see figure 54 in 4.9.1) or two or more ZPSDSes are being merged (see figure 55 in 4.9.1), the zone manager locks all of the zoning expander devices that are to be included in the final ZPSDS. The zone manager configures the zone phy information in each zoning expander device (e.g., sets the REQUESTED INSIDE ZPSDS bit to one for phys inside the final ZPSDS) and configures all of the zone permission tables to be identical.

If the zone lock inactivity timer expires, then the zoning expander device performs the unlock step. The zoning expander device is unlocked and the zoning expander shadow values are not activated.

4.9.6.2 Lock step

The lock step ensures that the same zone manager locks each zoning expander device. A zone manager sends the SMP ZONE LOCK request (see 9.4.3.21) to lock a zoning expander device. A zoning expander device is locked while the ZONE LOCKED bit is set to one in the SMP REPORT GENERAL response and after the SAS address of the zone manager has been stored. The management device server in a locked zoning expander device processes SMP zone configuration function requests, SMP ZONE ACTIVATE requests, and SMP ZONE UNLOCK requests.

If more than one zone manager attempts to lock a group of zoning expander devices, then the following rules ensure that any concurrent requests are resolved:

- a) If the first SMP ZONE LOCK response received by a zone manager has the FUNCTION RESULT field set to ZONE LOCK VIOLATION (see 9.4.3.3), then the group of zoning expander devices is locked by another zone manager, and the zone manager should originate no further requests until it receives a Broadcast (Change);
- b) If at least one SMP ZONE LOCK request is successful and at least one other response has:
 - A) the FUNCTION RESULT field set to ZONE LOCK VIOLATION (see 9.4.3.3); and
 - B) the ZONE CONFIGURING bit set to one (see 4.9.6.3),

then at least one zoning expander device is locked and being configured by another zone manager. The zone manager that failed to lock the zoning expander devices should unlock all zoning expander devices that it has locked. When a Broadcast (Change) is received, then the zone manager should retry the lock step; and

- c) If at least one SMP ZONE LOCK request is successful and at least one other response has:
 - A) the FUNCTION RESULT field set to ZONE LOCK VIOLATION (see 9.4.3.3); and
 - B) the ZONE CONFIGURING bit set to zero,

then another zone manager has locked at least one zoning expander device in the group of zoning expander devices, and the zone manager shall evaluate the ACTIVE ZONE MANAGER SAS ADDRESS field in the SMP ZONE LOCK response as follows:

- A) if the returned SAS address has a lower numeric value than the SMP port SAS address of the zone manager, then the zone manager with the higher numeric value SAS address shall repeat the SMP ZONE LOCK request to all zoning expander devices that it has not already locked until all required zoning expander devices are locked, or until a Broadcast (Change) is received; or
- B) if the returned ACTIVE ZONE MANAGER SAS ADDRESS field has a higher numeric value than the SMP port SAS address of the zone manager, then the zone manager with the lower numeric value SAS address shall originate an SMP ZONE UNLOCK request to unlock all zoning expander devices that it locked.

The lock step is complete after a zone manager receives a successful SMP ZONE LOCK response from all required zoning expander devices.

4.9.6.3 Load step

The load step stores SMP zone configuration information as zoning expander shadow values. A zoning expander device only processes SMP zone configuration function requests originated by the active zone manager while it is locked.

The SMP zone configuration functions are:

- a) SMP CONFIGURE ZONE PHY INFORMATION (see 9.4.3.25);
- b) SMP CONFIGURE ZONE PERMISSION TABLE (see 9.4.3.26); and
- c) SMP ENABLE DISABLE ZONING (see 9.4.3.19).

After a locked zoning expander device processes any SMP zone configuration function request, it sets the ZONE CONFIGURING bit to one in the SMP REPORT GENERAL response (see 9.4.3.4).

While the ZONE CONFIGURING bit is set to one, the expander device shall return OPEN_REJECT (RETRY) for any connection requests that would otherwise have resulted in OPEN_REJECT (ZONE VIOLATION) (see 4.6.6.3).

SMP zone configuration functions change the zoning expander shadow values and do not affect the zoning expander current values. The zoning expander shadow values become zoning expander current values during the activate step (see 4.9.6.4).

If the active zone manager receives a response to an SMP zone configuration function with the FUNCTION RESULT field set to ZONE LOCK VIOLATION (see 9.4.3.3), then it should unlock all locked zoning expander devices.

The load step may be skipped when a locked zoning expander device is unlocked:

- a) by a zone manager with a higher SAS address during the lock step (see 4.9.6.2); or
- b) because the zone lock inactivity timer expires.

4.9.6.4 Activate step

The activate step:

- a) saves the zoning expander shadow values, if saving was requested;
- b) copies the zoning expander shadow zone permission table to the current zone permission table;

- c) for each phy with the INSIDE ZPSDS bit set to zero, copies the shadow zone phy information to the current zone phy information; and
- d) for each phy with the INSIDE ZPSDS bit set to one:
 - A) sets the current zone group to 01h; and
 - B) for the remaining fields in the zone phy information, copies the shadow zone phy information to the current zone phy information.

The active zone manager originates one of the following:

- a) a Broadcast (Zone Activate) (see 4.1.13); or
- b) an SMP ZONE ACTIVATE request (see 9.4.3.22) to all locked zoning expander devices.

After a locked zoning expander device receives a Broadcast (Zone Activate) or processes an SMP ZONE ACTIVATE request the zoning expander device shall:

- a) copy the zoning expander shadow zone permission table to the current zone permission table;
- b) for each phy with the INSIDE ZPSDS bit set to zero, copy the shadow zone phy information to the current zone phy information; and
- c) for each phy with the INSIDE ZPSDS bit set to one:
 - A) set the current zone group to 01h; and
 - B) for the remaining fields in the zone phy information, copy the shadow zone phy information to the current zone phy information.

If the active zone manager receives an SMP ZONE ACTIVATE response with the FUNCTION RESULT field set to ZONE LOCK VIOLATION (see 9.4.3.3), then it should unlock all locked zoning expander devices.

The activate step may be skipped when a locked zoning expander device is unlocked:

- a) by a zone manager with a higher SAS address during the lock step (see 4.9.6.2); or
- b) because the zone lock inactivity timer expires.

4.9.6.5 Unlock step

The unlock step ensures that:

- a) the active zone manager unlocks the locked zoning expander devices; or
- b) if the zone manager fails, then the zone lock inactivity timer expires, and the zoning expander devices unlock.

If the active zone manager originated Broadcast (Zone Activate), then it sends an SMP ZONE UNLOCK request (see 9.4.3.23) with the ACTIVATE REQUIRED bit set to one to each of the locked zoning expander devices. This ensures that the activate step precedes the unlock step in each zoning expander device. If the active zone manager receives an SMP ZONE UNLOCK response with the FUNCTION RESULT field set to NOT ACTIVATED (see 9.4.3.3), then it retries the SMP ZONE UNLOCK request a vendor specific number of times, then originates an SMP ZONE ACTIVATE request to each locked zoning expander device.

If the active zone manager originated SMP ZONE ACTIVATE request(s), then after all the SMP ZONE ACTIVATE functions have completed without error, it sends an SMP ZONE UNLOCK request with the ACTIVATE REQUIRED bit set to zero to each of the locked zoning expander devices. If the active zone manager receives an SMP ZONE UNLOCK response with the FUNCTION RESULT field set to BUSY (see 9.4.3.3), then it retries the SMP ZONE UNLOCK request.

When the SMP ZONE UNLOCK request is successful or the zone lock inactivity timer expires, then the zoning expander device is unlocked and shall:

- a) set the ZONE LOCKED bit to zero in the SMP REPORT GENERAL response (see 9.4.3.4);
- b) set the ZONE CONFIGURING bit to zero in the SMP REPORT GENERAL response;
- c) if the zone lock timer expired, originate a Broadcast (Change) from zone group 1; and
- d) if the management device server processed an SMP ZONE UNLOCK request, originate a Broadcast (Change) (see 6.12) from either:
 - A) each zone group whose zone permission table entries or zone phy information has changed; or
 - B) zone group 1.

When all SMP ZONE UNLOCK requests are successful, the zone configuration process is complete.

4.9.6.6 Zone lock inactivity timer

The zone lock inactivity timer ensures that if the zone manager disappears without performing the unlock step that all locked zoning expander devices are unlocked.

When a zoning expander device processes an SMP ZONE LOCK request (see 9.4.3.21) then the zone lock inactivity timer default value is set to the value of the ZONE LOCK INACTIVITY TIME LIMIT field.

The zone lock inactivity timer is initialized and started if the default value is non-zero and:

- a) the zoning expander device completes processing of any SMP zone configuration function request or SMP ZONE ACTIVATE request while the ZONE LOCKED bit is set to one in the SMP REPORT GENERAL response (see 9.4.3.4); or
- b) the zoning expander device completes processing of a successful SMP ZONE LOCK request.

The zone lock inactivity timer is stopped if:

- a) the ZONE LOCK INACTIVITY TIME LIMIT field is set to zero in an SMP ZONE LOCK request; or
- b) the ZONE LOCKED bit is set to zero in the SMP REPORT GENERAL response (e.g., an SMP ZONE UNLOCK request (see 9.4.3.23) is processed, or the zone lock inactivity timer expires).

If the zone lock inactivity timer expires then the zoning expander device:

- a) sets the ZONE LOCKED bit to zero in the SMP REPORT GENERAL response;
- b) sets the ZONE CONFIGURING bit to zero in the SMP REPORT GENERAL response; and
- c) sends Broadcast (Change) on all ports.

If the zone lock inactivity timer expires while the zoning expander device is processing an SMP configuration function then the zoning expander device may complete the request without error or return a function result of ZONE LOCK VIOLATION.

4.9.6.7 Enable a zoning expander device

If a zoning expander device has the ZONING SUPPORTED bit set to one and the ZONING ENABLED bit set to zero in the REPORT GENERAL response (see 9.4.3.4), then a zone manager configures the zoning expander device using the zone configuration process. This ensures that the zone permission table is the same in all zoning expander devices inside the ZPSDS.

Changes made by the SMP ENABLE DISABLE ZONING function sent by the active zone manager become active during the activate step (see 4.9.6.4).

4.10 Phy power conditions

4.10.1 Low phy power conditions

4.10.1.1 Low phy power conditions overview

Low phy power conditions are phy conditions where the phy is in a reduced power state (e.g., has disabled circuitry in order to reduce power). This standard defines low phy power conditions that are differentiated by time to return to the active phy power condition (see 4.10.1.2 and table 44) and the amount of power consumed in that low phy power condition. The low phy power conditions include the partial phy power condition (see 4.10.1.3) and the slumber phy power condition (see 4.10.1.4).

A phy in a low phy power condition shall not change to a different low phy power condition without first making a change to the active phy power condition (see 5.8.2).

Low phy power conditions shall only be enabled on a phy if multiplexing (see 5.11) and optical mode are disabled.

If the partial phy power condition is enabled and the received IDENTIFY address frame has the PARTIAL CAPABLE bit set to one (see 6.8.2), then the phy may generate a PS_REQ (PARTIAL) primitive sequence. If the slumber phy power condition is enabled and the received IDENTIFY address frame has the SLUMBER

CAPABLE bit set to one (see 6.8.2), then the phy may generate a PS_REQ (SLUMBER) primitive sequence. If low phy power conditions are enabled, then the phy may reply with a PS_ACK primitive sequence to accept a low phy power condition request. If low phy power conditions are supported and disabled, then the phy shall reject a low phy power condition request by replying with a PS_NAK primitive sequence.

If a SAS phy or expander phy is in a low phy power condition and that phy is requested to transmit a NOTIFY, then that phy shall not transmit the NOTIFY and shall remain in the same low phy power condition.

4.10.1.2 Active phy power condition

While in the active phy power condition:

- a) the phy is capable of transmitting information and responding to received information without changing that phy's power condition; and
- b) the phy may consume more power than while the phy is in a low phy power condition.

4.10.1.3 Partial phy power condition

While in the partial phy power condition:

- a) the phy is only capable of processing a COMINIT or COMWAKE;
- b) the phy may take less time to return to the active phy power condition (see table 44) than while in the slumber phy power condition; and
- c) the power consumed by the phy should be less than the power consumed while the phy is in the active phy power condition and may be greater than the power consumed while the phy is in the slumber phy power condition.

4.10.1.4 Slumber phy power condition

While in the slumber phy power condition:

- a) the phy is only capable of processing a COMINIT or COMWAKE;
- b) the phy may take more time to return to the active phy power condition (see table 44) than when in the partial phy power condition; and
- c) the power consumed by the phy should be less than the power consumed while the phy is in the active phy power condition or when the phy is in the partial phy power condition.

4.10.1.5 End device low phy power conditions

Low phy power conditions are reported in SAS target devices using the Enhanced Phy Control mode page (see 9.2.7.7).

Partial phy power condition may be enabled and disabled in SAS target devices using the Enhanced Phy Control mode page (see 9.2.7.7).

Slumber phy power condition may be enabled and disabled in SAS target devices using the Enhanced Phy Control mode page (see 9.2.7.7).

The management application layer shall only enable low phy power conditions (i.e., send a Manage Power Conditions (Accept Partial) request or Manage Power Conditions (Accept Slumber) request) and may request phys enter a low phy condition (i.e., send a Change Phy Power Condition request) after receiving a Phy Power Condition Status (Enable Low Phy Power Conditions) message from the SA_PC state machine (see 9.2.10.2).

The management application layer shall disable low phy power conditions (i.e., send a Manage Power Conditions (Reject Partial) request and Manage Power Conditions (Reject Slumber) request) and shall not request phys enter a low phy power condition (i.e., send a Change Phy Power Condition request) after receiving a Phy Power Condition Status (Disable Low Phy Power Conditions) message from the SA_PC state machine (see 9.2.10.2).

If a SAS phy is in a low phy power condition, then to originate a Broadcast the application layer:

- 1) shall initiate the exit power condition procedure (see 5.8.2) on that SAS phy;

- 2) shall originate the Broadcast; and
- 3) may initiate the procedure to return that SAS phy to a low phy power condition.

4.10.1.6 Expander low phy power conditions

Low phy power conditions are reported in expander devices using the SMP DISCOVER function (see 9.4.3.10).

Partial phy power conditions may be enabled and disabled in SAS expander devices using the SMP PHY CONTROL function (see 9.4.3.28).

If an expander phy is in the partial phy power condition and the ECM receives a connection request routed to that expander phy, then the expander device initiates the exit power condition procedure (see 5.8.2) on that expander phy and responds with AIP (NORMAL) until the OPEN address frame is forwarded to that expander phy.

Slumber phy power conditions may be enabled and disabled in SAS expander devices using the SMP PHY CONTROL function (see 9.4.3.28).

If an expander phy is in slumber phy power condition and the ECM receives a connection request routed to that expander phy, then the expander device initiates the exit power condition procedure (see 5.8.2) on that expander phy and responds with OPEN_REJECT (RETRY) until a phy ready state (see 5.9.4.10) is established for that expander phy.

If an expander phy is in a low phy power condition, then to originate or forward a Broadcast the BPP:

- 1) shall initiate the exit power condition procedure (see 5.8.2) on that expander phy;
- 2) shall originate or forward the Broadcast; and
- 3) may initiate the procedure to return that expander phy to a low phy power condition.

4.10.2 SATA phy power conditions

STP initiator ports shall not generate SATA_PMREQ_P, SATA_PMREQ_S, or SATA_PMACK. If an STP initiator port receives SATA_PMREQ_P or SATA_PMREQ_S, then the STP initiator port shall reply with:

- a) SATA_X_RDY if the STP initiator port has a FIS ready to transmit to the STP target port; or
- b) a CLOSE primitive sequence.

NOTE 15 SAS-2 required the STP initiator port to transmit SATA_PMNAK.

SATA interface power management sequences (see SATA) may be enabled in an expander phy using the SMP PHY CONTROL function (see 9.4.3.28).

If an expander device receives SATA_PMREQ_P or SATA_PMREQ_S from a SATA device while an STP connection is not open, then it shall not forward the primitive to any STP initiator port and shall reply with SATA_PMNAK or SATA_PMACK as defined by SATA. If SATA interface power management sequences are not enabled, then the expander device shall reply with SATA_PMNAK.

If an expander device receives SATA_PMREQ_P or SATA_PMREQ_S while an STP connection is open, then the expander device may or may not forward the primitive to the STP initiator port. If the expander device forwards a SATA_PMREQ_P or SATA_PMREQ_S to the STP initiator port during an STP connection, then the expander device shall not reply with SATA_PMACK or SATA_PMNAK within that connection.

4.11 SAS device and expander device power conditions

SCSI idle and standby power conditions, implemented with the START STOP UNIT command (see SBC-3) and the Power Condition mode page (see SPC-4), may be supported by SSP initiator ports and SSP target ports as described in 9.2.10.

Except for Sleep mode (e.g., requested with the ATA SLEEP command), the ATA Power Management feature set, Extended Power Management feature set, or Advanced Power Management feature set (see ACS-2) may be supported by an ATA application client using an STP initiator port.

4.12 Phy test functions

4.12.1 Phy test functions overview

Phy test functions (e.g., transmission of test patterns) are used for phy and interconnect characterization and diagnosis. The phy may be attached to test equipment while performing a phy test function. The following optional mechanisms are defined for invoking phy test functions:

- a) the Protocol-Specific diagnostic page for SAS (see 9.2.9.2) invokes a phy test function in a selected phy in a SAS target device with an SSP target port. The SEND DIAGNOSTIC command (see SPC-4) may be sent through any SSP target port to any logical unit in the SAS target device that contains the phy that is to perform the phy test function. The phy test function starts some time after the SSP target port receives an ACK for the RESPONSE frame transmitted in response to the SEND DIAGNOSTIC command; and
- b) the SMP PHY TEST FUNCTION function (see 9.4.3.29) invokes a phy test function in a phy controlled by a management device server other than the phy that receives the function. The phy test function starts some time after the SMP target port transmits the SMP response frame.

Each phy test function is optional.

If the phy test function requires a specific phy test pattern and/or phy test function physical link rate, then the mechanism for invoking the phy test function also specifies the phy test pattern and phy test function physical link rate.

The phy test function on one phy may affect the previously negotiated settings on other phys (e.g., in a device with a common SSC clock, the SSC modulation type may change from none to down-spreading even on phys that negotiated no SSC).

While a phy is performing a phy test function, the link layer receivers (i.e., the SL_IR receiver, SL receiver, SSP receiver, STP receiver, and SMP receiver) shall ignore all incoming dwords and the OOB signal detector shall detect COMINIT. The phy shall ignore any other OOB signals (i.e., COMSAS and COMWAKE).

A phy stops performing a phy test function:

- a) after the SCSI device server, if any, processes a Protocol-Specific diagnostic page specifying the phy and specifying a phy test function of 00h (i.e., STOP);
- b) after the management device server, if any, processes an SMP PHY TEST FUNCTION request specifying the phy and specifying a phy test function of 00h (i.e., STOP);
- c) after the phy receives COMINIT; or
- d) upon power off.

It is vendor specific how long a phy takes to stop performing the phy test function. After a phy stops performing a phy test function, the phy performs a link reset sequence.

4.12.2 Transmit pattern phy test function

While a phy is performing the transmit pattern phy test function, the test equipment attached to that phy:

- a) shall not transmit COMSAS or COMWAKE; and
- b) shall not transmit COMINIT except to stop the phy test function.

While performing the transmit pattern phy test function, a phy:

- a) shall ignore all dwords received; and
- b) shall repeatedly transmit the specified pattern at the specified physical link rate.

4.13 Phy events

Phys shall count the following events using saturating counters and report them in the Protocol-Specific Port log page (see 9.2.8.1) and/or the SMP REPORT PHY ERROR LOG function (see 9.4.3.11):

- a) invalid dwords received;
- b) dwords received with running disparity errors;
- c) loss of dword synchronization; and
- d) phy reset problems.

The saturating counters are each up to 4 bytes wide.

Phys may count those events and certain other events (e.g., elasticity buffer overflows) using wrapping counters and record peak values for certain events (e.g., the longest connection time) using peak value detectors, reporting them in the Protocol-Specific Port log page (see 9.2.8.1), SMP REPORT PHY EVENT function (see 9.4.3.14), and/or the SMP REPORT PHY EVENT LIST function (see 9.4.3.16). The wrapping counters and peak value detectors are each 4 bytes wide. Peak value detectors trigger Broadcast (Expander) under certain circumstances (see 6.2.6.4).

For phys not controlled by SMP target ports, the number of additional events monitored and which events are monitored is vendor specific.

For phys controlled by SMP target ports, the number of additional events that are simultaneously monitored is vendor specific, but the SMP CONFIGURE PHY EVENT function (see 9.4.3.30) allows for specification of the events to monitor.

The management device server shall maintain phy events for the last vendor specific number of events and should maintain at least one phy event per phy. The management device server shall assign descriptors to the events sequentially starting at 0001h and shall return the descriptors in the SMP REPORT PHY EVENT LIST response (see 9.4.3.16). The management device server shall return the index of the descriptor for the last phy event in the SMP REPORT GENERAL response (see 9.4.3.4), the SMP REPORT PHY EVENT LIST response (see 9.4.3.16), and the SMP DISCOVER LIST response (see 9.4.3.15). The management device server shall wrap the index to 0001h when the highest supported descriptor index has been used.

The management device server shall support phy event list descriptor (see 9.4.3.16.4) indexes from 0001h to FFFFh. The actual number of phy event list descriptors that the management device server maintains for retrieval with the REPORT PHY EVENT LIST request is vendor specific and is indicated by the MAXIMUM NUMBER OF STORED PHY EVENT LIST DESCRIPTORS field defined in the REPORT GENERAL response (see 9.4.3.4). The volatility of these stored descriptors is vendor specific. The management device server shall replace the oldest phy event list descriptor with a new one once the number of recorded descriptors exceeds the value indicated by the MAXIMUM NUMBER OF STORED PHY EVENT LIST DESCRIPTORS field.

The PHY EVENT SOURCE field, defined in table 37, is used in the Protocol-Specific Port log page (see 9.2.8.1), the REPORT PHY EVENT function (see 9.4.3.14), the REPORT PHY EVENT LIST function (see 9.4.3.16), and the CONFIGURE PHY EVENT function (see 9.4.3.30) and indicates or specifies the type of phy event in the accompanying PHY EVENT field.

Table 37 — PHY EVENT SOURCE field (part 1 of 4)

Code	Name	Type ^a	Description
00h	No event	N/A	No event. The PHY EVENT field is not valid.
Phy layer-based phy events (01h to 1Fh)			
01h	Invalid dword count ^b	WC	Number of invalid dwords (see 3.1.94) that have been received outside of phy reset sequences (i.e., between when the SP state machine (see 5.9) sends a Phy Layer Ready (SAS) confirmation or Phy Layer Ready (SATA) confirmation and when it sends a Phy Layer Not Ready confirmation to the link layer).
02h	Running disparity error count ^b	WC	Number of dwords containing running disparity errors (see 5.3.5) that have been received outside of phy reset sequences.
03h	Loss of dword synchronization count ^b	WC	Number of times the phy has restarted the link reset sequence because it lost dword synchronization (i.e., the SP state machine transitioned from SP15:SAS_PHY_Ready or SP22:SATA_PHY_Ready to SP0:OOB_COMINIT (see 5.9)).
04h	Phy reset problem count ^b	WC	Number of times a phy reset problem has occurred (see 5.7.4.2.4).
05h	Elasticity buffer overflow count	WC	Number of times the phy's elasticity buffer (see 6.3) has overflowed outside of phy reset sequences (e.g., because it did not receive a sufficient number of deletable primitives).
06h	Received ERROR count	WC	Number of times the phy received an ERROR.
07h to 1Fh	Reserved for phy layer-based phy events		
SAS arbitration-related phy events (20h to 3Fh)			
20h	Received address frame error count	WC	Number of times the phy detected an invalid address frame (see 6.8) (e.g., because of a CRC error).
21h	Transmitted abandon-class OPEN_REJECT count	WC	Number of times the phy received an OPEN address frame and transmitted an abandon-class OPEN_REJECT (see 6.2.6.10). In expander devices, forwarded OPEN_REJECTs shall not be counted.
22h	Received abandon-class OPEN_REJECT count	WC	Number of times the phy originated an OPEN address frame and received an abandon-class OPEN_REJECT (see 6.2.6.10). In expander devices, OPEN_REJECTs in response to forwarded OPEN address frames shall not be counted.

^a The Type column indicates the source type:

- a) WC = wrapping counter;
- b) PVD = peak value detector; and
- c) N/A = not applicable.

^b This standard also defines a saturating counter that counts this event (see 9.2.8.1 and 9.4.3.11).

Table 37 — PHY EVENT SOURCE field (part 2 of 4)

Code	Name	Type ^a	Description
23h	Transmitted retry-class OPEN_REJECT count	WC	Number of times the phy received an OPEN address frame and transmitted a retry-class OPEN_REJECT (see 6.2.6.10). In expander devices, forwarded OPEN_REJECTs shall not be counted.
24h	Received retry-class OPEN_REJECT count	WC	Number of times the phy originated an OPEN address frame and received a retry-class OPEN_REJECT (see 6.2.6.10). In expander devices, OPEN_REJECTs in response to forwarded OPEN address frames shall not be counted.
25h	Received AIP (WAITING ON PARTIAL) count	WC	Number of times the phy received an AIP (WAITING ON PARTIAL) or AIP (RESERVED WAITING ON PARTIAL). In expander devices, forwarded AIPs shall be counted.
26h	Received AIP (WAITING ON CONNECTION) count	WC	Number of times the phy received an AIP (WAITING ON CONNECTION). In expander devices, forwarded AIPs shall be counted.
27h	Transmitted BREAK count	WC	Number of times the phy transmitted a BREAK primitive sequence that was not a response to a BREAK primitive sequence it received (e.g., a Close Timeout was detected by the SL state machines interfacing to the SMP target port).
28h	Received BREAK count	WC	Number of times the phy received a BREAK primitive sequence that was not a response to a BREAK primitive sequence that it transmitted.
29h	Break Timeout count	WC	Number of times the phy transmitted a BREAK primitive sequence and did not receive a BREAK primitive sequence or BREAK_REPLY primitive sequence in response (e.g., as detected by the XL state machine and/or the SL state machines interfacing to the SMP target port).
2Ah	Connection count	WC	Number of connections in which the phy was involved.
2Bh	Peak transmitted pathway blocked count	PVD	Peak value of a PATHWAY BLOCKED COUNT field in an OPEN address frame transmitted by the phy. Since the maximum value of the PATHWAY BLOCKED COUNT field is FFh, only byte 3 of the PHY EVENT field is used.
2Ch	Peak transmitted arbitration wait time	PVD	Peak value of an ARBITRATION WAIT TIME field in an OPEN address frame transmitted by the phy. Since the maximum value of the ARBITRATION WAIT TIME field is FFFFh, only byte 2 and byte 3 of the PHY EVENT field are used.
2Dh	Peak arbitration time	PVD	Peak time in microseconds after transmitting an OPEN address frame that the phy has waited for connection response (e.g., OPEN_ACCEPT or OPEN_REJECT).
2Eh	Peak connection time	PVD	The peak duration, in microseconds, of any connection in which the phy was involved.

^a The Type column indicates the source type:

- a) WC = wrapping counter;
- b) PVD = peak value detector; and
- c) N/A = not applicable.

^b This standard also defines a saturating counter that counts this event (see 9.2.8.1 and 9.4.3.11).

Table 37 — PHY EVENT SOURCE field (part 3 of 4)

Code	Name	Type ^a	Description
2Fh to 3Fh	Reserved for SAS arbitration-related phy information		
SSP-related phy events (40h to 4Fh)			
40h	Transmitted SSP frame count	WC	Number of SSP frames transmitted.
41h	Received SSP frame count	WC	Number of SSP frames received.
42h	Transmitted SSP frame error count	WC	Number of times the phy was used in a connection involving the SSP target port, transmitted an SSP frame, and received a NAK or an ACK/NAK timeout.
43h	Received SSP frame error count	WC	Number of times the phy was used in a connection involving the SSP target port, detected an invalid SSP frame, and transmitted a NAK (CRC ERROR) (e.g., because of a CRC error).
44h	Transmitted CREDIT_BLOCKED count	WC	Number of times the phy transmitted a CREDIT_BLOCKED.
45h	Received CREDIT_BLOCKED count	WC	Number of times the phy received a CREDIT_BLOCKED.
46h to 4Fh	Reserved for SSP-related phy events		
STP and SATA-related phy events (50h to 5Fh)			
50h	Transmitted SATA frame count	WC	Number of STP or SATA frames transmitted.
51h	Received SATA frame count	WC	Number of STP or SATA frames received.
52h	SATA flow control buffer overflow count	WC	Number of times the phy's STP flow control buffer (see 6.18.2) has overflowed (e.g., because it received more data dwords than allowed after transmitting SATA_HOLD during an STP connection). In an expander device, this count should be maintained in the expander phy transmitting the SATA_HOLD and receiving the data dwords, but may be maintained in the expander phy receiving the SATA_HOLD and transmitting the data dwords.
53h to 5Fh	Reserved for STP and SATA-related phy events		
^a The Type column indicates the source type: a) WC = wrapping counter; b) PVD = peak value detector; and c) N/A = not applicable. ^b This standard also defines a saturating counter that counts this event (see 9.2.8.1 and 9.4.3.11).			

Table 37 — PHY EVENT SOURCE field (part 4 of 4)

Code	Name	Type ^a	Description
SMP-related phy events (60h to 6Fh)			
60h	Transmitted SMP frame count	WC	Number of SMP frames transmitted.
61h	Received SMP frame count	WC	Number of SMP frames received.
62h	Reserved for SMP-related phy events		
63h	Received SMP frame error count	WC	Number of times the phy was used to access the SMP target port and the SMP target port detected an invalid SMP frame and transmitted a BREAK primitive sequence (e.g., because of a CRC error).
64h to 6Fh	Reserved for SMP-related phy events		
Other (70h to FFh)			
70h to CFh	Reserved		
D0h to FFh	Vendor specific		
^a The Type column indicates the source type: a) WC = wrapping counter; b) PVD = peak value detector; and c) N/A = not applicable. ^b This standard also defines a saturating counter that counts this event (see 9.2.8.1 and 9.4.3.11).			

5 Phy layer

5.1 Phy layer overview

The phy layer defines 8b10b coding and OOB signals. Phy layer state machines interface between the link layer and the physical layer to perform the phy reset sequence and keep track of dword synchronization.

5.2 8b10b coding

5.2.1 8b10b coding overview

All information transferred in SAS is encoded into 10-bit characters using 8b10b encoding. 10-bit characters are transmitted serially bit-by-bit across the physical link.

Out of all 1024 possible 10-bit characters:

- a) some of the characters are data characters, representing the 256 possible 8-bit data bytes;
- b) some of the characters are control characters, used for primitives (e.g., frame delimiters) and other control purposes; and
- c) the rest of the characters are invalid characters.

8b10b coding ensures that sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. 8b10b coding also increases the likelihood of detecting any single or multiple bit errors that occur during transmission and reception. In addition, some of the control characters of the transmission code contain a distinct bit pattern, called a comma pattern, which assists a receiver in achieving character and dword alignment on the incoming bit stream.

5.2.2 8b10b coding notation conventions

This subclause uses letter notation for describing information bits and control variables. Such notation differs from the bit notation specified by the remainder of this standard. The following text describes the translation process between these notations and provides a translation example. This subclause also describes the conventions used to name valid characters. This text is provided for the purposes of terminology clarification only.

An unencoded information byte is composed of:

- a) eight information bits labeled A, B, C, D, E, F, G, and H. Each information bit contains either a binary zero or a binary one; and
- b) a control variable labeled Z. A control variable has either the value D or the value K:
 - A) D means the information byte is a data byte; and
 - B) K means the information byte is a control byte.

The information bit labeled A corresponds to bit 0 in the numbering scheme of this standard, B corresponds to bit 1, and so on, as shown in table 38. Bit H is the most significant bit of the byte and bit A is the least significant bit of the byte.

Table 38 — Bit designations

Bit notation:	7	6	5	4	3	2	1	0	Control variable
Unencoded bit notation:	H	G	F	E	D	C	B	A	Z

Each valid character is named using the following convention:

Zxx.y

where:

- Z is the control variable of the unencoded information byte. The value of Z is used to indicate whether the character is a data character (i.e., Z = D) or a control character (i.e., Z = K).
- xx is the decimal value of the binary number composed of the bits E, D, C, B, and A of the unencoded information byte in that order.
- y is the decimal value of the binary number composed of the bits H, G, and F of the unencoded information byte in that order.

Table 39 shows the conversion from byte notation to the character naming convention.

Table 39 — Conversion from byte notation to character name example

Byte notation	BCh									
Bit notation	7	6	5	4	3	2	1	0	Control	
	1	0	1	1	1	1	0	0	K	
Unencoded bit notation	H	G	F	E	D	C	B	A	Z	
	1	0	1	1	1	1	0	0	K	
Unencoded bit notation reordered to conform with Zxx.y naming convention	Z	E	D	C	B	A	H	G	F	
	K	1	1	1	0	0	1	0	1	
Character name	K 28 . 5									

Most Kxx.y combinations do not result in valid characters within the 8b10b coding scheme. Only those combinations that result in control characters defined in table 41 (see 5.3.7) are considered valid.

5.3 Character encoding and decoding

5.3.1 Introduction

This subclause describes how to select valid characters (i.e., 8b10b encoding) and check the validity of received characters (i.e., 10b8b decoding), and specifies the ordering rules to be followed when transmitting the bits within a character.

5.3.2 Bit transmission order

An information byte is encoded into a 10-bit character containing bits labeled a, b, c, d, e, i, f, g, h, and j. Bit a shall be transmitted first, followed by bits b, c, d, e, i, f, g, h, and j, in that order.

5.3.3 Character transmission order

Characters within primitives shall be transmitted sequentially beginning with the control character used to distinguish the primitive (e.g., K28.3 or K28.5) and proceeding character by character from left to right within the definition of the primitive until all characters of the primitive are transmitted.

5.3.4 Frame transmission order

The contents of a frame shall be transmitted sequentially beginning with the primitive used to denote the start of frame (e.g., SOAF, SOF, or SATA_SOF) and proceeding character-by-character from left to right within the definition of the frame until the primitive used to denote the end of frame (e.g., EOAF, EOF, or SATA_EOF) is transmitted.

5.3.5 Running disparity (RD)

RD is a binary parameter with a negative (-) or positive (+) value. After power on, the transmitter may initialize the current RD to either positive or negative.

Each data character and control character is defined in a table by two columns that represent two characters that may or may not be different, corresponding to the current value of the running disparity (i.e., current RD - or current RD +).

Upon transmission of any character, the transmitter shall calculate a new value for its RD based on the contents of the transmitted character.

After power on, the receiver shall assume either the positive or negative value for its initial RD. Upon reception of any character, the receiver shall determine whether the character is valid or invalid and shall calculate a new value for its RD based on the contents of the received character.

The following rules for RD shall be used to calculate the new RD value for characters that have been transmitted (i.e., the transmitter's RD) and that have been received (i.e., the receiver's RD).

RD for a character shall be calculated on the basis of sub-blocks, where the first six bits (i.e., bits a, b, c, d, e, and i) form one sub-block (i.e., the six-bit sub-block) and the second four bits (i.e., bits f, g, h, and j) form the other sub-block (i.e., the four-bit sub-block). RD has the following properties:

- a) RD at the beginning of the six-bit sub-block is the RD at the end of the preceding character;
- b) RD at the beginning of the four-bit sub-block is the RD at the end of the preceding six-bit sub-block; and
- c) RD at the end of the character is the RD at the end of the four-bit sub-block.

RD for the sub-blocks shall be calculated as follows:

- a) if the sub-block contains more ones than zeros, then RD at the end of a sub-block is positive;
- b) if the sub-block contains more zeros than ones, then RD at the end of a sub-block is negative; or
- c) if the sub-block contains equal numbers of zeros and ones, then:
 - A) if it is a six-bit sub-block containing 000111b, then RD at the end of the sub-block is positive;
 - B) if it is a six-bit sub-block containing 111000b, then RD at the end of the sub-block is negative;
 - C) if it is a four-bit sub-block containing 0011b, then RD at the end of the sub-block is positive;
 - D) if it is a four-bit sub-block containing 1100b, then RD at the end of the sub-block is negative; or
 - E) otherwise, RD at the end of the sub-block is the same as at the beginning of the sub-block.

All sub-blocks with equal numbers of zeros and ones have neutral disparity (i.e., the ending disparity is the same as the beginning disparity). In order to limit the run length of zeros or ones across adjacent sub-blocks, the 8b10b code rules specify that sub-blocks encoded as 000111b or 0011b are generated only when the RD at the beginning of the sub-block is positive, ensuring that RD at the end of these sub-blocks is also positive. Likewise, sub-blocks containing 111000b or 1100b are generated only when the RD at the beginning of the sub-block is negative, ensuring that RD at the end of these sub-blocks is also negative.

Running disparity (RD) shall be maintained separately on each physical link in each direction. During a connection (see 4.1.12), expander devices shall convert incoming 10-bit characters to 8-bit bytes and generate the 10-bit character with correct disparity for the output physical link. Phys within a device may or may not begin operation with the same disparity.

5.3.6 Data characters

Table 40 defines the data characters (i.e., Dxx.y characters), and shall be used for both generating characters (i.e., encoding) and checking the validity of received characters (i.e., decoding).

Table 40 defines the data characters.

Table 40 — Data characters (part 1 of 5)

Name	Data byte		Data character (binary representation)	
	Binary representation (HGF EDCBA)	Hexadecimal representation	Current RD - abcdei fghj	Current RD + abcdei fghj
D00.0	000 00000	00h	100111 0100	011000 1011
D01.0	000 00001	01h	011101 0100	100010 1011
D02.0	000 00010	02h	101101 0100	010010 1011
D03.0	000 00011	03h	110001 1011	110001 0100
D04.0	000 00100	04h	110101 0100	001010 1011
D05.0	000 00101	05h	101001 1011	101001 0100
D06.0	000 00110	06h	011001 1011	011001 0100
D07.0	000 00111	07h	111000 1011	000111 0100
D08.0	000 01000	08h	111001 0100	000110 1011
D09.0	000 01001	09h	100101 1011	100101 0100
D10.0	000 01010	0Ah	010101 1011	010101 0100
D11.0	000 01011	0Bh	110100 1011	110100 0100
D12.0	000 01100	0Ch	001101 1011	001101 0100
D13.0	000 01101	0Dh	101100 1011	101100 0100
D14.0	000 01110	0Eh	011100 1011	011100 0100
D15.0	000 01111	0Fh	010111 0100	101000 1011
D16.0	000 10000	10h	011011 0100	100100 1011
D17.0	000 10001	11h	100011 1011	100011 0100
D18.0	000 10010	12h	010011 1011	010011 0100
D19.0	000 10011	13h	110010 1011	110010 0100
D20.0	000 10100	14h	001011 1011	001011 0100
D21.0	000 10101	15h	101010 1011	101010 0100
D22.0	000 10110	16h	011010 1011	011010 0100
D23.0	000 10111	17h	111010 0100	000101 1011
D24.0	000 11000	18h	110011 0100	001100 1011
D25.0	000 11001	19h	100110 1011	100110 0100
D26.0	000 11010	1Ah	010110 1011	010110 0100
D27.0	000 11011	1Bh	110110 0100	001001 1011
D28.0	000 11100	1Ch	001110 1011	001110 0100
D29.0	000 11101	1Dh	101110 0100	010001 1011
D30.0	000 11110	1Eh	011110 0100	100001 1011
D31.0	000 11111	1Fh	101011 0100	010100 1011
D00.1	001 00000	20h	100111 1001	011000 1001
D01.1	001 00001	21h	011101 1001	100010 1001
D02.1	001 00010	22h	101101 1001	010010 1001
D03.1	001 00011	23h	110001 1001	110001 1001
D04.1	001 00100	24h	110101 1001	001010 1001
D05.1	001 00101	25h	101001 1001	101001 1001
D06.1	001 00110	26h	011001 1001	011001 1001
D07.1	001 00111	27h	111000 1001	000111 1001
D08.1	001 01000	28h	111001 1001	000110 1001
D09.1	001 01001	29h	100101 1001	100101 1001
D10.1	001 01010	2Ah	010101 1001	010101 1001
D11.1	001 01011	2Bh	110100 1001	110100 1001
D12.1	001 01100	2Ch	001101 1001	001101 1001
D13.1	001 01101	2Dh	101100 1001	101100 1001
D14.1	001 01110	2Eh	011100 1001	011100 1001
D15.1	001 01111	2Fh	010111 1001	101000 1001
D16.1	001 10000	30h	011011 1001	100100 1001
D17.1	001 10001	31h	100011 1001	100011 1001
D18.1	001 10010	32h	010011 1001	010011 1001
D19.1	001 10011	33h	110010 1001	110010 1001
D20.1	001 10100	34h	001011 1001	001011 1001

Table 40 — Data characters (part 2 of 5)

Name	Data byte		Data character (binary representation)	
	Binary representation (HGF EDCBA)	Hexadecimal representation	Current RD - abcdei fghj	Current RD + abcdei fghj
D21.1	001 10101	35h	101010 1001	101010 1001
D22.1	001 10110	36h	011010 1001	011010 1001
D23.1	001 10111	37h	111010 1001	000101 1001
D24.1	001 11000	38h	110011 1001	001100 1001
D25.1	001 11001	39h	100110 1001	100110 1001
D26.1	001 11010	3Ah	010110 1001	010110 1001
D27.1	001 11011	3Bh	110110 1001	001001 1001
D28.1	001 11100	3Ch	001110 1001	001110 1001
D29.1	001 11101	3Dh	101110 1001	010001 1001
D30.1	001 11110	3Eh	011110 1001	100001 1001
D31.1	001 11111	3Fh	101011 1001	010100 1001
D00.2	010 00000	40h	100111 0101	011000 0101
D01.2	010 00001	41h	011101 0101	100010 0101
D02.2	010 00010	42h	101101 0101	010010 0101
D03.2	010 00011	43h	110001 0101	110001 0101
D04.2	010 00100	44h	110101 0101	001010 0101
D05.2	010 00101	45h	101001 0101	101001 0101
D06.2	010 00110	46h	011001 0101	011001 0101
D07.2	010 00111	47h	111000 0101	000111 0101
D08.2	010 01000	48h	111001 0101	000110 0101
D09.2	010 01001	49h	100101 0101	100101 0101
D10.2	010 01010	4Ah	010101 0101	010101 0101
D11.2	010 01011	4Bh	110100 0101	110100 0101
D12.2	010 01100	4Ch	001101 0101	001101 0101
D13.2	010 01101	4Dh	101100 0101	101100 0101
D14.2	010 01110	4Eh	011100 0101	011100 0101
D15.2	010 01111	4Fh	010111 0101	101000 0101
D16.2	010 10000	50h	011011 0101	100100 0101
D17.2	010 10001	51h	100011 0101	100011 0101
D18.2	010 10010	52h	010011 0101	010011 0101
D19.2	010 10011	53h	110010 0101	110010 0101
D20.2	010 10100	54h	001011 0101	001011 0101
D21.2	010 10101	55h	101010 0101	101010 0101
D22.2	010 10110	56h	011010 0101	011010 0101
D23.2	010 10111	57h	111010 0101	000101 0101
D24.2	010 11000	58h	110011 0101	001100 0101
D25.2	010 11001	59h	100110 0101	100110 0101
D26.2	010 11010	5Ah	010110 0101	010110 0101
D27.2	010 11011	5Bh	110110 0101	001001 0101
D28.2	010 11100	5Ch	001110 0101	001110 0101
D29.2	010 11101	5Dh	101110 0101	010001 0101
D30.2	010 11110	5Eh	011110 0101	100001 0101
D31.2	010 11111	5Fh	101011 0101	010100 0101
D00.3	011 00000	60h	100111 0011	011000 1100
D01.3	011 00001	61h	011101 0011	100010 1100
D02.3	011 00010	62h	101101 0011	010010 1100
D03.3	011 00011	63h	110001 1100	110001 0011
D04.3	011 00100	64h	110101 0011	001010 1100
D05.3	011 00101	65h	101001 1100	101001 0011
D06.3	011 00110	66h	011001 1100	011001 0011
D07.3	011 00111	67h	111000 1100	000111 0011
D08.3	011 01000	68h	111001 0011	000110 1100
D09.3	011 01001	69h	100101 1100	100101 0011
D10.3	011 01010	6Ah	010101 1100	010101 0011
D11.3	011 01011	6Bh	110100 1100	110100 0011

Table 40 — Data characters (part 3 of 5)

Name	Data byte		Data character (binary representation)	
	Binary representation (HGF EDCBA)	Hexadecimal representation	Current RD - abcdei fghj	Current RD + abcdei fghj
D12.3	011 01100	6Ch	001101 1100	001101 0011
D13.3	011 01101	6Dh	101100 1100	101100 0011
D14.3	011 01110	6Eh	011100 1100	011100 0011
D15.3	011 01111	6Fh	010111 0011	101000 1100
D16.3	011 10000	70h	011011 0011	100100 1100
D17.3	011 10001	71h	100011 1100	100011 0011
D18.3	011 10010	72h	010011 1100	010011 0011
D19.3	011 10011	73h	110010 1100	110010 0011
D20.3	011 10100	74h	001011 1100	001011 0011
D21.3	011 10101	75h	101010 1100	101010 0011
D22.3	011 10110	76h	011010 1100	011010 0011
D23.3	011 10111	77h	111010 0011	000101 1100
D24.3	011 11000	78h	110011 0011	001100 1100
D25.3	011 11001	79h	100110 1100	100110 0011
D26.3	011 11010	7Ah	010110 1100	010110 0011
D27.3	011 11011	7Bh	110110 0011	001001 1100
D28.3	011 11100	7Ch	001110 1100	001110 0011
D29.3	011 11101	7Dh	101110 0011	010001 1100
D30.3	011 11110	7Eh	011110 0011	100001 1100
D31.3	011 11111	7Fh	101011 0011	010100 1100
D00.4	100 00000	80h	100111 0010	011000 1101
D01.4	100 00001	81h	011101 0010	100010 1101
D02.4	100 00010	82h	101101 0010	010010 1101
D03.4	100 00011	83h	110001 1101	110001 0010
D04.4	100 00100	84h	110101 0010	001010 1101
D05.4	100 00101	85h	101001 1101	101001 0010
D06.4	100 00110	86h	011001 1101	011001 0010
D07.4	100 00111	87h	111000 1101	000111 0010
D08.4	100 01000	88h	111001 0010	000110 1101
D09.4	100 01001	89h	100101 1101	100101 0010
D10.4	100 01010	8Ah	010101 1101	010101 0010
D11.4	100 01011	8Bh	110100 1101	110100 0010
D12.4	100 01100	8Ch	001101 1101	001101 0010
D13.4	100 01101	8Dh	101100 1101	101100 0010
D14.4	100 01110	8Eh	011100 1101	011100 0010
D15.4	100 01111	8Fh	010111 0010	101000 1101
D16.4	100 10000	90h	011011 0010	100100 1101
D17.4	100 10001	91h	100011 1101	100011 0010
D18.4	100 10010	92h	010011 1101	010011 0010
D19.4	100 10011	93h	110010 1101	110010 0010
D20.4	100 10100	94h	001011 1101	001011 0010
D21.4	100 10101	95h	101010 1101	101010 0010
D22.4	100 10110	96h	011010 1101	011010 0010
D23.4	100 10111	97h	111010 0010	000101 1101
D24.4	100 11000	98h	110011 0010	001100 1101
D25.4	100 11001	99h	100110 1101	100110 0010
D26.4	100 11010	9Ah	010110 1101	010110 0010
D27.4	100 11011	9Bh	110110 0010	001001 1101
D28.4	100 11100	9Ch	001110 1101	001110 0010
D29.4	100 11101	9Dh	101110 0010	010001 1101
D30.4	100 11110	9Eh	011110 0010	100001 1101
D31.4	100 11111	9Fh	101011 0010	010100 1101
D00.5	101 00000	A0h	100111 1010	011000 1010
D01.5	101 00001	A1h	011101 1010	100010 1010
D02.5	101 00010	A2h	101101 1010	010010 1010

Table 40 — Data characters (part 4 of 5)

Name	Data byte		Data character (binary representation)	
	Binary representation (HGF EDCBA)	Hexadecimal representation	Current RD - abcdei fghj	Current RD + abcdei fghj
D03.5	101 00011	A3h	110001 1010	110001 1010
D04.5	101 00100	A4h	110101 1010	001010 1010
D05.5	101 00101	A5h	101001 1010	101001 1010
D06.5	101 00110	A6h	011001 1010	011001 1010
D07.5	101 00111	A7h	111000 1010	000111 1010
D08.5	101 01000	A8h	111001 1010	000110 1010
D09.5	101 01001	A9h	100101 1010	100101 1010
D10.5	101 01010	AAh	010101 1010	010101 1010
D11.5	101 01011	ABh	110100 1010	110100 1010
D12.5	101 01100	ACH	001101 1010	001101 1010
D13.5	101 01101	ADh	101100 1010	101100 1010
D14.5	101 01110	A Eh	011100 1010	011100 1010
D15.5	101 01111	AFh	010111 1010	101000 1010
D16.5	101 10000	B0h	011011 1010	100100 1010
D17.5	101 10001	B1h	100011 1010	100011 1010
D18.5	101 10010	B2h	010011 1010	010011 1010
D19.5	101 10011	B3h	110010 1010	110010 1010
D20.5	101 10100	B4h	001011 1010	001011 1010
D21.5	101 10101	B5h	101010 1010	101010 1010
D22.5	101 10110	B6h	011010 1010	011010 1010
D23.5	101 10111	B7h	111010 1010	000101 1010
D24.5	101 11000	B8h	110011 1010	001100 1010
D25.5	101 11001	B9h	100110 1010	100110 1010
D26.5	101 11010	BAh	010110 1010	010110 1010
D27.5	101 11011	BBh	110110 1010	001001 1010
D28.5	101 11100	BCh	001110 1010	001110 1010
D29.5	101 11101	BDh	101110 1010	010001 1010
D30.5	101 11110	BEh	011110 1010	100001 1010
D31.5	101 11111	BFh	101011 1010	010100 1010
D00.6	110 00000	C0h	100111 0110	011000 0110
D01.6	110 00001	C1h	011101 0110	100010 0110
D02.6	110 00010	C2h	101101 0110	010010 0110
D03.6	110 00011	C3h	110001 0110	110001 0110
D04.6	110 00100	C4h	110101 0110	001010 0110
D05.6	110 00101	C5h	101001 0110	101001 0110
D06.6	110 00110	C6h	011001 0110	011001 0110
D07.6	110 00111	C7h	111000 0110	000111 0110
D08.6	110 01000	C8h	111001 0110	000110 0110
D09.6	110 01001	C9h	100101 0110	100101 0110
D10.6	110 01010	CAh	010101 0110	010101 0110
D11.6	110 01011	CBh	110100 0110	110100 0110
D12.6	110 01100	CCh	001101 0110	001101 0110
D13.6	110 01101	CDh	101100 0110	101100 0110
D14.6	110 01110	C Eh	011100 0110	011100 0110
D15.6	110 01111	CFh	010111 0110	101000 0110
D16.6	110 10000	D0h	011011 0110	100100 0110
D17.6	110 10001	D1h	100011 0110	100011 0110
D18.6	110 10010	D2h	010011 0110	010011 0110
D19.6	110 10011	D3h	110010 0110	110010 0110
D20.6	110 10100	D4h	001011 0110	001011 0110
D21.6	110 10101	D5h	101010 0110	101010 0110
D22.6	110 10110	D6h	011010 0110	011010 0110
D23.6	110 10111	D7h	111010 0110	000101 0110
D24.6	110 11000	D8h	110011 0110	001100 0110
D25.6	110 11001	D9h	100110 0110	100110 0110

Table 40 — Data characters (part 5 of 5)

Name	Data byte		Data character (binary representation)	
	Binary representation (HGF EDCBA)	Hexadecimal representation	Current RD - abcdei fghj	Current RD + abcdei fghj
D26.6	110 11010	DAh	010110 0110	010110 0110
D27.6	110 11011	DBh	110110 0110	001001 0110
D28.6	110 11100	DCh	001110 0110	001110 0110
D29.6	110 11101	DDh	101110 0110	010001 0110
D30.6	110 11110	DEh	011110 0110	100001 0110
D31.6	110 11111	DFh	101011 0110	010100 0110
D00.7	111 00000	E0h	100111 0001	011000 1110
D01.7	111 00001	E1h	011101 0001	100010 1110
D02.7	111 00010	E2h	101101 0001	010010 1110
D03.7	111 00011	E3h	110001 1110	110001 0001
D04.7	111 00100	E4h	110101 0001	001010 1110
D05.7	111 00101	E5h	101001 1110	101001 0001
D06.7	111 00110	E6h	011001 1110	011001 0001
D07.7	111 00111	E7h	111000 1110	000111 0001
D08.7	111 01000	E8h	111001 0001	000110 1110
D09.7	111 01001	E9h	100101 1110	100101 0001
D10.7	111 01010	EAh	010101 1110	010101 0001
D11.7	111 01011	EBh	110100 1110	110100 1000
D12.7	111 01100	ECh	001101 1110	001101 0001
D13.7	111 01101	EDh	101100 1110	101100 1000
D14.7	111 01110	EEh	011100 1110	011100 1000
D15.7	111 01111	EFh	010111 0001	101000 1110
D16.7	111 10000	F0h	011011 0001	100100 1110
D17.7	111 10001	F1h	100011 0111	100011 0001
D18.7	111 10010	F2h	010011 0111	010011 0001
D19.7	111 10011	F3h	110010 1110	110010 0001
D20.7	111 10100	F4h	001011 0111	001011 0001
D21.7	111 10101	F5h	101010 1110	101010 0001
D22.7	111 10110	F6h	011010 1110	011010 0001
D23.7	111 10111	F7h	111010 0001	000101 1110
D24.7	111 11000	F8h	110011 0001	001100 1110
D25.7	111 11001	F9h	100110 1110	100110 0001
D26.7	111 11010	FAh	010110 1110	010110 0001
D27.7	111 11011	FBh	110110 0001	001001 1110
D28.7	111 11100	FCh	001110 1110	001110 0001
D29.7	111 11101	FDh	101110 0001	010001 1110
D30.7	111 11110	FEh	011110 0001	100001 1110
D31.7	111 11111	FFh	101011 0001	010100 1110

5.3.7 Control characters

Table 41 defines the control characters (i.e., Kxx.y characters), and shall be used for both generating characters (i.e., encoding) and checking the validity of received characters (i.e., decoding).

Table 41 — Control characters

Name	Control byte		Control character (binary representation) ^a	
	Binary representation (HGF EDCBA)	Hexadecimal representation	Current RD - abcdei fghj	Current RD + abcdei fghj
K28.0	000 11100	1Ch	001111 0100	110000 1011
K28.1 ^b	001 11100	3Ch	<u>001111</u> 1001	<u>110000</u> 0110
K28.2	010 11100	5Ch	001111 0101	110000 1010
K28.3	011 11100	7Ch	001111 0011	110000 1100
K28.4	100 11100	9Ch	001111 0010	110000 1101
K28.5 ^b	101 11100	BCh	<u>001111</u> 1010	<u>110000</u> 0101
K28.6	110 11100	DCh	001111 0110	110000 1001
K28.7 ^{b, c}	111 11100	FCh	<u>001111</u> 1000	<u>110000</u> 0111
K23.7	111 10111	F7h	111010 1000	000101 0111
K27.7	111 11011	FBh	110110 1000	001001 0111
K29.7	111 11101	FDh	101110 1000	010001 0111
K30.7	111 11110	FEh	011110 1000	100001 0111

^a Comma patterns, which are two bits of one polarity followed by five bits of the opposite polarity (i.e., 0011111b or 1100000b), are underlined.

^b K28.1, K28.5, and K28.7 are the only characters which contain comma patterns. Comma patterns do not appear in any data characters and do not appear across any adjacent data characters.

^c The K28.7 control character introduces an additional comma pattern starting with bits i and f when followed by any of the following characters: K28.y, D3.y, D11.y, D12.y, D19.y, D20.y, or D28.y, where y is a value in the range 0 to 7, inclusive. None of the other control characters introduce a comma pattern when adjacent to any other character. Therefore, K28.7 is not used, ensuring that comma patterns do not appear in any sequence of characters except the first 7 bits of K28.1 or K28.5.

The only control characters used in this standard are K28.3, K28.5, and K28.6, as defined in table 42.

Table 42 — Control character usage

First character of a dword	Usage in SAS physical links	Usage in SATA physical links
K28.3	Primitives used only inside STP connections	All primitives except ALIGN
K28.5	ALIGN and most primitives defined in this standard	ALIGN
K28.6	Not used	SATA_ERROR

See 6.2 for details on primitives, which use those control characters.

5.3.8 Encoding characters in the transmitter

To transmit a data byte, the transmitter shall select the appropriate character from table 40 based on the current value of the transmitter's RD. To transmit a control byte, the transmitter shall select the appropriate character from table 41 based on the current value of the transmitter's RD. After the transmitting the character, the transmitter shall calculate a new value for its RD based on that character. This new value shall be used as the transmitter's current RD for the next character transmitted. This process is called 8b10b encoding.

5.3.9 Decoding characters in the receiver

After receiving a character, the receiver shall search the character column in table 40 and table 41 corresponding to its current RD to determine the data byte or control byte to which the character corresponds. This process is called 10b8b decoding. If the received character is not found in the proper column, then the character shall be considered invalid and the dword containing the character shall be considered an invalid dword.

Regardless of the received character's validity, the received character shall be used to calculate a new value of RD in the receiver. This new value shall be used as the receiver's current RD for the next received character.

Detection of a code violation in a character does not always indicate that the character in which the code violation was detected is in error. Code violations may result from an error in a previous character that altered the RD of the bit stream but did not result in a detectable error in the character in which the error occurred. Table 43 shows an example of this behavior. Code violation errors may span dword boundaries. Expanders forwarding a dword in which a code violation was detected, forward the dword as an ERROR (see 6.2.6.7).

Table 43 — Delayed code violation example

	RD	First character	RD	Second character	RD	Third character	RD
Transmitted character stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Bit stream after error	-	101010 10 <u>1</u> 1 (error in second to last bit)	+	010101 0101	+	111010 1010	+
Decoded character stream	-	D21.0 (rather than D21.1) (not detected as an error)	+	D10.2 (no error)	+	Code violation (although D23.5 was received without error)	+

5.4 Dwords, primitives, data dwords, and invalid dwords

All characters transferred in SAS are grouped into four-character sequences called dwords.

A primitive is a dword whose first character is K28.3 or K28.5 and whose remaining three characters are data characters with correct disparity.

Primitives are defined with both negative and positive starting RD (see 5.3.5). SAS defines primitives starting with K28.5 (see 6.2.6 and 6.2.7). SATA defines primitives starting with K28.3 and K28.5, which are used in SAS during STP connections (see 6.2.8).

A data dword is a dword that contains four data characters with correct disparity.

A dword containing an invalid character shall be considered an invalid dword.

5.5 Bit order

Dwords transmitted in an STP connection shall be transmitted in the bit order specified by SATA.

Dwords for other types of connections and outside connections shall be transmitted in the bit order shown in figure 57.

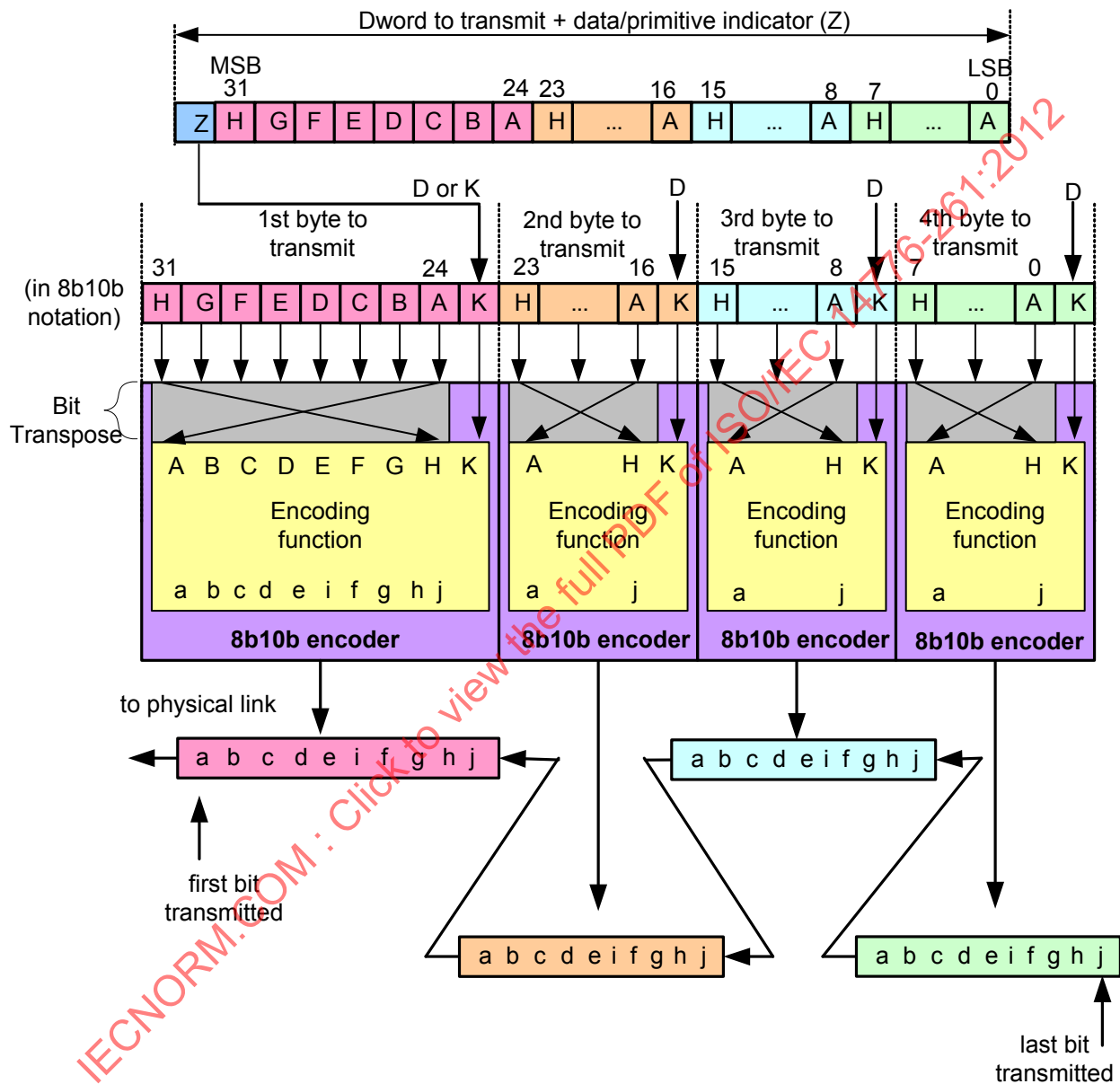


Figure 57 — SAS bit transmission logic

Figure 58 shows the SAS bit reception order.

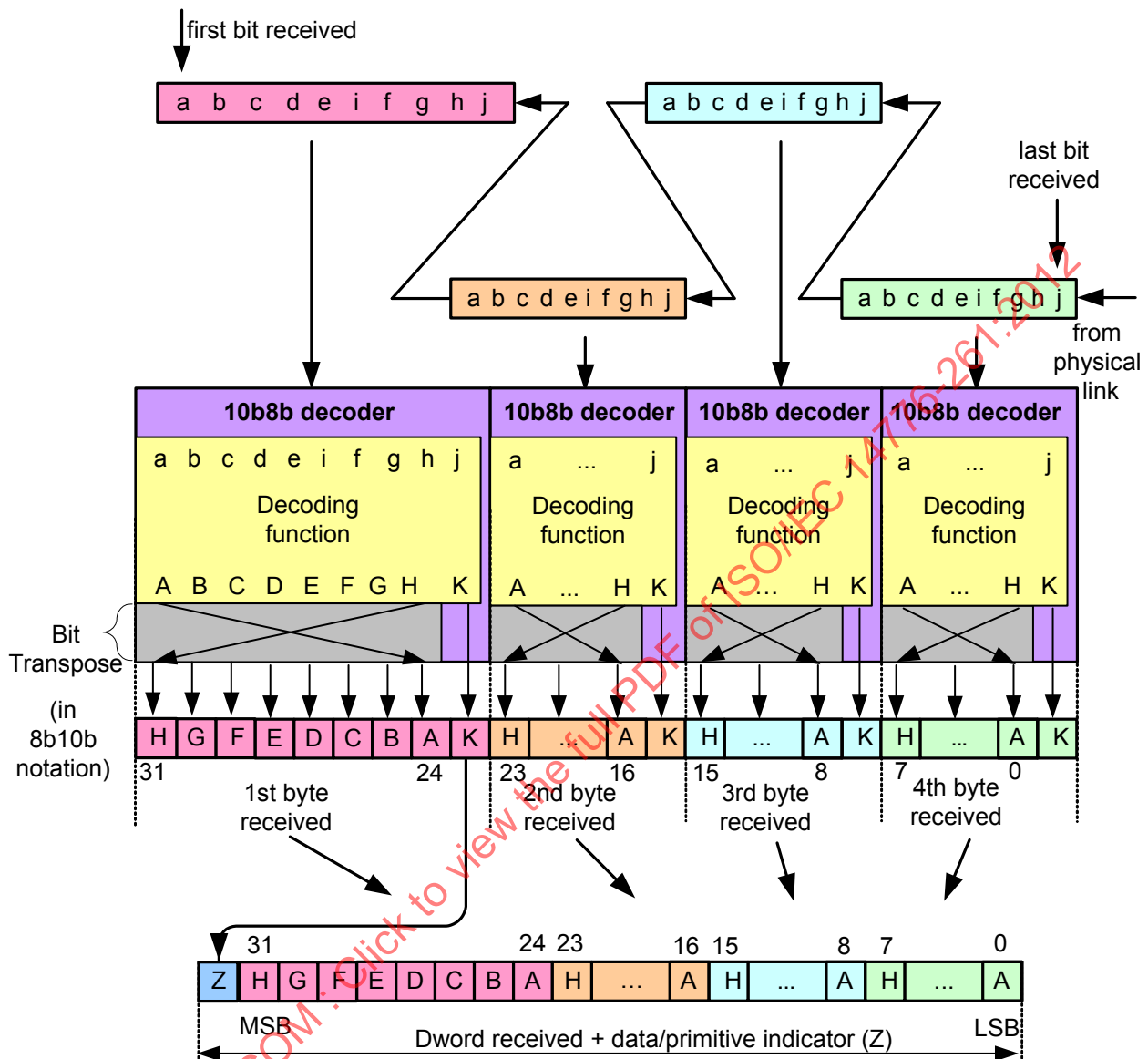


Figure 58 — SAS bit reception logic

5.6 Out of band (OOB) signals

5.6.1 OOB signals overview

For the timing and timing characteristics of transmitted and received OOB signals see SAS-2.1.

5.6.2 SP Transmission of OOB signals

Figure 59 describes OOB signal transmission by the SP transmitter (see 5.9). The COMWAKE Transmitted, COMINIT Transmitted, and COMSAS Transmitted messages are sent to the SP state machine (see 5.9).

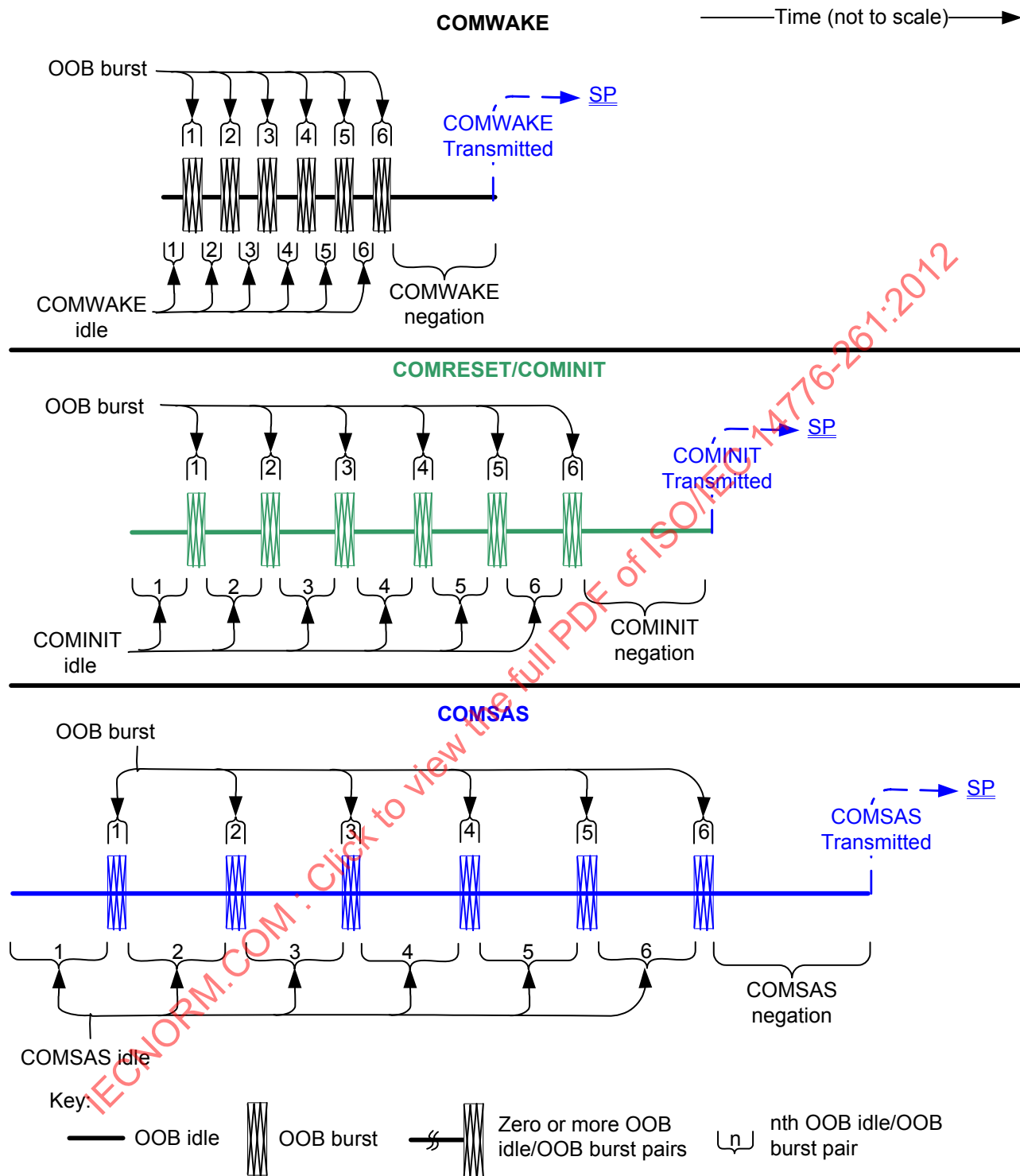


Figure 59 — OOB signal transmission

5.6.3 Receiver detection of OOB signals

If optical mode is enabled or D.C. mode is enabled, then a receiver device shall detect an OOB signal after receiving four consecutive idle time/burst time pairs (see figure 60) while the SP_DWS state machine (see 5.10) has not achieved dword synchronization (see 5.9.4.10 and 5.9.6.8). If optical mode is enabled, then a

receiver device shall detect an OOB signal after receiving four consecutive idle time/burst time pairs while the SP_DWS state machine has achieved dword synchronization. It is not an error to receive more than four idle time/burst time pairs. A receiver device shall not detect the same OOB signal again until it has detected the corresponding negation time (e.g., a COMINIT negation time for a COMINIT) or has detected a different OOB signal (e.g., if a receiver device that previously detected COMINIT receives four COMWAKE idle time/burst time pairs, then that receiver device detects COMWAKE and may then detect COMINIT at a later time).

A SAS receiver device shall not qualify the OOB burst based on the characters received.

Figure 60 describes SAS OOB signal detection by the SP receiver (see 5.9.2 and 5.9.3). The COMWAKE Detected, COMWAKE Completed, COMINIT Detected, COMSAS Detected, and COMSAS Completed messages are sent to the SP state machine (see 5.9) to indicate that an OOB signal has been partially or fully detected.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

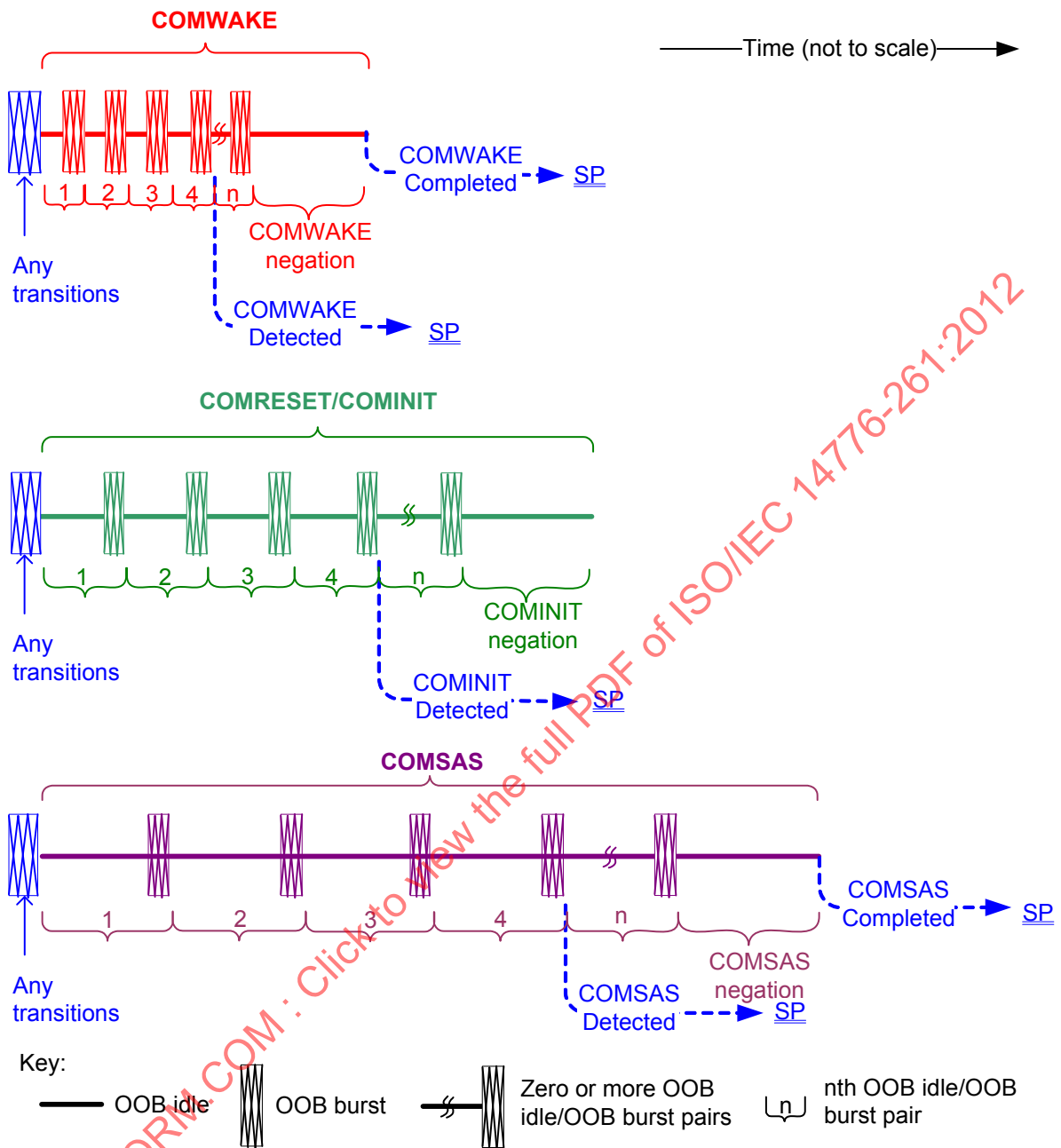


Figure 60 — OOB signal detection

Expander devices shall not forward OOB signals. An expander device shall run the link reset sequence independently on each physical link.

5.6.4 SATA port selection signal

For the timing characteristics of transmitted SATA port selection signal see SAS-2.1.

See 5.9.7 and 9.4.3.28 for information on usage of the SATA port selection signal.

5.6.5 Phy power conditions

During a low phy power condition (see 4.10.1) the phy shall transmit D.C. idle (see SAS-2.1). A phy that is in a low phy power condition shall meet the phy wakeup timeout requirements shown in table 44.

5.7 Phy reset sequences

5.7.1 Phy reset sequences overview

The phy reset sequence consists of:

- 1) an OOB sequence (see 5.7.2.1 and 5.7.4.1);
- 2) a speed negotiation sequence (see 5.7.2.2 and 5.7.4.2); and
- 3) if the physical link is a SAS physical link and multiplexing (see 5.11) is enabled (see table 51 in 5.7.4.2.3.3), a multiplexing sequence (see 5.7.4.3).

The phy reset sequence shall only affect the phy, not the port or device containing the phy or other phys in the same port or device.

A phy shall originate a phy reset sequence after:

- a) power on;
- b) hard reset (i.e., receiving a HARD_RESET primitive sequence before an IDENTIFY address frame) (see 4.4.2);
- c) management application layer request (see 5.9.1);
- d) losing dword synchronization and not attempting to re-acquire dword synchronization (see 5.9.4.10 and 5.9.6.8);
- e) Receive Identify Timeout timer expires (see 6.10);
- f) a hot-plug timeout (see 5.7.5) occurs for an expander phy;
- g) a hot-plug timeout occurs while in a low phy power condition (see 4.10.1); or
- h) the SNTT timer expires while in a low phy power condition.

A SAS phy may originate a phy reset sequence after a hot-plug timeout (see 5.7.5).

After receiving a HARD_RESET primitive sequence before an IDENTIFY address frame, a phy should start the phy reset sequence within 250 ms.

Table 44 defines phy reset sequence timing parameters used by the SP state machine (see 5.9).

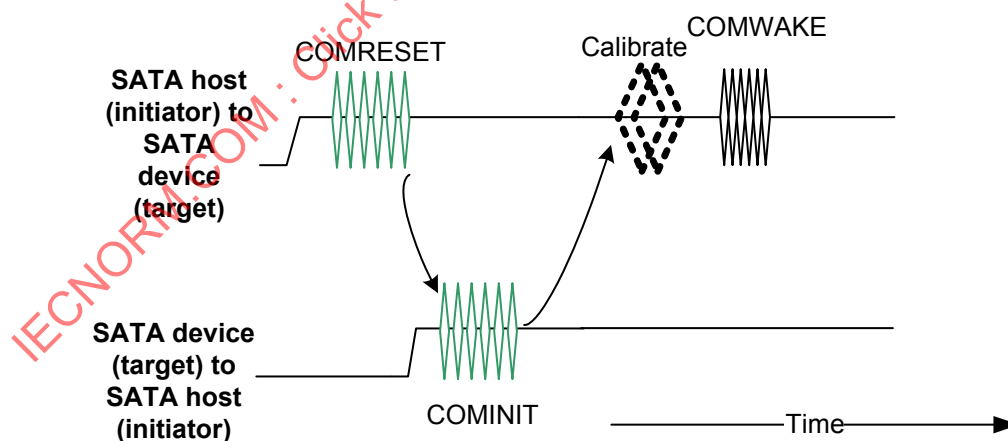
Table 44 — Phy reset sequence timing specifications

Parameter	Minimum	Maximum	Comments
Hot-plug timeout	10 ms	500 ms	The time after which: a) an expander phy shall retry an unsuccessful phy reset sequence; b) a SAS initiator phy should retry an unsuccessful phy reset sequence (see 5.7.5); and c) a phy shall initiate a phy reset sequence if that phy does not receive a COMWAKE Completed message as defined in the SP31:SAS_PS_Low_Phy_Power state (see 5.9.5.2).
Phy wakeup partial timeout	none	10 µs	When a phy is in the partial phy power condition (see 4.10.1.3) or when the management application layer requests an exit power condition procedure (see 5.9.5.2.1), the time within which the phy shall transmit a COMWAKE after detecting a COMWAKE.
Phy wakeup slumber timeout	none	10 ms	When a phy is in the slumber phy power condition (see 4.10.1.4) or when the management application layer requests an exit power condition procedure (see 5.9.5.2.1), the time within which the phy shall transmit a COMWAKE after detecting a COMWAKE.

5.7.2 SATA phy reset sequence

5.7.2.1 SATA OOB sequence

Figure 61 shows the SATA OOB sequence between a SATA host and SATA device. The SATA OOB sequence is defined by SATA.

**Figure 61 — SATA OOB sequence**

5.7.2.2 SATA speed negotiation sequence

Figure 62 shows the speed negotiation sequence between a SATA host and SATA device. The SATA speed negotiation sequence is defined by SATA (see SATA).

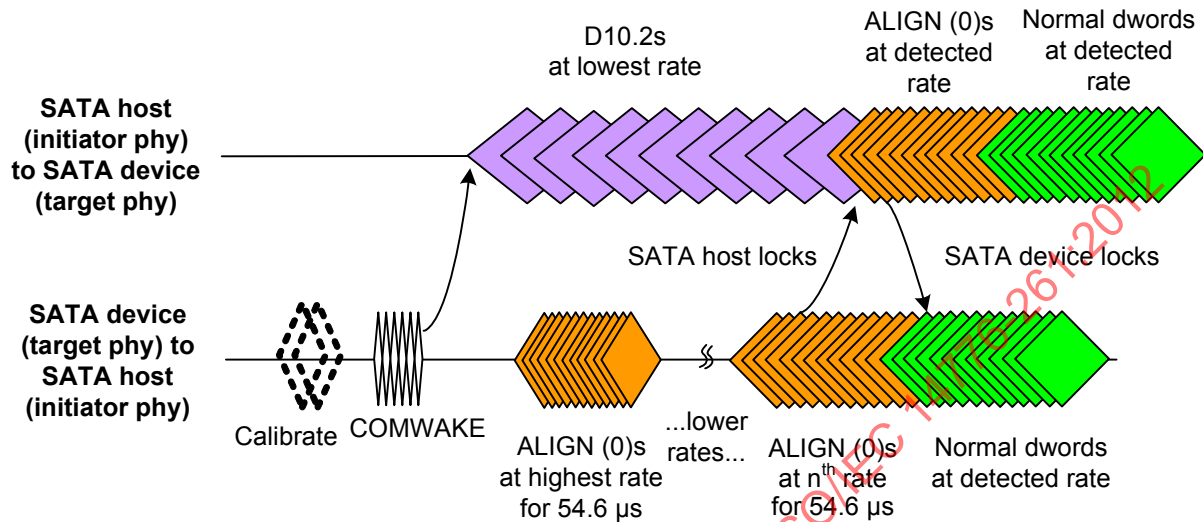


Figure 62 — SATA speed negotiation sequence

Table 45 defines SATA speed negotiation sequence timing parameters used by the SP state machine (see 5.9).

Table 45 — SATA speed negotiation sequence timing specifications

Parameter	Time	Comments
Await ALIGN timeout	873.813 μs ^a	The minimum time during SATA speed negotiation that a phy shall allow for an ALIGN (0) to be received after detecting COMWAKE Completed.
COMWAKE response time	533 ns ^b	The maximum time during SATA speed negotiation after detecting COMWAKE Completed before which a phy shall start transmitting D10.2 characters.
^a 873.813 μs is 32 768 × 40 × nominal OOB (see SAS-2.1 and SATA). ^b 533 ns is 200 × 40 × nominal OOB (see SATA).		

The transmitter device shall use SATA signal output levels during the SATA speed negotiation sequence as described in SAS-2.1.

The phy shall not transmit deletable primitives for physical link rate tolerance management (see 6.3) during the SATA speed negotiation sequence.

5.7.3 SAS to SATA phy reset sequence

SAS initiator phys and expander phys may support SATA (e.g., support being directly attached to a SATA device or a SATA port selector).

To initiate a phy reset sequence a SAS initiator phy or expander phy shall:

- 1) transmit a COMINIT; and
- 2) in response to receiving a COMINIT, transmit a COMSAS.

The COMSAS identifies the phy as a SAS phy or expander phy instead of a SATA phy.

If a SATA phy is attached to the physical link, that SATA phy either:

- a) misinterprets the COMSAS to be a COMRESET and responds with a COMINIT; or
- b) ignores the COMSAS and provides no response within a COMSAS detect timeout.

Either response indicates to the SAS initiator phy or expander phy that a SATA phy is attached. As a result, the SAS initiator phy or expander phy shall transmit COMWAKE and enter the SATA speed negotiation sequence (see 5.7.2.2).

Figure 63 shows an OOB sequence between a SAS phy or expander phy (i.e., a phy compliant with this standard) and a SATA phy (i.e., a phy in a SATA device, defined by SATA). The two possible cases are presented. The first case is that the SATA phy ignores the COMSAS and provides no response within a COMSAS detect timeout. The second case is that the SATA phy misinterprets the COMSAS to be a COMRESET and responds with a COMINIT. The SP state machine treats these two cases the same (see 5.9.3.9), and determines that a SATA phy is attached after a COMSAS detect timeout. The SATA speed negotiation sequence is entered after COMWAKE is detected.

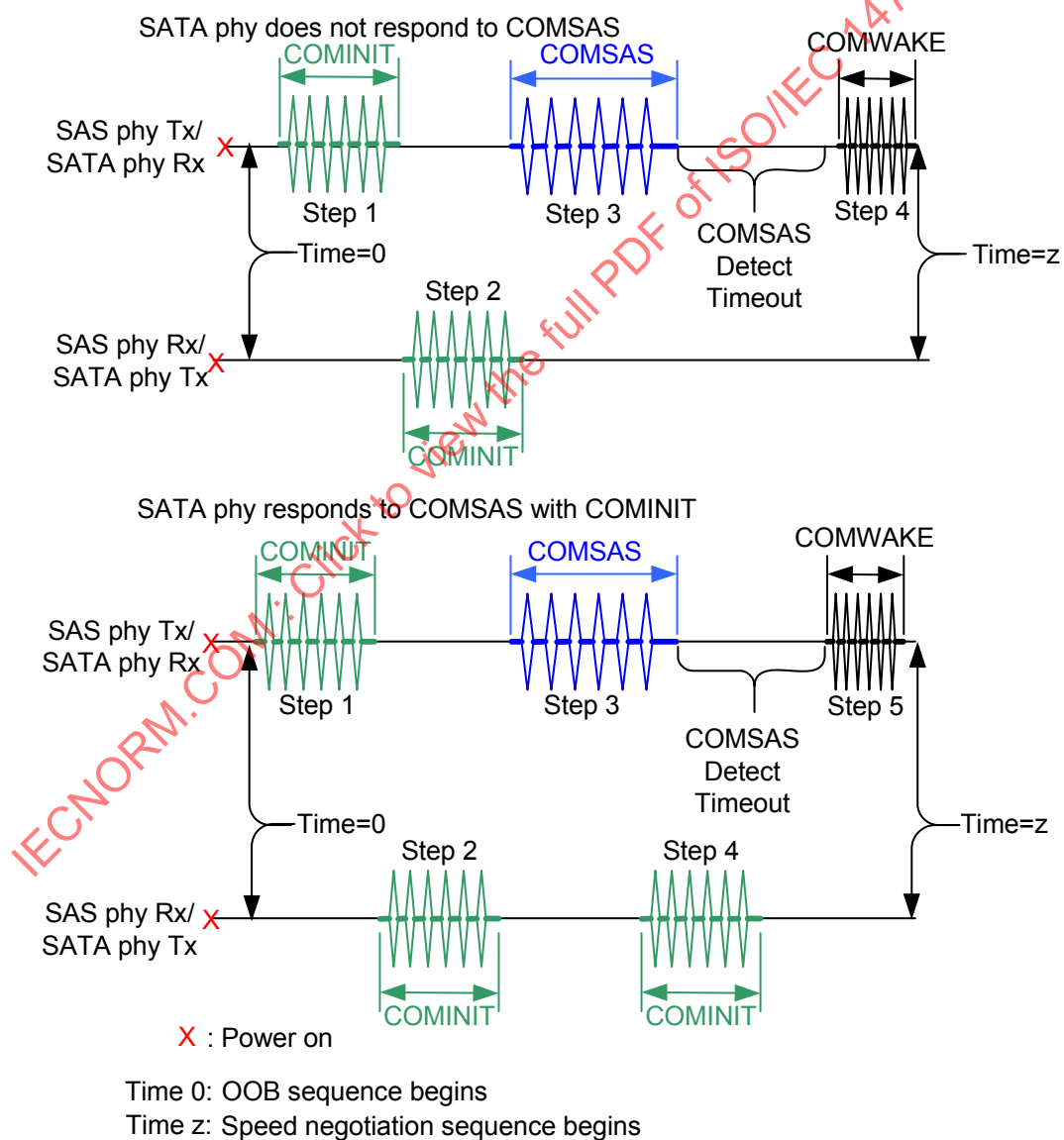


Figure 63 — SAS to SATA OOB sequence

5.7.4 SAS to SAS phy reset sequence

5.7.4.1 SAS OOB sequence

To initiate a SAS OOB sequence a phy transmits a COMINIT.

On receipt of a COMINIT the receiving phy either:

- a) if the phy has not yet transmitted a COMINIT, transmit a COMINIT followed by a COMSAS; or
- b) if the phy has transmitted a COMINIT, transmit a COMSAS.

NOTE 16 If the receiving phy does not respond to a COMINIT within the minimum hot-plug timeout (see 5.7.5), then the attached phy may transmit another COMINIT. If repeated, then this results in a livelock.

On receipt of a COMSAS, if the receiving phy has not yet transmitted a COMSAS, then the phy transmits a COMSAS.

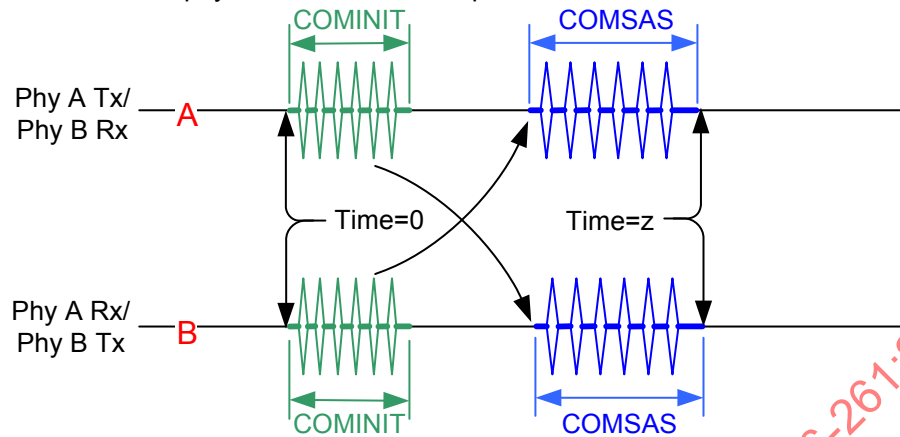
After completing the transmission of a COMSAS and the successful receipt of a COMSAS the SAS OOB sequence is complete and the SAS speed negotiation sequence begins.

A phy distinguishes between COMINIT and COMSAS and continues with a SAS speed negotiation sequence (see 5.7.4.2) after completing the SAS OOB sequence.

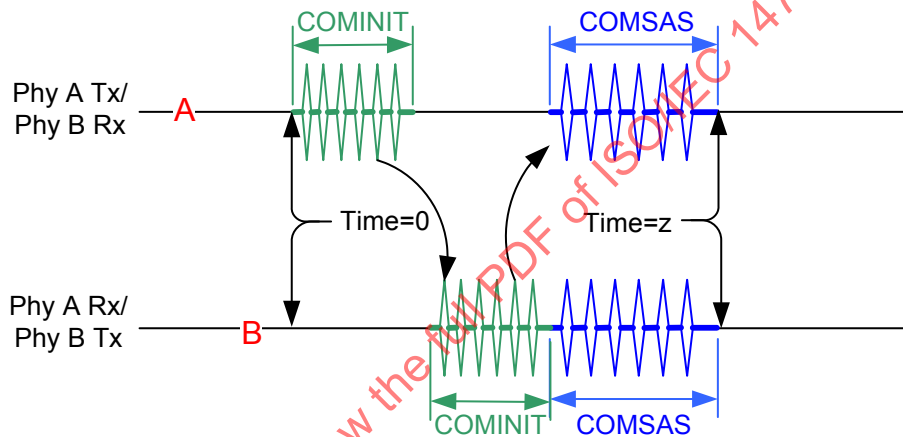
Figure 64 shows several different SAS OOB sequences between phy A and phy B, with phy A starting the SAS OOB sequence at the same time as phy B, before phy B, and before phy B powers on.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

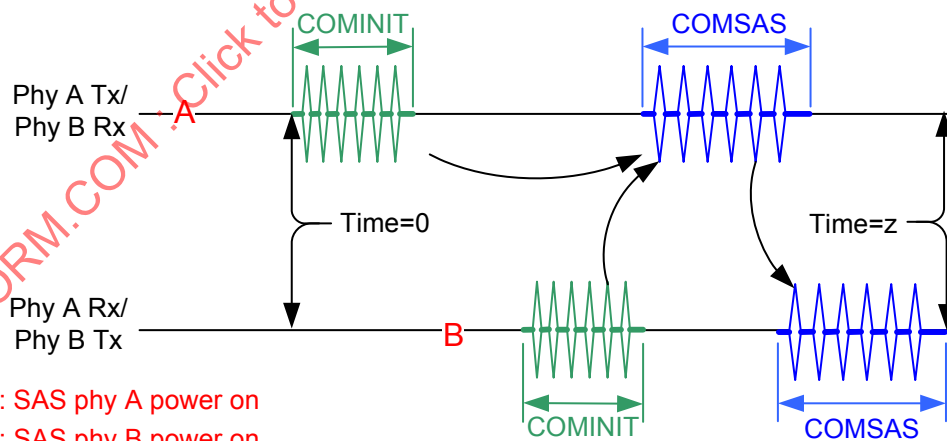
Scenario 1: Both SAS phys start SAS OOB sequence at same time



Scenario 2: SAS phy A starts SAS OOB sequence



Scenario 3: SAS phy B misses SAS phy A's COMINIT



A : SAS phy A power on
B : SAS phy B power on

Time 0: SAS phy reset sequence begins

Time z: SAS speed negotiation sequence begins

Figure 64 — SAS to SAS OOB sequence

5.7.4.2 SAS speed negotiation sequence

5.7.4.2.1 SAS speed negotiation sequence overview

The SAS speed negotiation sequence establishes communications between the two phys on a physical link at the highest possible transmission rate.

The SAS speed negotiation sequence is a peer-to-peer negotiation technique that does not assume initiator and target (i.e., host and device) roles. The rules for speed negotiation are the same for both participating phys.

The SAS speed negotiation sequence consists of a set of speed negotiation windows (SNWs). Each SNW is identified by a name (e.g., Speed Negotiation Window-1 or SNW-1).

SNWs conform to one of three defined types:

- a) speed negotiation without training: SNW-1, SNW-2 and Final-SNW (see 5.7.4.2.3.2);
- b) phy capabilities exchange: SNW-3 (see 5.7.4.2.3.3); and
- c) speed negotiation with training: Train-SNW (see 5.7.4.2.3.4).

Many of the timing parameters used for defining the SNWs are common to multiple SNW types. All of the timing specifications for all SNW types are defined in 5.7.4.2.2.

A SAS speed negotiation sequence may or may not include all three types of SNWs. Phys may implement a subset of SNWs provided that the subset implements a valid speed negotiation sequence. SAS speed negotiation sequences are defined in 5.7.4.2.4.

The transmitter device shall use SAS signal output levels during the SAS speed negotiation sequence as described in SAS-2.1.

The phy shall not transmit deletable primitives for physical link rate tolerance management (see 6.3) during the SAS speed negotiation sequence.

5.7.4.2.2 SAS speed negotiation sequence timing specifications

Table 46 defines the timing specifications for the SAS speed negotiation sequence.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Table 46 — SAS speed negotiation sequence timing specifications

Parameter	Acronym	Time ^a	Comments
Rate change delay time	RCDT	750 000 OOB ^b	The time the transmitter device shall transmit negotiation idle at the beginning of SNW-1, SNW-2, SNW-3, Final-SNW, and Train-SNW.
Speed negotiation transmit time	SNTT	163 840 OOB ^c	During SNW-1, SNW-2, and Final-SNW, the time after RCDT during which ALIGN (0) or ALIGN (1) is transmitted. During SNW-3, the time after RCDT in which bit cells and OOB idle are transmitted. During low phy power conditions, the maximum time for a phy to transmit ALIGN (0)s or ALIGN (1)s while in the SP32:SAS_PS_ALIGN0 state or SP33:SAS_PS_ALIGN1 state (see 5.9.5.4).
Speed negotiation lock time	SNLT	153 600 OOB ^d	The maximum time for a phy to reply with ALIGN (1) during SNW-1, SNW-2, and Final-SNW. During low phy power conditions, the maximum time for a phy to reply with an ALIGN (0) or ALIGN (1) while in the SP32:SAS_PS_ALIGN0 state or SP33:SAS_PS_ALIGN1 state (see 5.9.5.4).
Actual lock time	none		The time during SNW-1, SNW-2, and Final-SNW at which actual dword synchronization occurs to the received ALIGN (0) or ALIGN (1) and the phy begins transmitting ALIGN (1) rather than ALIGN (0).
SNW time	SNWT	913 840 OOB ^e	The duration of SNW-1, SNW-2, SNW-3, or Final-SNW.
Bit cell time	none	2 200 OOB ^f	The time to transmit a COMWAKE or OOB idle during SNW-3.
Maximum training time	MTT	29 998 080 OOB ^g	The maximum time for training to complete during Train-SNW.
Training lock time	TLT	28 497 920 OOB ^h	The maximum time for a phy to reply with TRAIN_DONE during Train-SNW.
Actual training time	none		The time at which training of the receiver is complete during Train-SNW.
Train-SNW time	none		The actual duration of Train-SNW.
Maximum Train-SNW time	MTWT	30 748 080 OOB ⁱ	The maximum duration of Train-SNW.

^a OOB is defined in SAS-2.1.

^b 750 000 OOB (e.g., RCDT) is nominally 500 μ s. Equal to: $18\,750 \times 40$ OOB.

^c 163 840 OOB (e.g., SNTT) is nominally 109.226 μ s. Equal to: $4\,096 \times 40$ OOB.

^d 153 600 OOB (e.g., SNLT) is nominally 102.4 μ s. Equal to: $(4\,096 - 256) \times 40$ OOB.

^e 913 840 OOB (e.g., SNWT) is nominally 609.226 μ s. Equal to: RCDT + SNTT.

^f 2 200 OOB is nominally 1 466.6 ns. Equal to the COMWAKE signal time (see SAS-2.1).

^g 29 998 080 OOB (e.g., MTT) is nominally 19.998 72 ms. Equal to: $11\,718 \times 64 \times 40$ OOB. This is the time of the maximum number of complete training patterns that fit into 20 ms.

^h 28 497 920 OOB (e.g., TLT) is nominally 18.998 613 ms. Equal to: $11\,132 \times 64 \times 40$ OOB. This is the time of the maximum number of complete training patterns that fit into 19 ms.

ⁱ 30 748 080 OOB (e.g., MTWT) is nominally 20.498 72 ms. Equal to: RCDT + MTT.

5.7.4.2.3 Speed negotiation window (SNW) definitions

5.7.4.2.3.1 SNW definitions overview

During each SNW, a phy shall either:

- a) if it supports the SNW, transmit and receive as defined for the SNW; or
- b) if it does not support the SNW, transmit negotiation idle and ignore the SNW information received.

If a phy supports the SNW and receives the expected transmission, then the SNW is valid. If a phy does not receive the expected transmission from the attached phy, then the SNW is invalid.

NOTE 17 If a phy transmits negotiation idle during a SNW, then the attached phy does not receive the expected transmission and the SNW is invalid.

5.7.4.2.3.2 SNW-1, SNW-2, and Final-SNW

Figure 65 defines SNW-1, SNW-2, and Final-SNW, including:

- a) SNW time (SNWT);
- b) rate change delay time (RCDT);
- c) speed negotiation transmit time (SNTT);
- d) speed negotiation lock time (SNLT); and
- e) actual lock time.

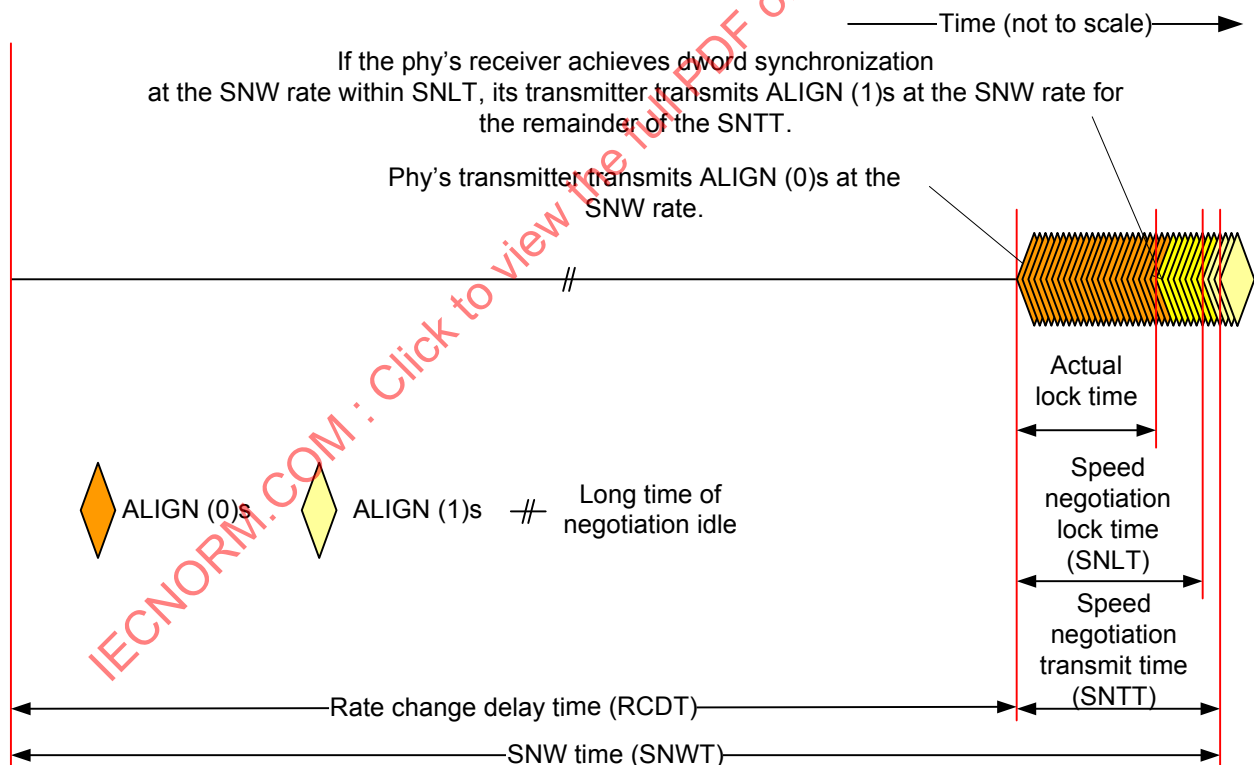


Figure 65 — SNW-1, SNW-2, and Final-SNW

If the phy supports the SNW, then it shall transmit:

- 1) negotiation idle for an RCDT; and
- 2) ALIGNs at the SNW rate for the remainder of the SNWT (i.e., for SNTT).

If the phy does not support the SNW, then it shall transmit negotiation idle for the entire SNWT.

Table 47 defines the SNW rate used in SNW-1, SNW-2, and Final-SNW.

Table 47 — SNW rates used in SNW-1, SNW-2, and Final-SNW

SNW	SNW rate
SNW-1	1.5 Gbps
SNW-2	3 Gbps
Final-SNW	Based on SNW-1, SNW-2, and SNW-3 validity: a) 1.5 Gbps if SNW-1 is valid and SNW-2 is invalid; or b) 3 Gbps if SNW-2 is valid and SNW-3 is invalid.

If the phy supports the SNW, then after RCDT it shall attempt to attain dword synchronization on an incoming series of dwords (e.g., ALIGN (0) or ALIGN (1) primitives) at that rate for the SNLT:

- if the phy achieves dword synchronization within the SNLT, then it shall change from transmitting ALIGN (0) primitives to transmitting ALIGN (1) primitives for the remainder of the SNTT (i.e., the remainder of the SNW time). The point at which the phy achieves dword synchronization is called the actual lock time; or
- if the phy does not achieve dword synchronization within the SNLT, then it shall continue transmitting ALIGN (0) primitives for the remainder of the SNTT (i.e., the remainder of the SNW time).

At the end of the SNTT:

- if the phy is both transmitting and receiving ALIGN (1) primitives, then it shall consider the SNW to be valid; or
- if the phy is not both transmitting and receiving ALIGN (1) primitives, then it shall consider the SNW to be invalid.

The phy shall disable SSC (see SAS-2.1) during SNW-1, SNW-2, and Final-SNW.

5.7.4.2.3.3 SNW-3

SNW-3 allows the phys to exchange phy capabilities to establish phy parameters used in Train-SNW.

Figure 66 defines SNW-3, including:

- SNW time (SNWT);
- rate change delay time (RCDT); and
- speed negotiation transmit time (SNTT).

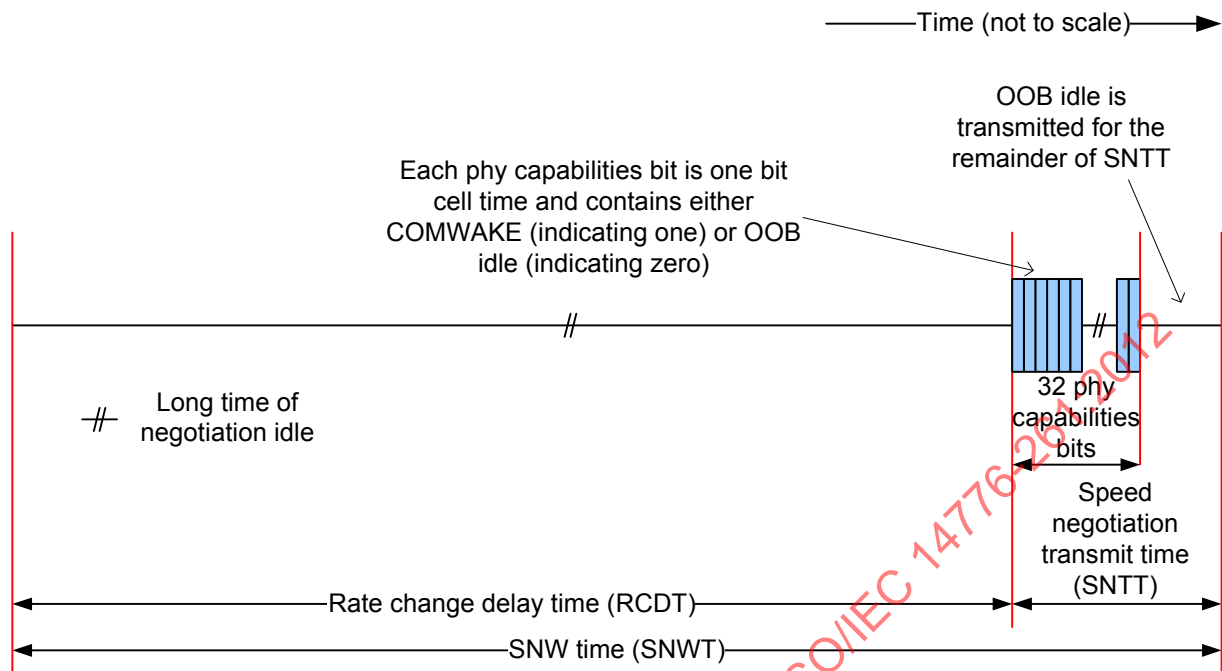


Figure 66 — SNW-3

Table 48 defines the content of each phy capabilities bit.

Table 48 — SNW-3 phy capabilities bit

Value	Transmitted
One	COMWAKE (see 5.6)
Zero	OOB idle

If the phy supports SNW-3, then:

- the phy shall:
 - transmit negotiation idle for an RCDT;
 - transmit 32 phy capabilities bits; and
 - transmit OOB idle for the remainder of SNTT;
 and
- the phy shall receive a 32-bit phy capabilities value from the attached phy.

The phy receives no capabilities bits (i.e., negotiation idle) if the attached phy does not support SNW-3.

If the phy does not support SNW-3, then it shall transmit negotiation idle for the entire SNWT and ignore any phy capabilities bits received.

The first phy capabilities bit is the START bit and is set to one. Each of the remaining 31 phy capabilities bits is set to one or zero. The receiver shall use the START bit to detect the beginning of the phy capabilities bits and establish the timing for subsequent bits.

The phy shall consider SNW-3 to be valid if it supports SNW-3 and receives at least one phy capabilities bit set to one. If the phy does not support SNW-3 or does not receive at least one phy capabilities bit set to one, then it shall consider SNW-3 to be invalid.

The phy may transmit with SSC enabled or disabled (see SAS-2.1) during SNW-3.

Table 49 defines the SNW-3 phy capabilities. For each bit defined as reserved, the phy shall transmit a zero (i.e., OOB idle) and shall ignore the received value. Byte 0 shall be transmitted first and byte 3 shall be transmitted last. Within each byte, bit 7 shall be transmitted first and bit 0 shall be transmitted last (e.g., overall, the START bit is transmitted first and the PARITY bit is transmitted last).

Table 49 — SNW-3 phy capabilities

Byte\Bit	7	6	5	4	3	2	1	0
0	START (1b)	TX SSC TYPE	Reserved		REQUESTED LOGICAL LINK RATE			
1	Supported settings						Reserved	
	G1 WITHOUT SSC	G1 WITH SSC	G2 WITHOUT SSC	G2 WITH SSC	G3 WITHOUT SSC	G3 WITH SSC		
2	Reserved							
3	Reserved							PARITY

The START bit shall be set to the value defined in table 49.

A TX SSC TYPE bit set to one indicates that the phy's transmitter uses center-spreading SSC when SSC is enabled (e.g., the phy is an expander phy) (see SAS-2.1). A TX SSC TYPE bit set to zero indicates that the phy's transmitter uses down-spreading SSC when SSC is enabled (e.g., the phy is a SAS phy), or that the phy does not support SSC.

NOTE 18 The phy receiver may use the TX SSC TYPE bit to optimize its CDR circuitry (see SAS-2.1).

The REQUESTED LOGICAL LINK RATE field indicates if the phy supports multiplexing (see 5.11) and, if so, the logical link rate that the phy is requesting.

Table 50 defines the requested logical link rate based on the transmitted and received REQUESTED LOGICAL LINK RATE fields.

Table 50 — Requested logical link rate

Transmitted REQUESTED LOGICAL LINK RATE field	Received REQUESTED LOGICAL LINK RATE field	Requested logical link rate
0h (i.e., no multiplexing)	Any	Negotiated physical link rate
8h (i.e., 1.5 Gbps)	8h (i.e., 1.5 Gbps)	1.5 Gbps
	9h (i.e., 3 Gbps)	
	Ah (i.e., 6 Gbps)	
	Bh to Fh (i.e., future rates)	
9h (i.e., 3 Gbps)	8h (i.e., 1.5 Gbps)	1.5 Gbps
	9h (i.e., 3 Gbps)	3 Gbps
	Ah (i.e., 6 Gbps)	
	Bh to Fh (i.e., future rates)	
Ah (i.e., 6 Gbps)	8h (i.e., 1.5 Gbps)	1.5 Gbps
	9h (i.e., 3 Gbps)	3 Gbps
	Ah (i.e., 6 Gbps)	6 Gbps
	Bh to Fh (i.e., future rates)	

Table 51 defines whether or not multiplexing is enabled and defines the negotiated logical link rate based on the requested logical link rate (see table 50) and the negotiated physical link rate (see 5.7.4.2.4).

Table 51 — Multiplexing negotiation

Requested logical link rate (see table 50)	Negotiated physical link rate	Multiplexing	Negotiated logical link rate
1.5 Gbps	1.5 Gbps	Disabled	1.5 Gbps
	3 Gbps	Enabled	1.5 Gbps
	6 Gbps		3 Gbps
3 Gbps	1.5 Gbps	Disabled	1.5 Gbps
	3 Gbps		3 Gbps
	6 Gbps	Enabled	3 Gbps
6 Gbps	1.5 Gbps	Disabled	1.5 Gbps
	3 Gbps		3 Gbps
	6 Gbps		6 Gbps
Negotiated physical link rate	1.5 Gbps	Disabled	1.5 Gbps
	3 Gbps		3 Gbps
	6 Gbps		6 Gbps

The supported settings bits include the G1 WITHOUT SSC bit, the G1 WITH SSC bit, the G2 WITHOUT SSC bit, the G2 WITH SSC bit, the G3 WITHOUT SSC bit, and the G3 WITH SSC bit.

A G1 WITHOUT SSC bit set to one indicates that the phy supports G1 (i.e., 1.5 Gbps) without SSC. A G1 WITHOUT SSC bit set to zero indicates that the phy does not support G1 without SSC.

A G1 WITH SSC bit set to one indicates that the phy supports G1 (i.e., 1.5 Gbps) with SSC. A G1 WITH SSC bit set to zero indicates that the phy does not support G1 with SSC.

A G2 WITHOUT SSC bit set to one indicates that the phy supports G2 (i.e., 3 Gbps) without SSC. A G2 WITHOUT SSC bit set to zero indicates that the phy does not support G2 without SSC.

A G2 WITH SSC bit set to one indicates that the phy supports G2 (i.e., 3 Gbps) with SSC. A G2 WITH SSC bit set to zero indicates that the phy does not support G2 with SSC.

A G3 WITHOUT SSC bit set to one indicates that the phy supports G3 (i.e., 6 Gbps) without SSC. A G3 WITHOUT SSC bit set to zero indicates that the phy does not support G3 without SSC.

A G3 WITH SSC bit set to one indicates that the phy supports G3 (i.e., 6 Gbps) with SSC. A G3 WITH SSC bit set to zero indicates that the phy does not support G3 with SSC.

Table 52 defines the priority of the supported settings bits.

Table 52 — Supported settings bit priorities

Priority	Bit
Highest	G3 WITH SSC bit
...	G3 WITHOUT SSC bit
...	G2 WITH SSC bit
...	G2 WITHOUT SSC bit
...	G1 WITH SSC bit
Lowest	G1 WITHOUT SSC bit

The PARITY bit provides for error detection for the SNW-3 phy capabilities bits. The PARITY bit shall be set to one or zero such that the total number of SNW-3 phy capabilities bits that are set to one is even, including the START bit and the PARITY bit. If the PARITY bit received is incorrect based upon the received SNW phy capabilities bits, then the parity is bad and the phy shall consider a phy reset problem (see 5.7.4.2.4) to have occurred.

Table 53 lists some example SNW-3 phy capabilities values.

Table 53 — Example SNW-3 phy capabilities values

Code ^a	Description
80540000h	Down-spreading SSC G1, G2, and G3 with SSC supported
80FC0001h	Down-spreading SSC G1, G2, and G3 with and without SSC supported
80A80000h	G1, G2, and G3 without SSC supported
C0FC0000h	Center-spreading SSC G1, G2, and G3 with and without SSC supported
C9FC0000h	Center-spreading SSC Requested 3 Gbps logical link rate G1, G2, and G3 with and without SSC supported
C8F00001h	Center-spreading SSC Requested 1.5 Gbps logical link rate G1 and G2 with and without SSC supported
^a Expressed as a 32-bit value with byte 0 bit 7 (i.e., the START bit) as the MSB and byte 3 bit 0 (i.e., the PARITY bit) as the LSB.	

5.7.4.2.3.4 Train-SNW

Figure 67 defines the Train-SNW, including:

- a) maximum Train-SNW window time (MTWT);
- b) rate change delay time (RCDT);
- c) maximum train time (MTT);
- d) train lock time (TLT); and
- e) actual training time.

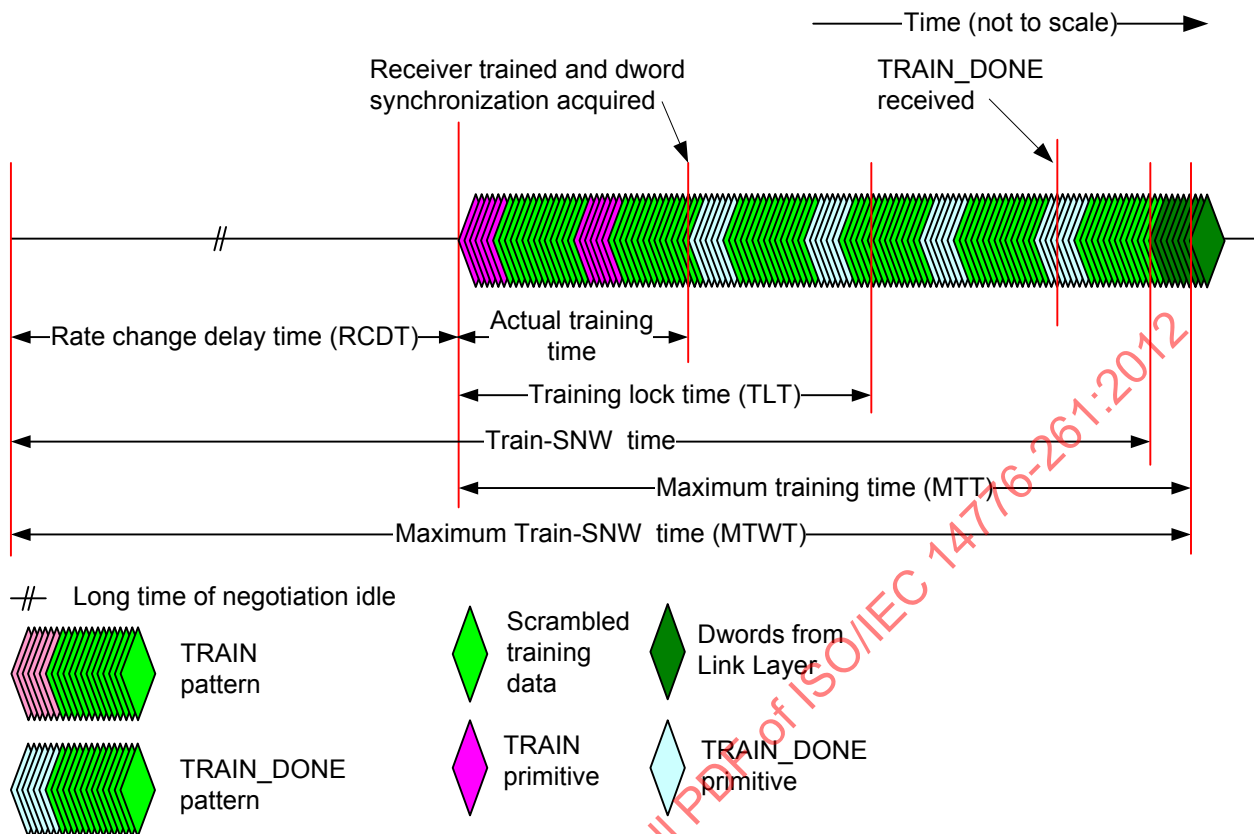


Figure 67 — Train-SNW

The Train-SNW contains training patterns formed by TRAIN and TRAIN_DONE (see 6.2) as defined in table 54.

Table 54 — Training patterns

Training pattern ^a	Description
TRAIN pattern	Sequence of: 1) TRAIN primitive sequence; and 2) 58 dwords set to 00000000h that are transmitted scrambled and 8b10b encoded.
TRAIN_DONE pattern	Sequence of: 1) TRAIN_DONE primitive sequence; and 2) 58 dwords set to 00000000h that are transmitted scrambled and 8b10b encoded.
^a Deletable primitives shall not be transmitted during a training pattern.	

The scrambler is the same as that defined for the link layer (see 6.6) and shall be initialized at the end of RCDT. The scrambler shall not be reinitialized for the remainder of the Train-SNW.

The phy shall start transmitting TRAIN patterns at the end of RCDT. The first TRAIN pattern may have either starting disparity. The number of TRAIN patterns transmitted is determined by the time required for the phy's receiver to complete training and acquire dword synchronization. The phy shall transmit at least one TRAIN pattern and shall transmit a minimum of four TRAIN_DONE patterns.

After RCDT, the phy shall attempt to attain dword synchronization using the commonly supported settings on an incoming series of dwords:

- a) if the phy achieves dword synchronization within the TLT, then, after completing transmission of the current TRAIN pattern, it shall change from transmitting TRAIN patterns to transmitting TRAIN_DONE patterns for the remainder of the Train-SNW window time (i.e., the remainder of the SNW time). The point at which the phy achieves dword synchronization is called the actual training time; or
- b) if the phy does not achieve dword synchronization within the TLT, then it shall continue transmitting TRAIN patterns for the remainder of the TWTT (i.e., the remainder of the SNW time).

The phy shall not perform pattern comparison on the data dwords in the training pattern.

If the phy:

- a) transmits four or more TRAIN_DONE patterns; and
- b) receives a minimum of one TRAIN_DONE primitive sequence before MTT,

then the phy shall:

- a) after completing transmission of the current TRAIN_DONE pattern, transmit at least one more TRAIN_DONE pattern, stop transmitting TRAIN_DONE patterns, and start transmitting dwords from the link layer; and
- b) consider the Train-SNW to be valid.

If the phy does not receive a TRAIN_DONE primitive sequence before MTT and transmits four or more TRAIN_DONE patterns, then it shall consider the Train-SNW to be invalid.

The phy shall not transmit deletable primitives during Train-SNW.

5.7.4.2.4 SAS speed negotiation sequence

The SAS speed negotiation sequence consists of a set of SNWs (see 5.7.4.2.3) in the order shown in figure 68.

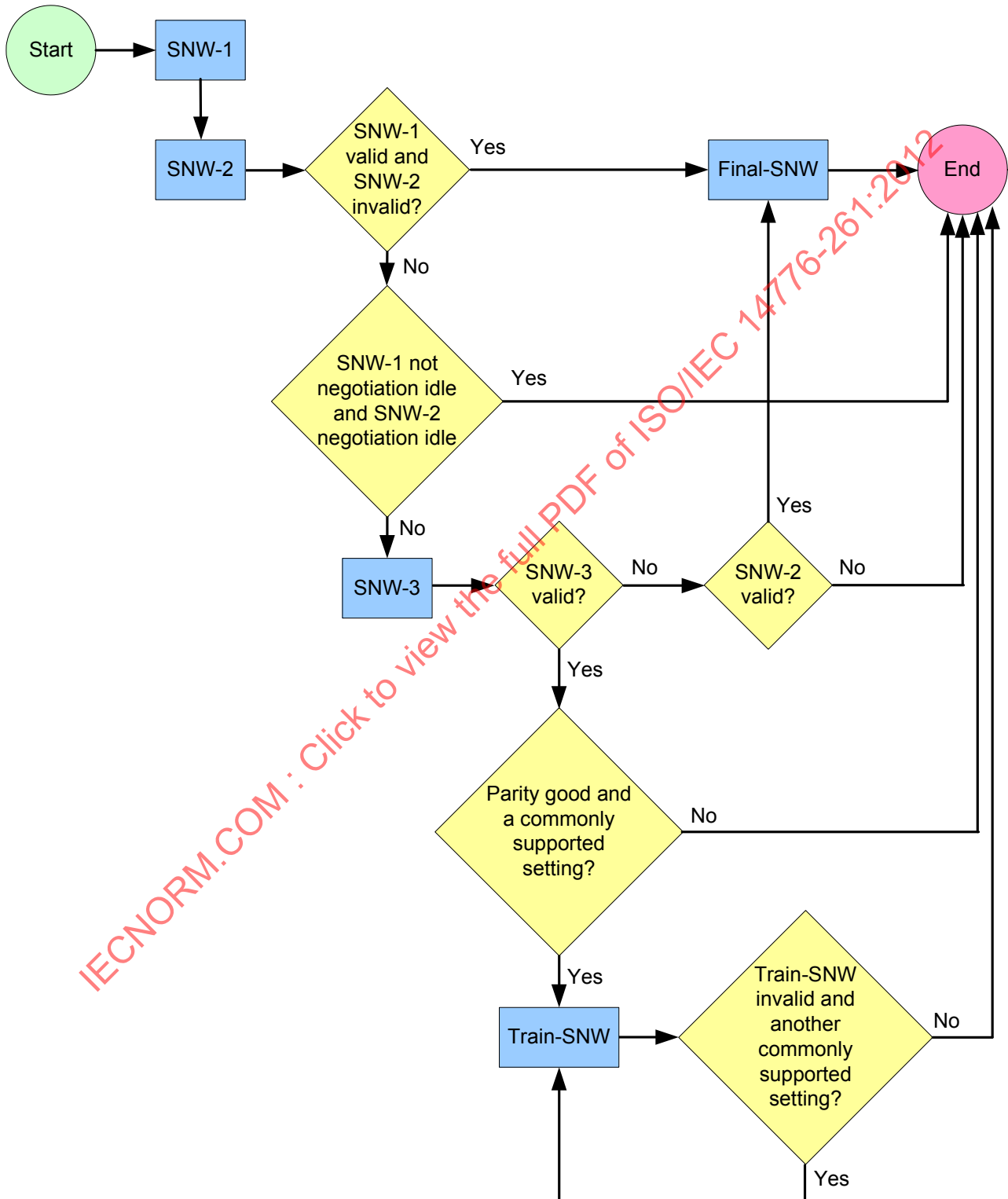


Figure 68 — SAS speed negotiation sequence SNW flowchart

A phy shall not support the following combinations of supported SNWs:

- a) no SNWs; and
- b) SNW-1 and SNW-3 only.

NOTE 19 If SNW-1 is successful and the combination of SNW-1 and SNW-3 only is used, then the phy is not able to reach SNW-3.

A phy shall detect whether the physical link is negotiation idle during SNW-1 and SNW-2, even if the phy does not support that SNW. If the phy detects:

- a) SNW-1 is not negotiation idle; and
- b) SNW-2 is negotiation idle,

then it shall end the speed negotiation sequence without progressing to SNW-3.

NOTE 20 This avoids causing an attached phy compliant with SAS-1.1 to misdetect a SATA port selector.

Train-SNW is based on the highest untried commonly supported settings based on the outgoing and incoming SNW-3 supported settings bits (see 5.7.4.2.3).

If a Train-SNW is invalid and there are additional, untried, commonly supported settings exchanged during SNW-3, then a new Train-SNW shall be performed based on the next highest, untried, commonly supported settings.

A phy reset problem occurs:

- a) after Final-SNW, if Final-SNW is invalid;
- b) after SNW-3, if SNW-3 is valid and the parity is bad; or
- c) after a Train-SNW, if the Train-SNW is invalid and there are no additional, untried, commonly supported settings.

Phy reset problems terminate the SAS speed negotiation sequence and are counted and reported in the PHY RESET PROBLEM COUNT field in the SMP REPORT PHY ERROR LOG page (see 9.4.3.11) and the Protocol-Specific Port log page (see 9.2.8.1).

5.7.4.2.5 SAS speed negotiation sequence examples

Figure 69 shows speed negotiation between a phy A and a phy B where both phys participate in:

- 1) SNW-1, supported by both phys;
- 2) SNW-2, supported by both phys;
- 3) SNW-3, supported by both phys; and
- 4) Train-SNW.

After phy A and phy B detect:

- a) SNW-1 valid;
- b) SNW-2 valid; and
- c) SNW-3 valid,

the phys proceed to Train-SNW negotiating based on SNW-3 phy capabilities bits.

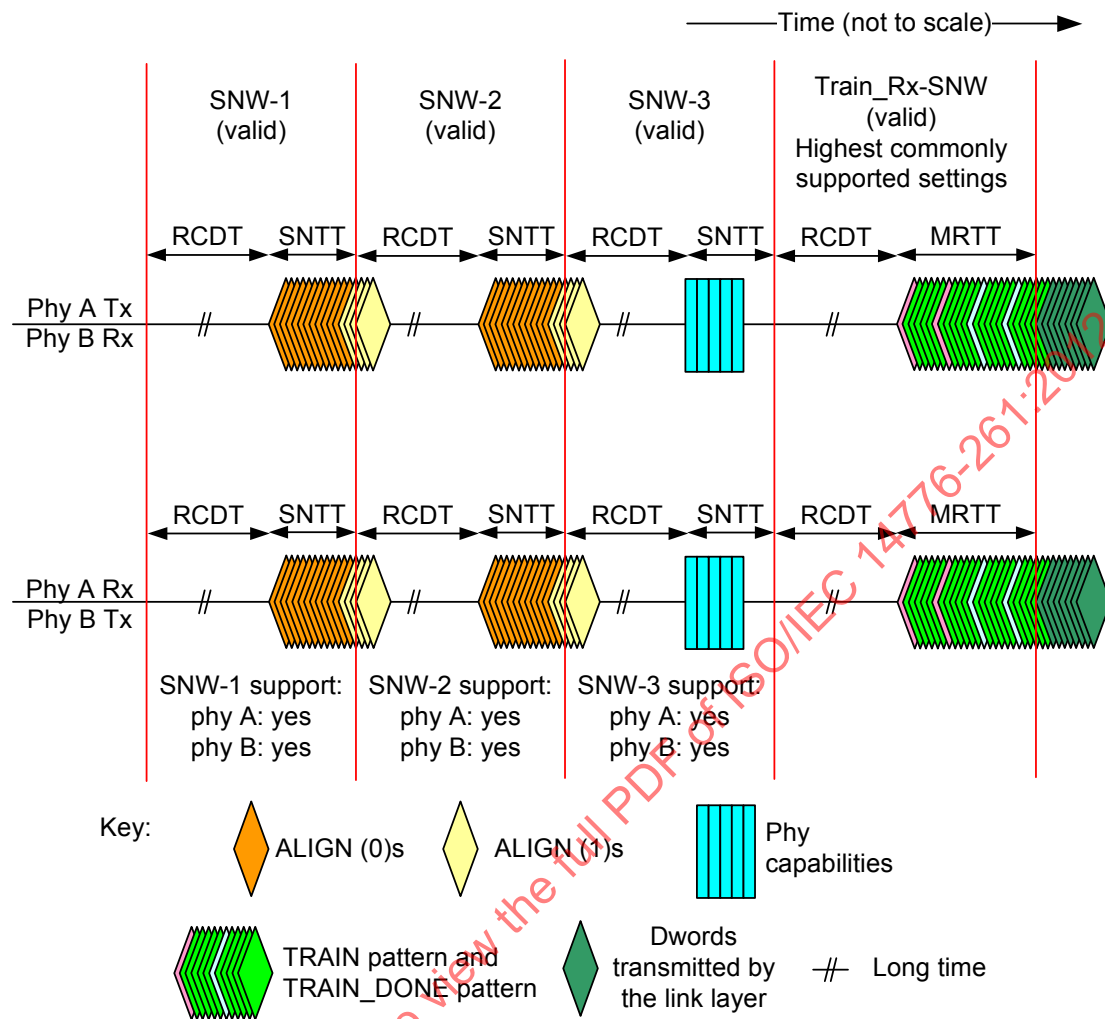


Figure 69 — SAS speed negotiation sequence (both phys SNW-1 through SNW-3)

Figure 70 shows speed negotiation between a phy A and phy B where phys participate in:

- 1) SNW-1, supported by phy A but not by phy B;
- 2) SNW-2, supported by both phys;
- 3) SNW-3, supported by phy A but not by phy B; and
- 4) Final-SNW negotiating 3 Gbps.

After phy A and phy B detect:

- a) SNW-1 invalid;
- b) SNW-2 valid; and
- c) SNW-3 invalid,

the phys proceed to Final-SNW negotiating 3 Gbps.

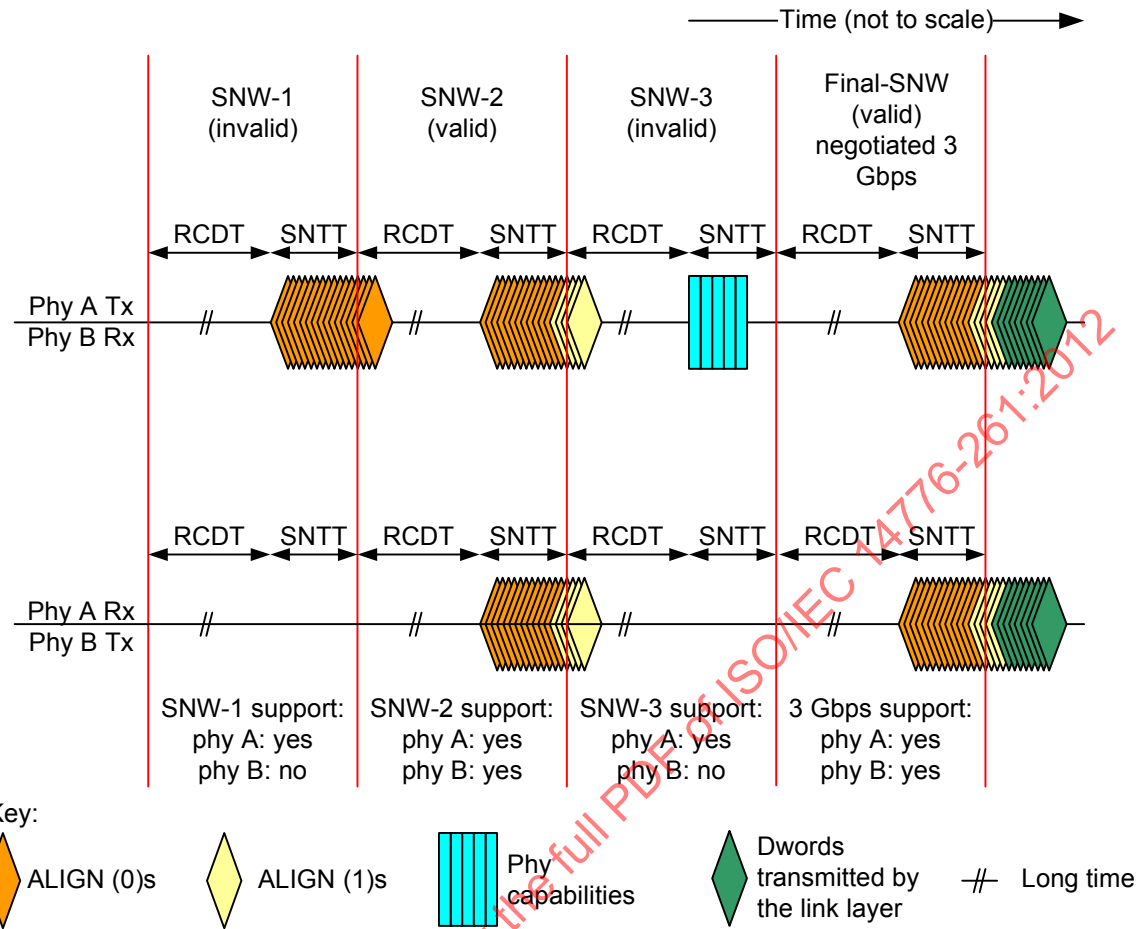


Figure 70 — SAS speed negotiation sequence (phy A: SNW-1 through SNW-3, phy B: SNW-2 only)

Figure 71 shows speed negotiation between a phy A and phy B where the phys participate in:

- 1) SNW-1, supported by phy B but not by phy A; and
- 2) SNW-2, supported by neither phy.

After phy A and phy B detect:

- a) SNW-1 invalid; and
- b) SNW-2 invalid,

phy A detects SNW-3 invalid.

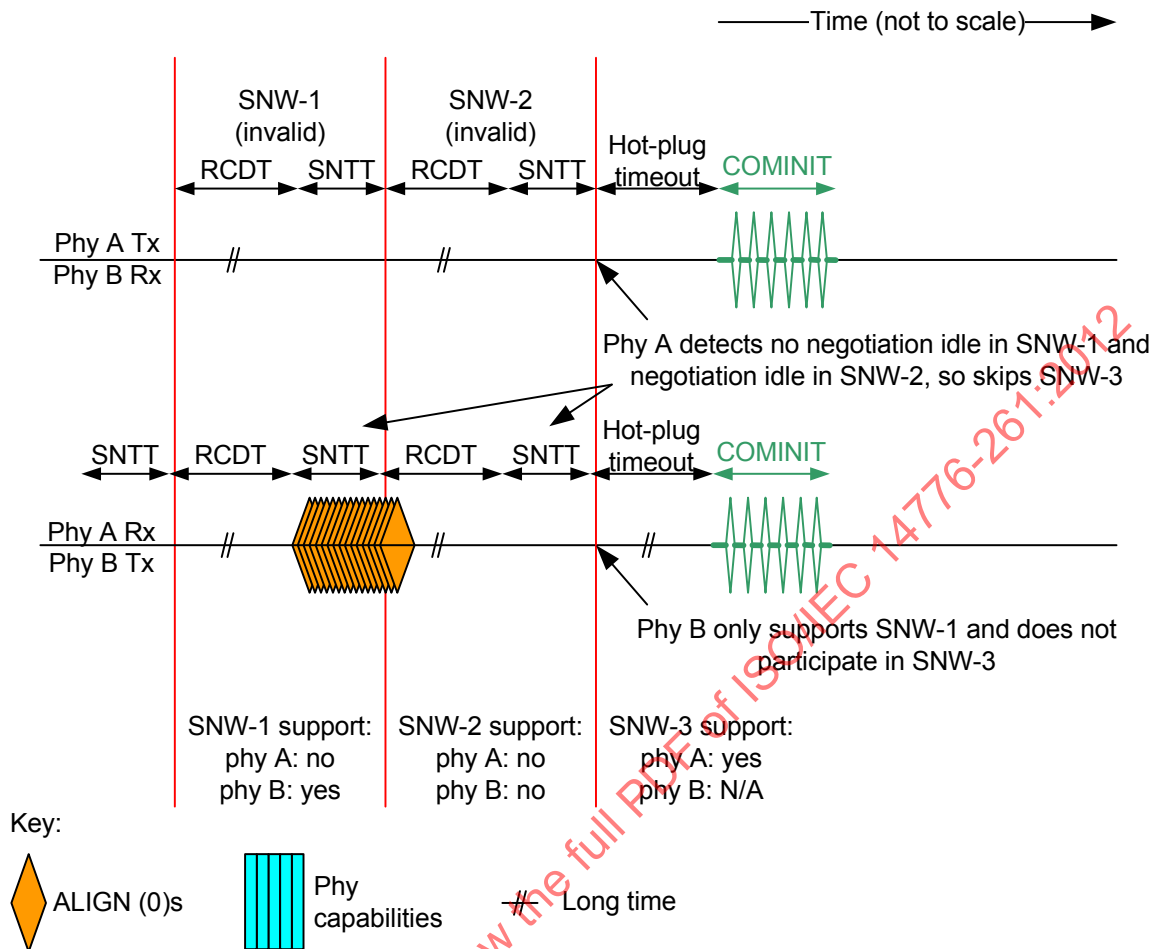


Figure 71 — SAS speed negotiation sequence (phy A: SNW-3 only, phy B: SNW-1 only)

Figure 72 shows a speed negotiation sequence where phy B does not achieve dword synchronization during Final-SNW, creating a phy reset problem. If this occurs, then the handshake is not complete and the phy reset sequence is retried.

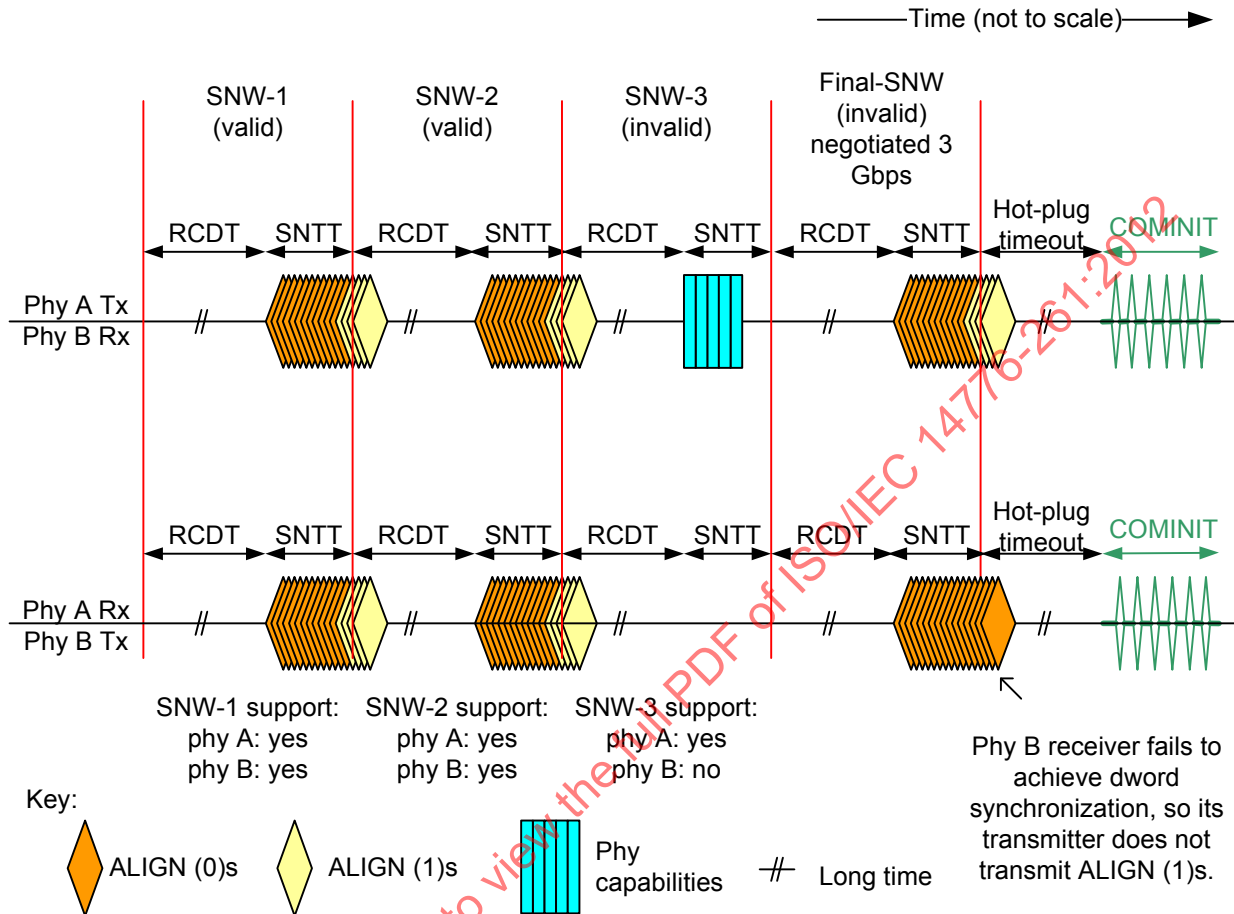


Figure 72 — SAS speed negotiation sequence - phy reset problem in Final-SNW

Figure 73 shows a speed negotiation sequence in which a phy reset problem is encountered in SNW-3 because the phys do not exchange the phy capabilities bits properly (e.g., due to a parity error).

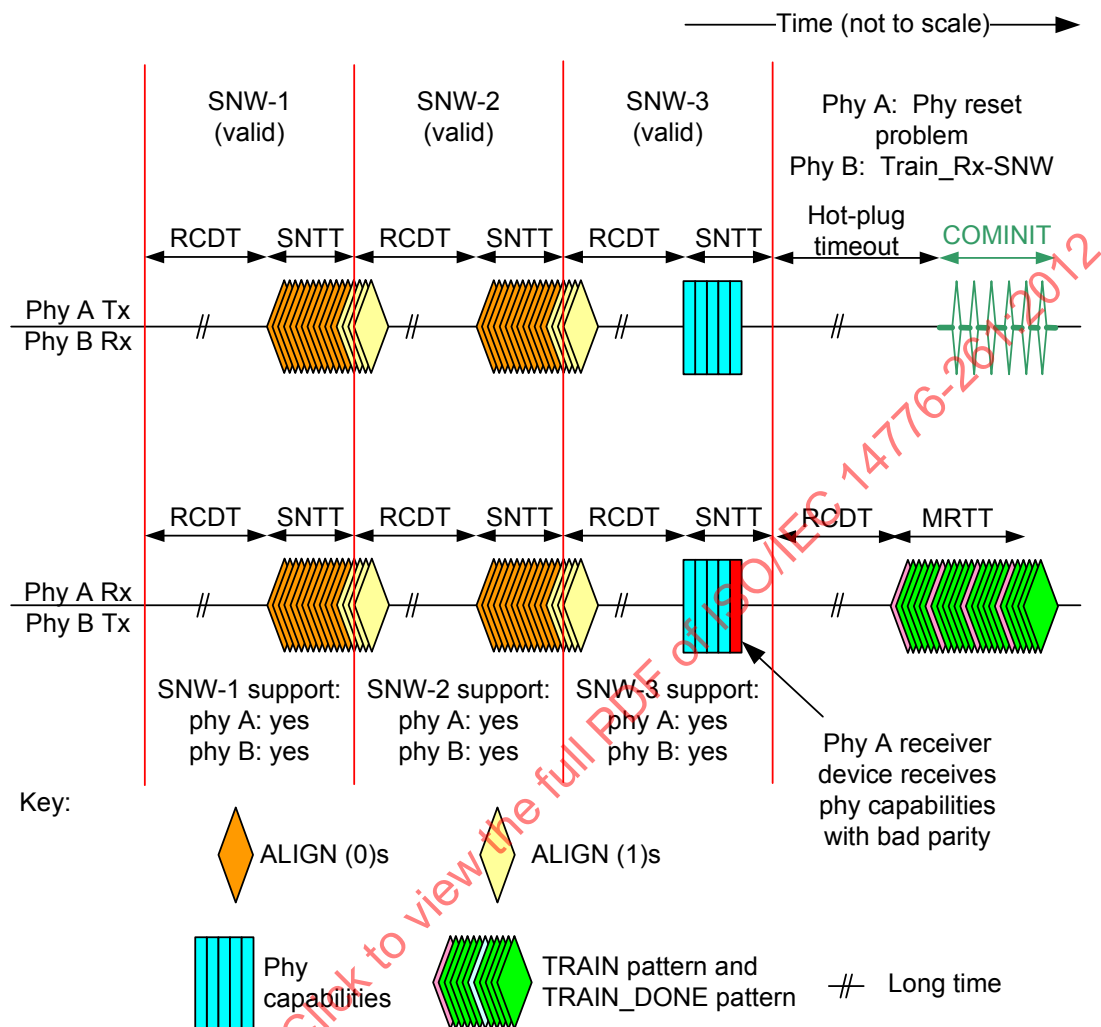


Figure 73 — SAS speed negotiation sequence - phy reset problem in SNW-3

Figure 74 shows a speed negotiation sequence in which a phy reset problem is encountered in Train-SNW because either phy does not complete training within MTT. This example assumes that only one commonly supported setting is exchanged in the phy capabilities bits.

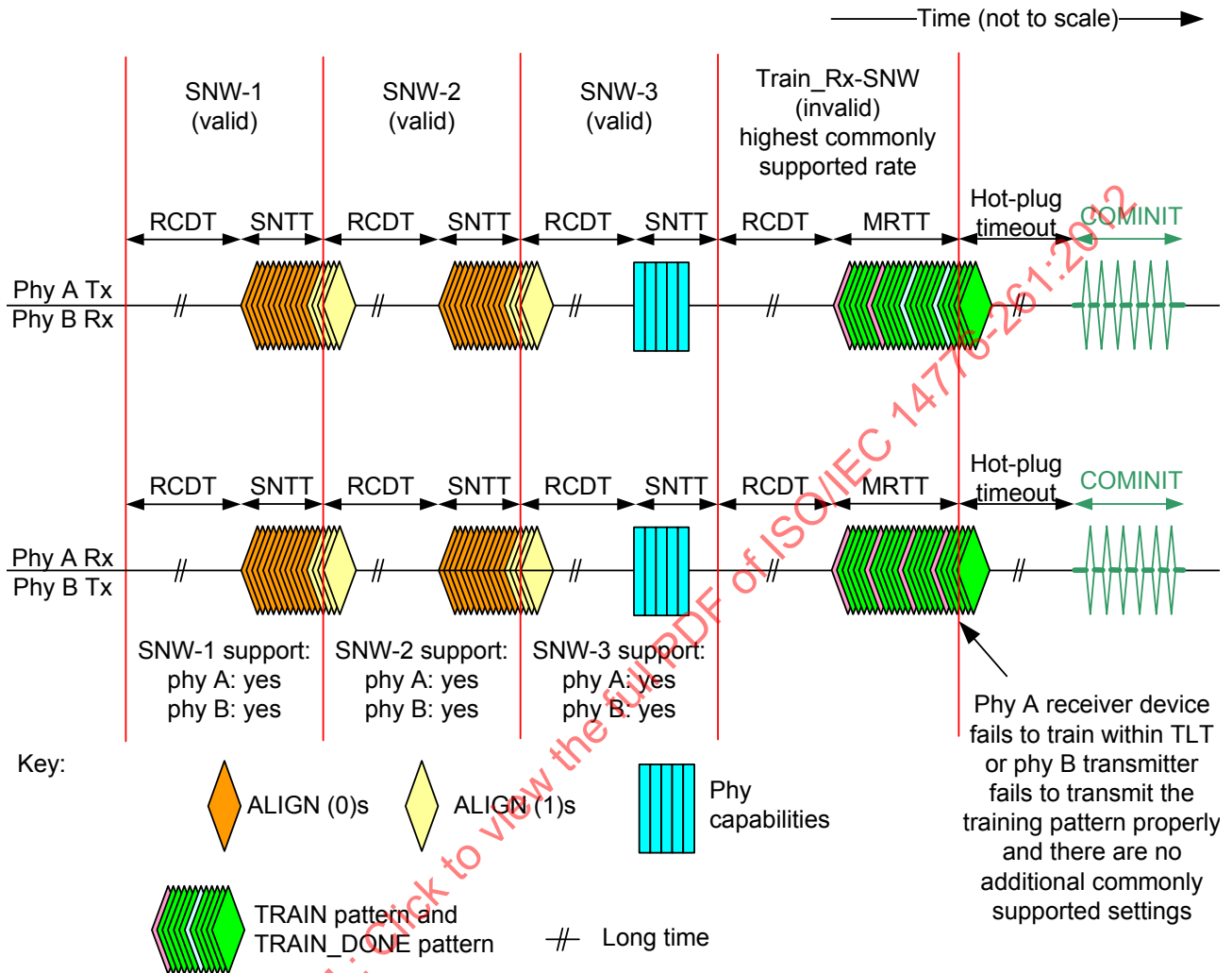


Figure 74 — SAS speed negotiation sequence - phy reset problem in Train-SNW

Figure 75 shows two Train-SNWs, where supported settings bits are exchanged that contain more than one commonly supported setting and the Train-SNW using the highest commonly supported setting is invalid, so a second Train-SNW is performed using the next highest commonly supported setting.

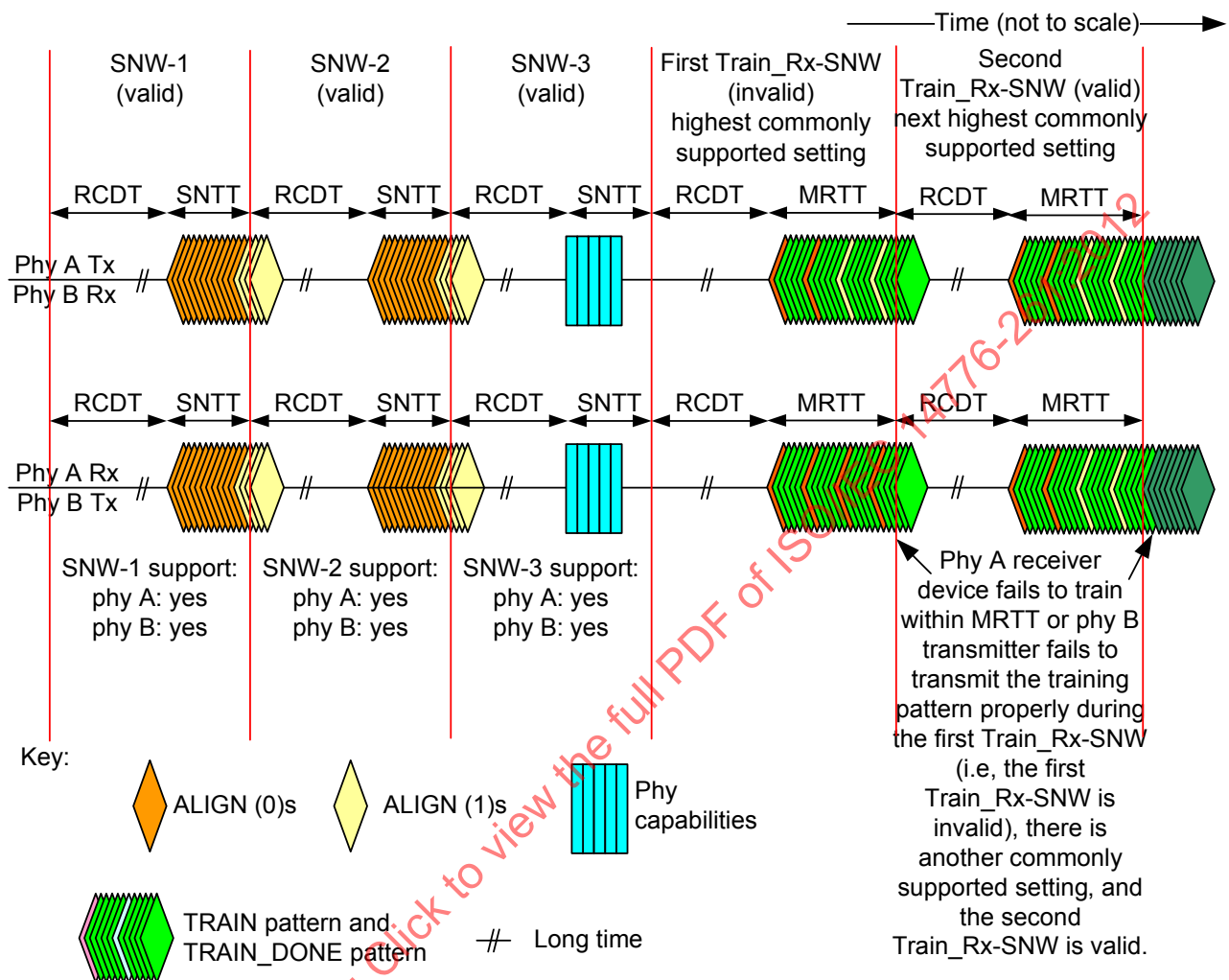


Figure 75 — SAS speed negotiation sequence - multiple Train-SNWs

For more examples of SAS speed negotiations, see Annex B.

5.7.4.3 Multiplexing sequence

If SNW-3 indicates multiplexing (see 5.11) is enabled (see table 51 in 5.7.4.2.3.3), then the phy shall transmit the multiplexing sequence immediately after the speed negotiation sequence.

The multiplexing sequence is:

- 1) MUX (LOGICAL LINK 0);
- 2) MUX (LOGICAL LINK 1);
- 3) MUX (LOGICAL LINK 0);
- 4) MUX (LOGICAL LINK 1);
- 5) MUX (LOGICAL LINK 0); and
- 6) MUX (LOGICAL LINK 1).

The phy shall not transmit deletable primitives for physical link rate tolerance management (see 6.3) during the multiplexing sequence.

If SNW-3 indicates multiplexing is not enabled, then the phy shall not transmit the multiplexing sequence.

The phy shall assign the incoming logical links to its logical phys based on the first MUX it receives:

- a) MUX (LOGICAL LINK 0) indicates the position of logical link 0 and indicates the next dword is in logical link 1; or
- b) MUX (LOGICAL LINK 1) indicates the position of logical link 1 and indicates the next dword is in logical link 0.

The phy shall handle errors during the multiplexing sequence (i.e., after receiving the first MUX) as follows:

- a) if the phy loses dword synchronization, then it shall restart the link reset sequence rather than attempt to reestablish dword synchronization;
- b) if the phy receives a dword that is not a MUX before receiving the MUX expected in that position, then it shall discard the dword;
- c) if the phy receives an invalid dword, then it shall discard the dword; or
- d) if the phy receives a MUX that does not match the MUX expected in that position (i.e., it receives MUX (LOGICAL LINK 1) on logical link 0 or receives MUX (LOGICAL LINK 0) on logical link 1), then it shall restart the link reset sequence.

5.7.5 Phy reset sequence after devices are attached

Since SATA and SAS signal cable connectors do not include power lines, it is not possible to detect the physical insertion of the signal cable connector onto a plug. Non-cabled environments may similarly not have a way to detect physical insertion of a device. As a result, every time a phy reset sequence is originated:

- a) expander phys that are enabled but not active shall originate a new phy reset sequence repeatedly, with no more than a hot-plug timeout (see table 44 in 5.7.1) between each attempt, until a speed negotiation sequence completes without error;
- b) SAS initiator phys should originate a new phy reset sequence after every hot-plug timeout; and
- c) SAS target phys should not originate a new phy reset sequence after their first attempt.

Figure 76 shows how two phys complete the phy reset sequence if the phys are not attached at power on. In this example, phy A and phy B are attached some time before phy B's second hot-plug timeout occurs. Phy B's OOB detection circuitry detects a COMINIT after the attachment, and therefore phy B transmits COMSAS, since it has both transmitted and received a COMINIT. Upon receiving COMSAS, phy A transmits its own COMSAS. The SAS speed negotiation sequence follows.

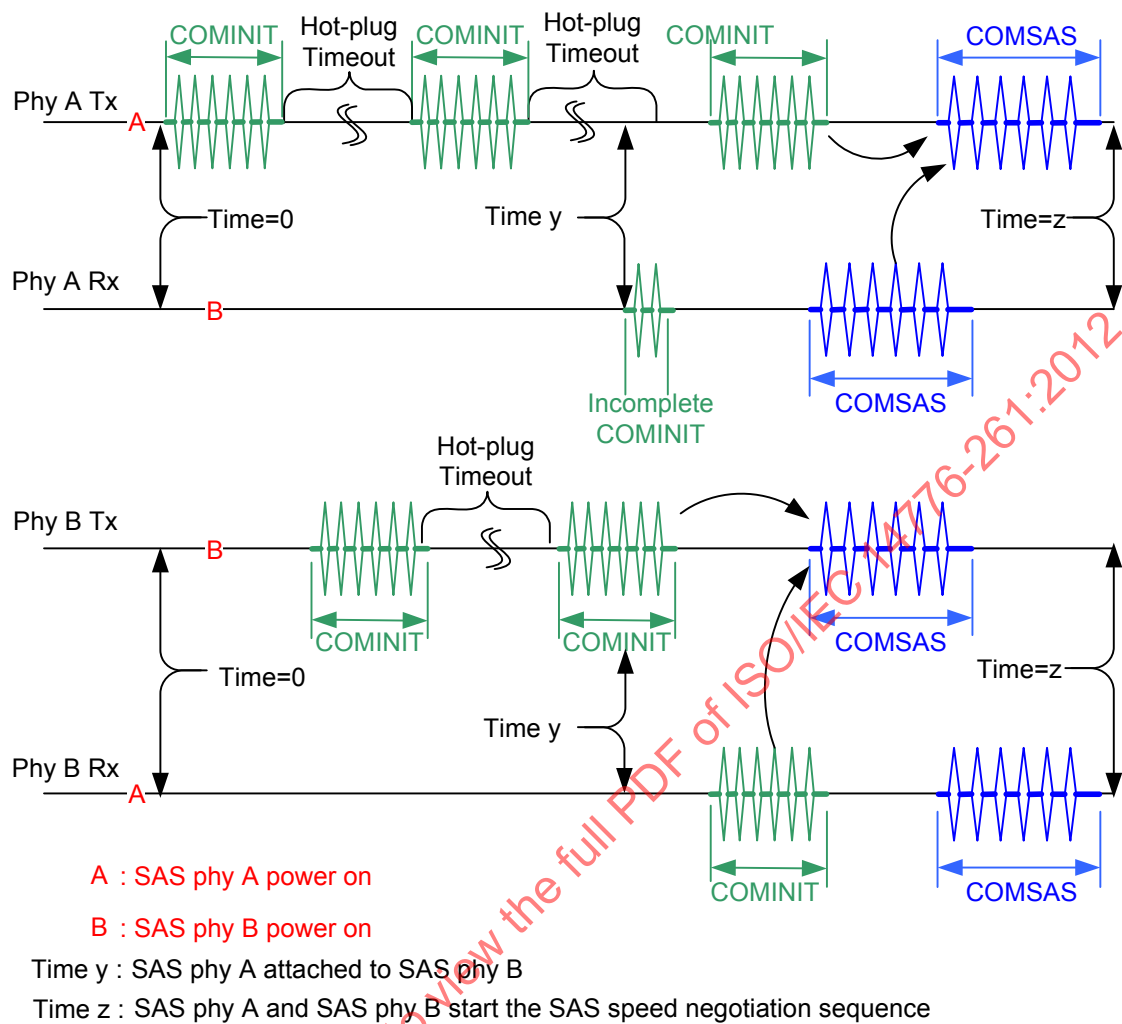


Figure 76 — Hot-plug and the phy reset sequence

5.8 Phy power condition sequences

5.8.1 Transitioning from the active phy power condition to a low phy power condition

See 6.11 for the sequence to transition from the active phy power condition to a low phy power condition.

5.8.2 Transitioning from a low phy power condition to the active phy power condition

Figure 77 shows the sequence to transition from a low phy power condition to the active phy power condition.

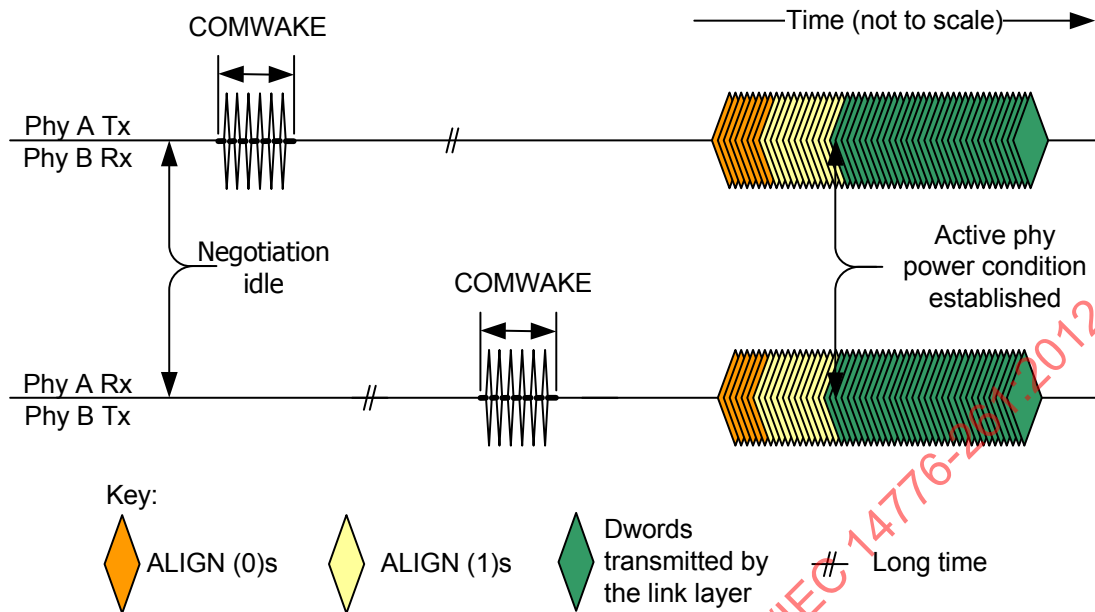


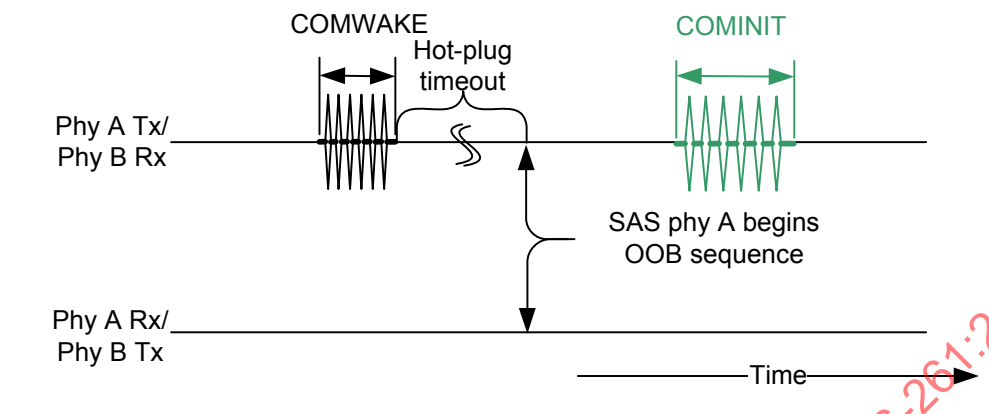
Figure 77 — Transition to active phy power condition

5.8.3 Events during low phy power condition

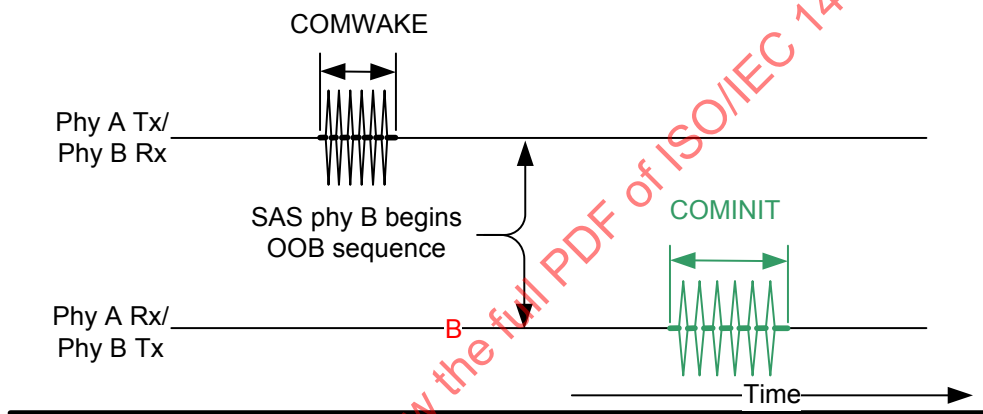
Figure 78 shows examples of responses to the following events that may occur during the transition from a low phy power condition to the active phy power condition:

- no response to COMWAKE within a hot plug timeout; and
- power on occurs after COMWAKE and before the hot plug timeout.

Sequence 1: no response to COMWAKE within a hot plug timeout



Sequence 2: power on occurs after COMWAKE and before the hot plug timeout



B : SAS phy B power on

Figure 78 — Hot plug and low phy power condition

The sequence for a no response to COMWAKE within a hot plug timeout depicted in sequence 1 in figure 78 proceeds as follows:

- 1) phy A transmits COMWAKE;
- 2) phy A detects no COMWAKE within a hot plug timeout; and
- 3) phy A transmits an OOB sequence.

The sequence for a power on occurs after COMWAKE and before the hot plug timeout depicted in sequence 2 in figure 78 proceeds as follows:

- 1) phy A transmits COMWAKE;
- 2) phy A detects an OOB sequence from phy B within a hot plug timeout; and
- 3) phy A transmits an OOB sequence.

5.9 SP (phy layer) state machine

5.9.1 SP state machine overview

The SP state machine controls the phy reset sequence. This state machine consists of the following sets of states:

- a) OOB sequence (OOB) states;
- b) SAS speed negotiation (SAS) states;
- c) SAS phy power condition (PS) states; and
- d) SATA host emulation (SATA) states.

This state machine consists of the following states:

- a) SP0:OOB_COMINIT (see 5.9.3.2) (initial state);
- b) SP1:OOB_AwaitCOMX (see 5.9.3.3);
- c) SP2:OOB_NoCOMSASTimeout (see 5.9.3.4);
- d) SP3:OOB_AwaitCOMINIT_Sent (see 5.9.3.5);
- e) SP4:OOB_COMSAS (see 5.9.3.6);
- f) SP5:OOB_AwaitCOMSAS_Sent (see 5.9.3.7);
- g) SP6:OOB_AwaitNoCOMSAS (see 5.9.3.8);
- h) SP7:OOB_AwaitCOMSAS (see 5.9.3.9);
- i) SP8:SAS_Start (see 5.9.4.3);
- j) SP9:SAS_WindowNotSupported (see 5.9.4.4);
- k) SP10:SAS_AwaitALIGN (see 5.9.4.5);
- l) SP11:SAS_AwaitALIGN1 (see 5.9.4.6);
- m) SP12:SAS_AwaitSNW (see 5.9.4.7);
- n) SP13:SAS_Pass (see 5.9.4.8);
- o) SP14:SAS_Fail (see 5.9.4.9);
- p) SP15:SAS_PHY_Ready (see 5.9.4.10);
- q) SP16:SATA_COMWAKE (see 5.9.6.2);
- r) SP17:SATA_AwaitCOMWAKE (see 5.9.6.3);
- s) SP18:SATA_AwaitNoCOMWAKE (see 5.9.6.4);
- t) SP19:SATA_AwaitALIGN (see 5.9.6.5);
- u) SP20:SATA_AdjustSpeed (see 5.9.6.6);
- v) SP21:SATA_TransmitALIGN (see 5.9.6.7);
- w) SP22:SATA_PHY_Ready (see 5.9.6.8);
- x) SP23:SATA_PM_Partial (see 5.9.6.9);
- y) SP24:SATA_PM_Slumber (see 5.9.6.10);
- z) SP25:SATA_PortSel (see 5.9.7.2);
- aa) SP26:SATA_SpinupHold (see 5.9.8.2);
- ab) SP27:SAS_Settings (see 5.9.4.11);
- ac) SP28:SAS_TrainSetup (see 5.9.4.12);
- ad) SP29:SAS_Train (see 5.9.4.13);
- ae) SP30:SAS_TrainingDone (see 5.9.4.14);
- af) SP31:SAS_PS_Low_Phy_Power state (see 5.9.5.2);
- ag) SP32:SAS_PS_ALIGN0 state (see 5.9.5.3); and
- ah) SP33:SAS_PS_ALIGN1 state (see 5.9.5.4).

This state machine receives the following requests from the management application layer or a layer outside the scope of this standard:

- a) Management Reset from the management application layer;
- b) Disable Phy from the management application layer;
- c) Transmit SATA Port Selection Signal from the management application layer;
- d) Enter Partial request from SATA link layer, if any;
- e) Enter Slumber from the SATA link layer, if any; and
- f) Exit from the SATA link layer, if any.

This state machine shall start in the SP0:OOB_COMINIT state after:

- a) a power on;
- b) a hard reset;
- c) receiving a Management Reset request (e.g., as a result of processing an SMP PHY CONTROL function requesting a phy operation of LINK RESET or HARD RESET); or
- d) receiving a Disable Phy request (e.g., as a result of processing an SMP PHY CONTROL function requesting a phy operation of DISABLE).

This state machine shall maintain a MgmtReset state machine variable to determine whether a Management Reset request has been received. Any SP state that receives a Management Reset request shall set the MgmtReset state machine variable to one before making a transition to the SP0:OOB_COMINIT state (see 5.9.3.2). Any SP state that receives a power on or a hard reset shall set the MgmtReset state machine variable to zero before making a transition to the SP0:OOB_COMINIT state.

This state machine shall maintain a Current SNW state machine variable to determine the current SNW (e.g., SNW-1, SNW-2, SNW-3, Final-SNW, Train-SNW, or Unsupported Phy Attached).

If the phy status is available through any of the following:

- a) the SMP DISCOVER response (see 9.4.3.10);
- b) the SMP DISCOVER LIST response (see 9.4.3.15);
- c) the Phy Control And Discover mode page (see 9.2.7.5); or
- d) the Protocol-Specific Port log page (see 9.2.8.1),

then this state machine shall maintain a ResetStatus state machine variable to determine the NEGOTIATED PHYSICAL LINK RATE field and/or the NEGOTIATED LOGICAL LINK RATE field.

If the phy supports SNW-3, then this state machine shall maintain a Commonly Supported Settings state machine variable that contains supported settings common between the phy and the attached phy.

If the phy supports SATA port selectors, then this state machine shall:

- a) maintain a COMWAKE_Received state machine variable to indicate whether a COMWAKE Detected message was received in the SP0:OOB_COMINIT state or the SP1:OOB_AwaitCOMX state since the last time the SP0:OOB_COMINIT state was entered; and
- b) transition to the SP25:SATA_PortSel state whenever it receives a Transmit SATA Port Selection Signal request.

If the phy supports low phy power conditions, then this state machine shall maintain a SASPhyPwrCond state machine variable to determine:

- a) the current phy power condition (see 4.10 and 5.6.5); and
- b) the PHY POWER CONDITION field in the DISCOVER response (see table 236).

This state machine sends the following messages to the SP_DWS state machine (see 5.10):

- a) Start DWS; and
- b) Stop DWS.

This state machine receives the following messages from the SP_DWS state machine:

- a) DWS Lost; and
- b) DWS Reset.

This state machine shall maintain the timers listed in table 55.

Table 55 — SP state machine timers

Timer	Initial value
COMSAS Detect Timeout timer	COMSAS detect timeout (see SAS-2.1)
Await ALIGN Timeout timer	Await ALIGN timeout (see table 45 in 5.7.2.2)
Hot-Plug Timeout timer	Hot plug timeout (see table 44 in 5.7.1)
RCDT timer	RCDT (see table 46 in 5.7.4.2)
SNLT timer	SNLT (see table 46 in 5.7.4.2)
SNTT timer	SNTT (see table 46 in 5.7.4.2)
TLT timer	TLT (see table 46 in 5.7.4.2)
MTT timer	MTT (see table 46 in 5.7.4.2)

5.9.2 SP transmitter and receiver

The SP transmitter transmits OOB signals and dwords on the physical link based on messages from the SP state machine (see 5.9).

The SP transmitter receives the following messages from the SP state machine:

- a) Transmit COMINIT;
- b) Transmit COMSAS;
- c) Transmit COMWAKE;
- d) Transmit SATA Port Selection Signal;
- e) Transmit D10.2;
- f) Set Physical Link Rate with an argument specifying the physical link rate (e.g., 1.5 Gbps, 3 Gbps, or 6 Gbps);
- g) Set SSC with an argument of On or Off;
- h) Transmit ALIGN with an argument indicating the specific type (e.g., Transmit ALIGN (0));
- i) Transmit Phy Capabilities Bits;
- j) Transmit TRAIN Pattern;
- k) Transmit TRAIN_DONE Pattern;
- l) Transmit MUX Sequence;
- m) Transmit OOB Idle;
- n) Enter Partial Phy Power Condition; and
- o) Enter Slumber Phy Power Condition.

When not otherwise instructed, the SP transmitter transmits negotiation idle.

Upon receiving a Transmit MUX Sequence message, the SP transmitter transmits:

- 1) MUX (LOGICAL LINK 0);
- 2) MUX (LOGICAL LINK 1);
- 3) MUX (LOGICAL LINK 0);
- 4) MUX (LOGICAL LINK 1);
- 5) MUX (LOGICAL LINK 0); and
- 6) MUX (LOGICAL LINK 1).

Upon receiving a Transmit OOB Idle message, the SP transmitter transmits OOB_IDLE or D.C. idle as defined in SAS-2.1.

The SP transmitter shall complete any physical link rate change and SSC change requested with the Set Physical Link Rate message and the Set SSC message within RCDT (see table 46 in 5.7.4.2).

The SP transmitter sends the following messages to the SP state machine:

- a) COMINIT Transmitted;
- b) COMSAS Transmitted;

- c) COMWAKE Transmitted;
- d) SATA Port Selection Signal Transmitted;
- e) TRAIN_DONE Pattern Transmitted;
- f) Phy Capabilities Bits Transmitted; and
- g) MUX Sequence Transmitted.

The SP receiver receives the following messages from the SP state machine:

- a) Set Physical Link Rate with an argument specifying the physical link rate (e.g., 1.5 Gbps, 3 Gbps, or 6 Gbps);
- b) Receive Phy Capabilities Bits;
- c) Start Training;
- d) Abort Training;
- e) Enter Partial Phy Power Condition; and
- f) Enter Slumber Phy Power Condition.

The SP receiver receives the following messages from the SP_DWS state machine (see 5.10):

- a) Sync Acquired; and
- b) Sync Lost.

The SP receiver sends the following messages to the SP state machine indicating OOB signals and dwords received from the physical link:

- a) COMINIT Detected;
- b) COMSAS Detected;
- c) COMWAKE Detected;
- d) COMSAS Completed;
- e) COMWAKE Completed;
- f) ALIGN Received with an argument indicating the specific type (e.g., ALIGN Received (0));
- g) Phy Capabilities Bits Received with arguments indicating the supported settings bits received and either Good Parity or Bad Parity;
- h) Training Completed;
- i) TRAIN_DONE Received; and
- j) Dword Received.

The ALIGN Received, Dword Received, and TRAIN_DONE Received messages are only sent after the SP_DWS state machine has achieved dword synchronization.

For SATA speed negotiation, the ALIGN Received (0) message includes an argument containing the physical link rate at which the ALIGN (0) primitives were detected. For SAS speed negotiation, only ALIGNs at the physical link rate specified by the last Set Physical Link Rate message received by the SP receiver cause ALIGN Received messages.

The SP transmitter relationship to other transmitters is defined in 4.3.2. The SP receiver relationship to other receivers is defined in 4.3.3.

5.9.3 OOB sequence states

5.9.3.1 OOB sequence states overview

Figure 79 shows the OOB sequence states. These states are indicated by state names with a prefix of OOB.

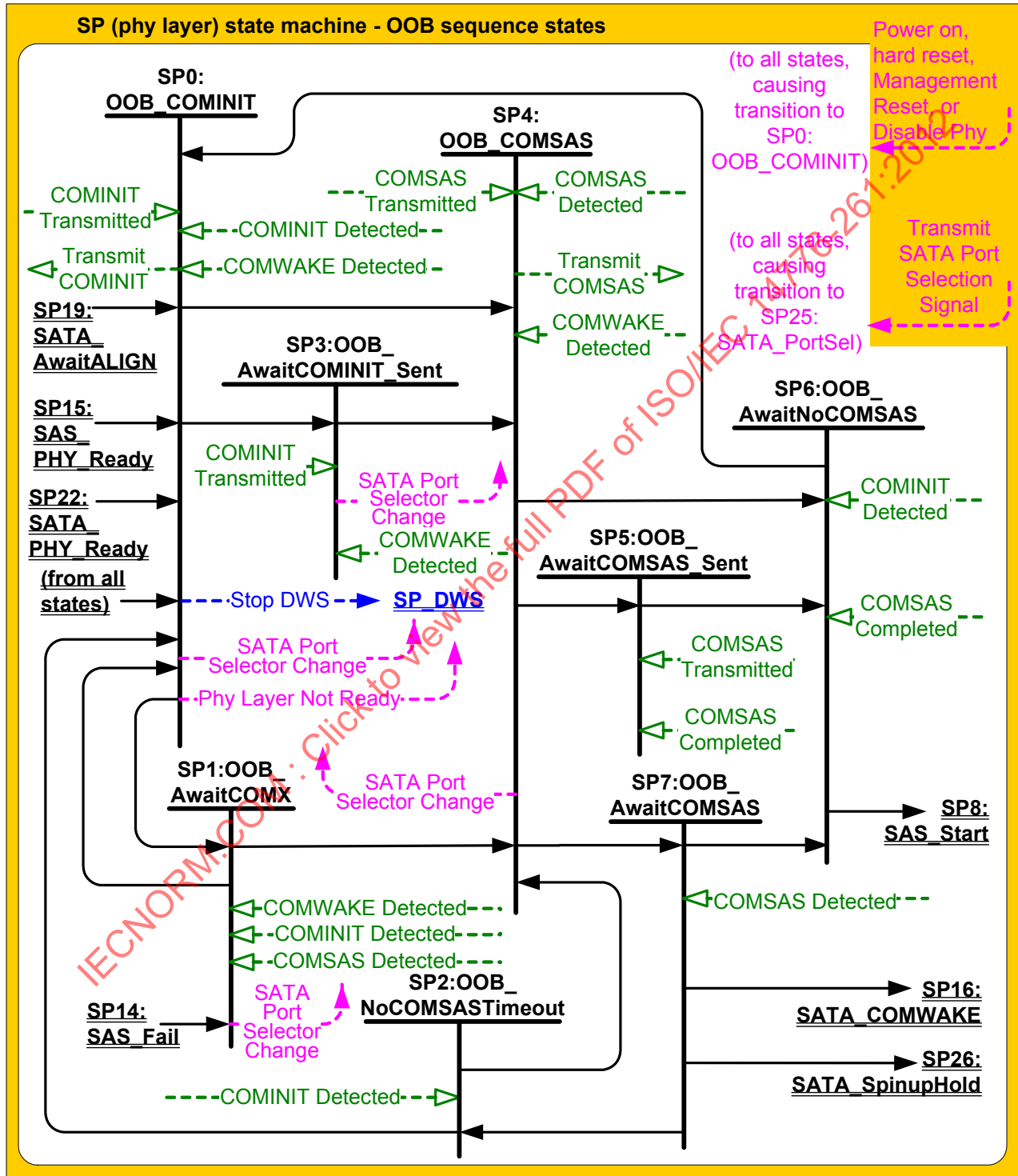


Figure 79 — SP (phy layer) state machine - OOB sequence states

5.9.3.2 SP0:OOB_COMINIT state

5.9.3.2.1 State description

This state is the initial state for this state machine.

Upon entry into this state, the phy shall:

- a) set the COMWAKE_Received state machine variable to zero;
- b) set the SASPhyPwrCond state machine variable to Active;
- c) send a Stop DWS message to the SP_DWS state machine;
- d) send a Phy Layer Not Ready confirmation to the link layer;
- e) set the ATTACHED SATA DEVICE bit to zero in the SMP DISCOVER response (see 9.4.3.10);
- f) if this state was entered due to power on, and the phy is an SMP target phy, then set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response (see 9.4.3.10);
- g) if this state was not entered because of a Disable Phy request and D.C. mode is enabled, then send a Transmit COMINIT message to the SP transmitter; and
- h) if this state was not entered because of a Disable Phy request and optical mode is enabled, then:
 - 1) transmit OOB idle for an RCDT time; and
 - 2) send a transmit COMWAKE message to the SP transmitter.

If this state was entered because of a Disable Phy request, then upon entry into this state, this state shall:

- a) ignore COMINIT Detected messages until this state is re-entered due to a power on, hard reset, or Management Reset request; and
- b) set the ResetStatus state machine variable to DISABLED.

If this state was entered due to power on or hard reset, then upon entry into this state, this state shall set the ResetStatus state machine variable to UNKNOWN.

If this state was entered because of a Management Reset request, then upon entry into this state, this state shall:

- a) if the ResetStatus state machine variable is not set to RESET_IN_PROGRESS, SPINUP_HOLD, G1, G2, or G3, then set the ResetStatus state machine variable to UNKNOWN; or
- b) if the ResetStatus state machine variable is set to RESET_IN_PROGRESS, SPINUP_HOLD, G1, G2, or G3, then set the ResetStatus state machine variable to RESET_IN_PROGRESS.

If this state was not entered due to a power on, hard reset, Disable Phy, or Management Reset request, then upon entry into this state, this state shall:

- a) if the ResetStatus state machine variable is not set to PHY_RESET_PROBLEM, SPINUP_HOLD, or UNSUPPORTED_PHY_ATTACHED, then set the ResetStatus state machine variable to UNKNOWN; or
- b) if the ResetStatus state machine variable is set to PHY_RESET_PROBLEM, SPINUP_HOLD, or UNSUPPORTED_PHY_ATTACHED, then not change the ResetStatus state machine variable.

If the phy supports SATA port selectors and this state receives a COMWAKE Detected message, then this state shall:

- a) if the ResetStatus state machine variable is not set to PHY_RESET_PROBLEM, SPINUP_HOLD, or UNSUPPORTED_PHY_ATTACHED, then set the ResetStatus state machine variable to PORT_SELECTOR;
- b) set the COMWAKE_Received state machine variable to one; and
- c) if the phy is an SMP target phy and the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 9.4.3.10), then:
 - A) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response; and
 - B) send a SATA Port Selector Change confirmation to the link layer.

This state machine waits for receipt of a COMINIT Transmitted message and/or a COMINIT Detected message.

5.9.3.2.2 Transition SP0:OOB_COMINIT to SP1:OOB_AwaitCOMX

This transition shall occur if this state receives a COMINIT Transmitted message and has not received a COMINIT Detected message.

5.9.3.2.3 Transition SP0:OOB_COMINIT to SP3:OOB_AwaitCOMINIT_Sent

This transition shall occur if this state receives a COMINIT Detected message and has not received a COMINIT Transmitted message.

5.9.3.2.4 Transition SP0:OOB_COMINIT to SP4:OOB_COMSAS

This transition shall occur if this state receives both a COMINIT Transmitted message and a COMINIT Detected message.

5.9.3.3 SP1:OOB_AwaitCOMX state

5.9.3.3.1 State description

Upon entry into this state, the phy shall initialize and start the Hot-Plug Timeout timer if this phy is:

- a) an expander phy; or
- b) an initiator phy or target phy implementing the Hot-Plug Timeout timer.

If the phy supports SATA port selectors and this state receives a COMWAKE Detected message, then this state shall:

- a) if the ResetStatus state machine variable is not set to PHY_RESET_PROBLEM, SPINUP_HOLD, or UNSUPPORTED_PHY_ATTACHED, then set the ResetStatus state machine variable to PORT_SELECTOR;
- b) set the COMWAKE_Received state machine variable to one; and
- c) if the phy is an SMP target phy and the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 9.4.3.10), then:
 - A) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response; and
 - B) send a SATA Port Selector Change confirmation to the link layer.

5.9.3.3.2 Transition SP1:OOB_AwaitCOMX to SP0:OOB_COMINIT

This transition shall occur if the Hot-Plug Timeout timer expires.

If the COMWAKE_Received state machine variable is set to zero and the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 9.4.3.10), then the state machine shall, before the transition:

- a) set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response; and
- b) send a SATA Port Selector Change confirmation to the link layer.

Before the transition, if this state was entered from SP0:OOB_COMINIT, then this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.3.3.3 Transition SP1:OOB_AwaitCOMX to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Detected message or a COMSAS Detected message. If COMSAS Detected was received, then this transition shall include a COMSAS Detected argument.

5.9.3.4 SP2:OOB_NoCOMSASTimeout state

5.9.3.4.1 State description

Upon entry into this state, the phy shall initialize and start the Hot-Plug Timeout timer if this phy is:

- a) this phy is an expander phy; or

- b) this phy is an initiator phy or target phy implementing the Hot-Plug Timeout timer.

5.9.3.4.2 Transition SP2:OOB_NoCOMSASTimeout to SP0:OOB_COMINIT

This transition shall occur if the Hot-Plug Timeout timer expires.

5.9.3.4.3 Transition SP2:OOB_NoCOMSASTimeout to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Detected message.

5.9.3.5 SP3:OOB_AwaitCOMINIT_Sent state

5.9.3.5.1 State description

This state waits for a COMINIT Transmitted message.

If the phy supports SATA port selectors and this state receives a COMWAKE Detected message, then this state shall:

- a) if the ResetStatus state machine variable is not set to PHY_RESET_PROBLEM, SPINUP_HOLD, or UNSUPPORTED_PHY_ATTACHED, then set the ResetStatus state machine variable to PORT_SELECTOR; and
- b) if the phy is an SMP target phy and the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 9.4.3.10), then:
 - A) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response; and
 - B) send a SATA Port Selector Change confirmation to the link layer.

5.9.3.5.2 Transition SP3:OOB_AwaitCOMINIT_Sent to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Transmitted message.

5.9.3.6 SP4:OOB_COMSAS state

5.9.3.6.1 State description

Upon entry into this state, the phy shall send a Transmit COMSAS message to the SP transmitter.

This state waits for receipt of a COMSAS Transmitted message and/or a COMSAS Detected message.

If the phy supports SATA port selectors and this state receives a COMWAKE Detected message, then this state shall:

- a) if the ResetStatus state machine variable is not set to PHY_RESET_PROBLEM, SPINUP_HOLD, or UNSUPPORTED_PHY_ATTACHED, then set the ResetStatus state machine variable to PORT_SELECTOR; and
- b) if the phy is an SMP target phy and the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 9.4.3.10), then:
 - A) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response; and
 - B) send a SATA Port Selector Change confirmation to the link layer.

5.9.3.6.2 Transition SP4:OOB_COMSAS to SP5:OOB_AwaitCOMSAS_Sent

This transition shall occur if this state receives a COMSAS Detected message or this state was entered with a COMSAS Detected argument, and this state has not received a COMSAS Transmitted message.

If the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 9.4.3.10), then the state machine shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response and send a SATA Port Selector Change confirmation to the link layer before the transition.

5.9.3.6.3 Transition SP4:OOB_COMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur if this state receives both a COMSAS Transmitted message and a COMSAS Detected message.

If the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 9.4.3.10), then the state machine shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response and send a SATA Port Selector Change confirmation to the link layer before the transition.

5.9.3.6.4 Transition SP4:OOB_COMSAS to SP7:OOB_AwaitCOMSAS

This transition shall occur if this state receives a COMSAS Transmitted message and has not received a COMSAS Detected message.

5.9.3.7 SP5:OOB_AwaitCOMSAS_Sent state

5.9.3.7.1 State description

This state waits for receipt of a COMSAS Transmitted message.

5.9.3.7.2 Transition SP5:OOB_AwaitCOMSAS_Sent to SP6:OOB_AwaitNoCOMSAS

This transition shall occur after receiving a COMSAS Transmitted message.

If this state received a COMSAS Completed message, then it shall include a COMSAS Completed argument with the transition.

5.9.3.8 SP6:OOB_AwaitNoCOMSAS state

5.9.3.8.1 State description

This state machine waits for a COMSAS Completed message, which indicates that COMSAS has been received.

5.9.3.8.2 Transition SP6:OOB_AwaitNoCOMSAS to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.3.8.3 Transition SP6:OOB_AwaitNoCOMSAS to SP8:SAS_Start

This transition shall occur after receiving a COMSAS Completed message, or shall occur if a COMSAS Completed argument was received in the transition.

5.9.3.9 SP7:OOB_AwaitCOMSAS state

5.9.3.9.1 State description

Upon entry into this state, the phy shall initialize and start the COMSAS Detect Timeout timer.

5.9.3.9.2 Transition SP7:OOB_AwaitCOMSAS to SP2:OOB_NoCOMSASTimeout

This transition shall occur if the phy does not support SATA and the COMSAS Detect Timeout timer expires.

The state machine shall set the MgmtReset state machine variable to zero before the transition.

5.9.3.9.3 Transition SP7:OOB_AwaitCOMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur after receiving a COMSAS Detected message.

The state machine shall set the MgmtReset state machine variable to zero before the transition.

The state machine shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response (see 9.4.3.10). If the ATTACHED SATA PORT SELECTOR bit in the SMP DISCOVER response was set to one before to this transition, the state machine shall send a SATA Port Selector Change confirmation to the link layer before the transition.

5.9.3.9.4 Transition SP7:OOB_AwaitCOMSAS to SP16:SATA_COMWAKE

This transition shall occur if:

- a) the phy supports SATA;
- b) the COMSAS Detect Timeout timer expires; and
 - A) the MgmtReset state machine variable is set to one; or
 - B) the phy does not implement SATA spinup hold.

The state machine shall set the MgmtReset state machine variable to zero before the transition.

The state machine shall set the ATTACHED SATA DEVICE bit to one in the SMP DISCOVER response (see 9.4.3.10) before the transition.

5.9.3.9.5 Transition SP7:OOB_AwaitCOMSAS to SP26:SATA_SpinupHold

This transition shall occur if:

- a) the phy supports SATA;
- b) the COMSAS Detect Timeout timer expires;
- c) the phy implements SATA spinup hold; and
- d) the MgmtReset state machine variable is set to zero.

The state machine shall set the ATTACHED SATA DEVICE bit to one in the SMP DISCOVER response (see 9.4.3.10) before the transition.

5.9.4 SAS speed negotiation states

5.9.4.1 SAS speed negotiation states overview

Figure 80 shows the SAS speed negotiation states, in which the phy has detected that it is attached to a SAS phy or expander phy rather than a SATA phy, and performs the SAS speed negotiation sequence. These states are indicated by state names with a prefix of SAS.

5.9.4.2 Negotiation idle

SAS speed negotiation states use negotiation idle at the beginning of each SNW.

If D.C. mode is enabled, then a negotiation idle consists of the transmission of D.C. idle.

If optical mode is enabled, then a negotiation idle consists of repetitions of the following steps:

- 1) transmission of six OOB_IDLEs with either starting disparity at a vendor specific bit rate of at least 1.5 Gbps; and
- 2) transmission of up to 512 data dwords set to 00000000h that are transmitted scrambled and 8b10b encoded at a vendor specific bit rate of at least 1.5 Gbps.

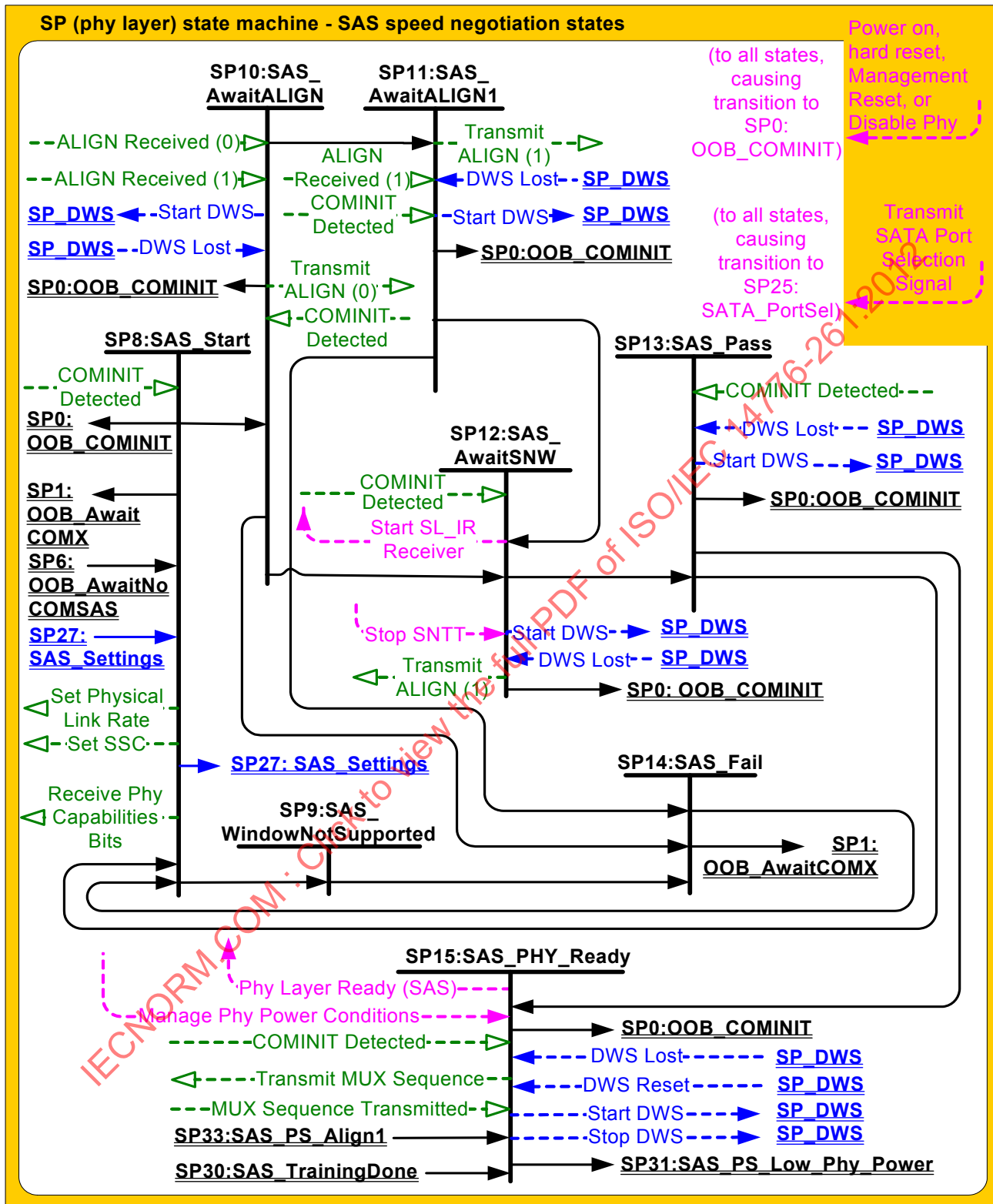


Figure 80 — SP (phy layer) state machine - SAS speed negotiation states

Figure 81 shows the SAS speed negotiation states related to SNW-3 and Train-SNW.

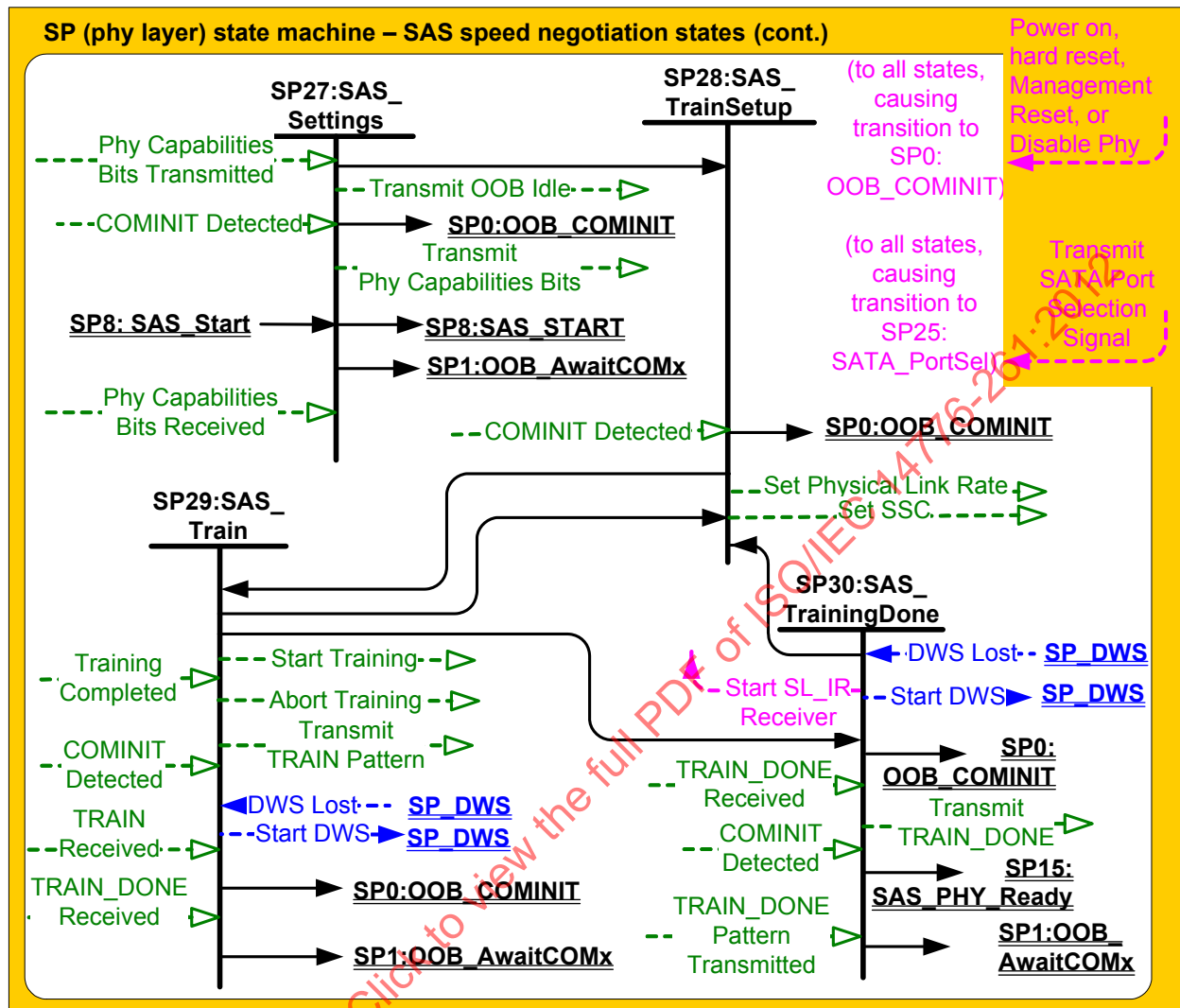


Figure 81 — SP (phy layer) state machine - SAS speed negotiation states for SNW-3 and Train-SNW

5.9.4.3 SP8:SAS_Start state

5.9.4.3.1 State description

This is the state in which the SAS speed negotiation sequence begins.

Upon entry into this state, the phy shall initialize and start the RCDT timer.

If this state is entered from SP6:OOB_AwaitNoCOMSAS, then the Current SNW state machine variable shall be set to SNW-1. If this state is not entered from SP6:OOB_AwaitNoCOMSAS, then the Current SNW state machine variable shall be set to:

- a) SNW-2 if the Current SNW state machine variable is set to SNW-1;
- b) SNW-3 if the Current SNW state machine variable is set to SNW-2, and either SNW-1 is invalid or SNW-2 is valid;
- c) Final-SNW if the Current SNW state machine variable is set to SNW-2, SNW-1 is valid, and SNW-2 is invalid;
- d) Final-SNW if the Current SNW state machine variable is set to SNW-3, SNW-3 is invalid, and SNW-2 is valid; or

- e) Unsupported Phy Attached if the Current SNW state machine variable is set to SNW-3, SNW-3 is invalid, and SNW-2 is invalid.

After the Current SNW state machine variable is updated, if:

- a) the Current SNW state machine variable is not set to Unsupported Phy Attached; and
b) the SNW specified by the Current SNW state machine variable is supported,

then this state shall send the messages specified in table 56 to the SP transmitter and SP receiver:

Table 56 — Messages to SP transmitter and SP receiver at start of RCDT

Current SNW state machine variable	Other conditions	Message(s) sent to SP transmitter	Message(s) sent to SP receiver
SNW-1	none	Set Physical Link Rate (1.5 Gbps) and Set SSC (Off)	Set Physical Link Rate (1.5 Gbps)
SNW-2	none	Set Physical Link Rate (3 Gbps) and Set SSC (Off)	Set Physical Link Rate (3 Gbps)
SNW-3	none	Set SSC (On) or Set SSC (Off)	Receive Phy Capabilities Bits
Final-SNW	SNW-1 was valid and SNW-2 was invalid	Set Physical Link Rate (1.5 Gbps) and Set SSC (Off)	Set Physical Link Rate (1.5 Gbps)
	SNW-2 was valid	Set Physical Link Rate (3 Gbps) and Set SSC (Off)	Set Physical Link Rate (3 Gbps)

5.9.4.3.2 Transition SP8:SAS_Start to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.3.3 Transition SP8:SAS_Start to SP1:OOB_AwaitCOMX

This transition shall occur if the Current SNW state machine variable is set to Unsupported Phy Attached.

Before the transition, this state shall set the ResetStatus state machine variable to UNSUPPORTED_PHY_ATTACHED.

5.9.4.3.4 Transition SP8:SAS_Start to SP9:SAS_WindowNotSupported

This transition shall occur after the RCDT timer expires if the SNW indicated by the Current SNW state machine variable is not supported.

5.9.4.3.5 Transition SP8:SAS_Start to SP10:SAS_AwaitALIGN

This transition shall occur after the RCDT timer expires if:

- a) the Current SNW state machine variable is not set to SNW-3; and
b) the SNW indicated by the Current SNW state machine variable is supported.

5.9.4.3.6 Transition SP8:SAS_Start to SP27:SAS_Settings

This transition shall occur after the RCDT timer expires if:

- a) the Current SNW state machine variable is set to SNW-3; and

- b) SNW-3 is supported.

5.9.4.4 SP9:SAS_WindowNotSupported state

5.9.4.4.1 State description

Upon entry into this state, the phy shall initialize and start the SNTT timer.

5.9.4.4.2 Transition SP9:SAS_WindowNotSupported to SP0:OOB_COMINIT

This transition should occur after receiving:

- a) a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.4.3 Transition SP9:SAS_WindowNotSupported to SP14:SAS_Fail

This transition shall occur after the SNTT timer expires.

5.9.4.5 SP10:SAS_AwaitALIGN state

5.9.4.5.1 State description

Upon entry into this state, the phy shall:

- a) initialize and start the SNTT timer and SNLT timer;
- b) send a Start DWS message to the SP_DWS state machine; and
- c) repeatedly send Transmit ALIGN (0) messages to the SP transmitter.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

5.9.4.5.2 Transition SP10:SAS_AwaitALIGN to SP0:OOB_COMINIT

This transition shall occur after receiving:

- a) a DWS Lost message, if this state does not send a Start DWS message; or
- b) a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.5.3 Transition SP10:SAS_AwaitALIGN to SP11:SAS_AwaitALIGN1

This transition shall occur if this state receives an ALIGN Received (0) message before the SNLT timer expires.

5.9.4.5.4 Transition SP10:SAS_AwaitALIGN to SP12:SAS_AwaitSNW

This transition shall occur if this state receives an ALIGN Received (1) message before the SNLT timer expires.

5.9.4.5.5 Transition SP10:SAS_AwaitALIGN to SP14:SAS_Fail

This transition shall occur if the SNTT timer expires.

5.9.4.6 SP11:SAS_AwaitALIGN1 state

5.9.4.6.1 State description

This state shall repeatedly send Transmit ALIGN (1) messages to the SP transmitter.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

5.9.4.6.2 Transition SP11:SAS_AwaitALIGN1 to SP0:OOB_COMINIT

This transition shall occur after receiving:

- a) a DWS Lost message, if this state does not send a Start DWS message; or
- b) a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.6.3 Transition SP11:SAS_AwaitALIGN1 to SP12:SAS_AwaitSNW

This transition shall occur if this state receives an ALIGN Received (1) message before the SNTT timer expires. This indicates that the attached phy has been able to achieve dword synchronization in the current SNW.

5.9.4.6.4 Transition SP11:SAS_AwaitALIGN1 to SP14:SAS_Fail

This transition shall occur if the SNTT timer expires. This indicates that the attached phy has not been able to achieve dword synchronization in the current SNW.

5.9.4.7 SP12:SAS_AwaitSNW state

5.9.4.7.1 State description

This state shall repeatedly send Transmit ALIGN (1) messages to the SP transmitter.

If the Current SNW state machine variable is set to Final-SNW, then this state shall send a Start SL_IR Receiver confirmation to the link layer.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

This state waits for the SNTT timer to expire or for a Stop SNTT request.

If this state receives a Stop SNTT request, then this state shall stop the SNTT timer.

5.9.4.7.2 Transition SP12:SAS_AwaitSNW to SP0:OOB_COMINIT

This transition shall occur after receiving:

- a) a DWS Lost message, if this state does not send a Start DWS message; or
- b) a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.7.3 Transition SP12:SAS_AwaitSNW to SP13:SAS_Pass

This transition shall occur after the SNTT timer expires or after receiving a Stop SNTT request.

5.9.4.8 SP13:SAS_Pass state

5.9.4.8.1 State description

This state determines if:

- a) another SAS SNW is required; or
- b) the SAS speed negotiation sequence is complete.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

5.9.4.8.2 Transition SP13:SAS_Pass to SP0:OOB_COMINIT

This transition shall occur after receiving:

- a) a DWS Lost message, if this state does not send a Start DWS message; or
- b) a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.8.3 Transition SP13:SAS_Pass to SP8:SAS_Start

This transition shall occur if the Current SNW state machine variable is not set to Final-SNW.

5.9.4.8.4 Transition SP13:SAS_Pass to SP15:SAS_PHY_Ready

This transition shall occur if the Current SNW state machine variable is set to Final-SNW.

5.9.4.9 SP14:SAS_Fail state**5.9.4.9.1 State description**

This state determines if:

- a) another SAS SNW is required; or
- b) the SAS speed negotiation sequence is complete.

5.9.4.9.2 Transition SP14:SAS_Fail to SP1:OOB_AwaitCOMX

This transition shall occur if the Current SNW state machine variable is set to Final-SNW.

Before the transition, this state shall set the ResetStatus state machine variable to PHY_RESET_PROBLEM.

5.9.4.9.3 Transition SP14:SAS_Fail to SP8:SAS_Start

This transition shall occur if the Current SNW state machine variable is not set to Final-SNW.

5.9.4.10 SP15:SAS_PHY_Ready state**5.9.4.10.1 State description**

This state waits for:

- a) a COMINIT Detected message;
- b) a DWS Lost message;
- c) a DWS Reset message; or
- d) a Manage Phy Power Conditions request.

Upon entry into this state, the phy shall:

- a) if multiplexing is enabled (see table 51 in 5.7.4.2.3.3), then:
 - 1) send a Transmit MUX Sequence message to the SP transmitter; and
 - 2) after receiving MUX Sequence Transmitted, send a Phy Layer Ready (SAS) confirmation to the link layer;
- b) if multiplexing is not enabled, then send a Phy Layer Ready (SAS) confirmation to the link layer;
- c) if the SP transmitter is transmitting at 1.5 Gbps, then set the ResetStatus state machine variable to G1;
- d) if the SP transmitter is transmitting at 3 Gbps, then set the ResetStatus state machine variable to G2; and
- e) if the SP transmitter is transmitting at 6 Gbps, then set the ResetStatus state machine variable to G3.

While in this state, dwords from the link layer are transmitted at the negotiated physical link rate (i.e., the rate established in the previous SNW).

If multiplexing is disabled, then each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

NOTE 21 If multiplexing is enabled and this state receives a DWS Lost message, then this state does not send a Start DWS message and the state machine transitions to SP0:OOB_COMINIT.

If a Manage Phy Power Conditions (Stop DWS) request is received, then this state shall send a Stop DWS message to the SP_DWS state machine.

5.9.4.10.2 Transition SP15:SAS_PHY_Ready to SP0:OOB_COMINIT

If optical mode is enabled or D.C. mode is enabled, then this transition shall occur after receiving:

- a) a DWS Lost message, if this state does not send a Start DWS message;
- b) a DWS Lost message followed by a COMINIT Detected message, if this state does not send a Start DWS message; or
- c) a DWS Reset message.

If optical mode is enabled, then this transition shall occur after:

- a) receiving a COMINIT Detected message before receiving a DWS Lost message; or
- b) receiving a COMINIT Detected message after sending a Start DWS message

5.9.4.10.3 Transition SP15:SAS_PHY_Ready to SP31:SAS_PS_Low_Phy_Power

This transition shall occur after this state receives a Manage Phy Power Conditions (Enter Partial) request or a Manage Phy Power Conditions (Enter Slumber) request.

If this transition is the result of this state receiving a Manage Phy Power Conditions (Enter Partial) request, then the transition shall include a Partial argument.

If this transition is the result of this state receiving a Manage Phy Power Conditions (Enter Slumber) request, then the transition shall include a Slumber argument.

5.9.4.11 SP27:SAS_Settings state

5.9.4.11.1 State description

This state transmits and receives phy capabilities bits.

Upon entry to this state, the phy shall:

- a) initialize and start the SNTT timer;
- b) set the Commonly Supported Settings state machine variable to indicate that there are no commonly supported settings; and
- c) send a Transmit Phy Capabilities Bits message to the SP transmitter.

If a Phy Capabilities Bits Received message is received with the argument of Good Parity, then this state shall set the Commonly Supported Settings state machine variable to the supported settings that are set to one in both the transmitted and received phy capabilities bits.

After this state receives a Phy Capabilities Bits Transmitted message this state shall request OOB idle be transmitted by repeatedly sending Transmit OOB Idle messages to the SP transmitter.

This state waits for the SNTT timer to expire.

5.9.4.11.2 Transition SP27:SAS_Settings to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.11.3 Transition SP27:SAS_Settings to SP1:OOB_AwaitCOMX

This transition shall occur after the SNTT timer expires if:

- a) a Phy Capabilities Bits Received message is received with an argument of Bad Parity; or
- b) no commonly supported settings exist after the Commonly Supported Settings state machine variable is set as a result of receiving a Phy Capabilities Bits Received message.

Before the transition, this state shall:

- a) if a Phy Capabilities Bits Received message is received with an argument of Bad Parity, then set the ResetStatus state machine variable to PHY_RESET_PROBLEM; or
- b) if no commonly supported settings exist after the Commonly Supported Settings state machine variable is set, then set the ResetStatus state machine variable to UNSUPPORTED_PHY_ATTACHED.

5.9.4.11.4 Transition SP27:SAS_Settings to SP8:SAS_Start

This transition shall occur if:

- a) the SNTT timer expires; and
- b) a Phy Capabilities Bits Received message is not received during this state.

5.9.4.11.5 Transition SP27:SAS_Settings to SP28:SAS_TrainSetup

This transition shall occur:

- a) after the SNTT timer expires; and
- b) if the Commonly Supported Settings state machine variable indicates there is at least one commonly supported setting.

5.9.4.12 SP28:SAS_TrainSetup**5.9.4.12.1 State description**

Upon entry into this state, the phy shall:

- a) initialize and start the RCDT timer; and
- b) send a Set Physical Link Rate message to the SP transmitter and to the SP receiver and send a Set SSC message to the SP transmitter with the arguments set to reflect the highest priority commonly supported setting contained in the Commonly Supported Settings state machine variable.

After the Set Physical Link Rate messages and Set SSC message are sent, the Commonly Supported Settings state machine variable shall be set to indicate that the selected supported settings bit is no longer in common.

5.9.4.12.2 Transition SP28:SAS_TrainSetup to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.12.3 Transition SP28:SAS_TrainSetup to SP29:SAS_Train

This transition shall occur after the RCDT timer expires.

5.9.4.13 SP29:SAS_Train state

5.9.4.13.1 State description

Upon entry into this state, the phy shall:

- a) initialize and start the MTT timer;
- b) initialize and start the TLT timer;
- c) send a Start Training message to the SP receiver; and
- d) send a Start DWS message to the SP_DWS state machine.

This state shall repeatedly send Transmit TRAIN Pattern messages to the SP transmitter.

Each time this state receives a DWS Lost message, this state shall send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization.

If the MTT timer expires, then this state shall send an Abort Training message to the SP receiver.

A phy reset problem occurs if:

- a) the MTT timer expires; and
- b) the Commonly Supported Settings state machine variable does not contain additional commonly supported settings.

5.9.4.13.2 Transition SP29:SAS_Train to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.13.3 Transition SP29:SAS_Train to SP1:OOB_AwaitCOMX

This transition shall occur if a phy reset problem occurs.

Before the transition, this state shall set the ResetStatus state machine variable to PHY_RESET_PROBLEM.

5.9.4.13.4 Transition SP29:SAS_Train to SP28:SAS_TrainSetup

This transition shall occur if:

- a) the MTT timer expires; and
- b) the Commonly Supported Settings state machine variable contains additional commonly supported settings.

5.9.4.13.5 Transition SP29:SAS_Train to SP30:SAS_TrainingDone

This transition shall occur if:

- a) the TLT timer has not expired;
- b) this state receives a Training Completed message;
- c) dword synchronization is acquired; and
- d) this state receives a TRAIN Received message or a TRAIN_DONE Received message.

If a TRAIN_DONE Received message was received, then the transition shall include a TRAIN_DONE Received argument.

Before the transition, this state shall stop the TLT timer.

5.9.4.14 SP30:SAS_TrainingDone state

5.9.4.14.1 State description

This state shall repeatedly send Transmit TRAIN_DONE Pattern messages to the SP transmitter.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

This state waits for the MTT timer to expire or a TRAIN_DONE Received message from the receiver.

If this state receives a TRAIN_DONE Received message, then this state shall stop the MTT timer.

This state shall send a Start SL_IR Receiver confirmation to the link layer when a TRAIN_DONE Received message is received.

A phy reset problem occurs if:

- a) TRAIN_DONE Received message is not received before the MTT timer expires; and
- b) the Commonly Supported Settings state machine variable does not contain additional commonly supported settings.

5.9.4.14.2 Transition SP30:SAS_TrainingDone to SP0:OOB_COMINIT

This transition shall occur after receiving:

- a) a DWS Lost message if this state does not send a Start DWS message; or
- b) a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.4.14.3 Transition SP30:SAS_TrainingDone to SP1:OOB_AwaitCOMX

This transition shall occur if a phy reset problem occurs.

Before the transition, this state shall set the ResetStatus state machine variable to PHY_RESET_PROBLEM.

5.9.4.14.4 Transition SP30:SAS_TrainingDone to SP28:SAS_TrainSetup

This transition shall occur if:

- a) the MTT timer expires; and
- b) the Commonly Supported Settings state machine variable contains additional commonly supported settings.

5.9.4.14.5 Transition SP30:SAS_TrainingDone to SP15:SAS_PHY_Ready

This transition shall occur if this state receives at least four TRAIN_DONE Pattern Transmitted messages and either:

- a) receives:
 - A) a TRAIN_DONE Received message before the MTT timer expires; and
 - B) at least one TRAIN_DONE Pattern Transmitted message after receiving the TRAIN_DONE Received message in A) (i.e., after receiving TRAIN_DONE, transmit at least one more TRAIN_DONE),
 or
- b) was entered with a TRAIN_DONE Received argument.

Before the transition, this state shall stop the MTT timer.

5.9.5 SAS phy power conditions states

5.9.5.1 SAS phy power conditions states overview

Figure 82 shows the SAS phy power conditions states. These states are entered when a phy is requested to enter a low phy power condition and process the actions that return a phy from a low phy power condition to the active phy power condition.

These states are indicated by state names with a prefix of SAS_PS.

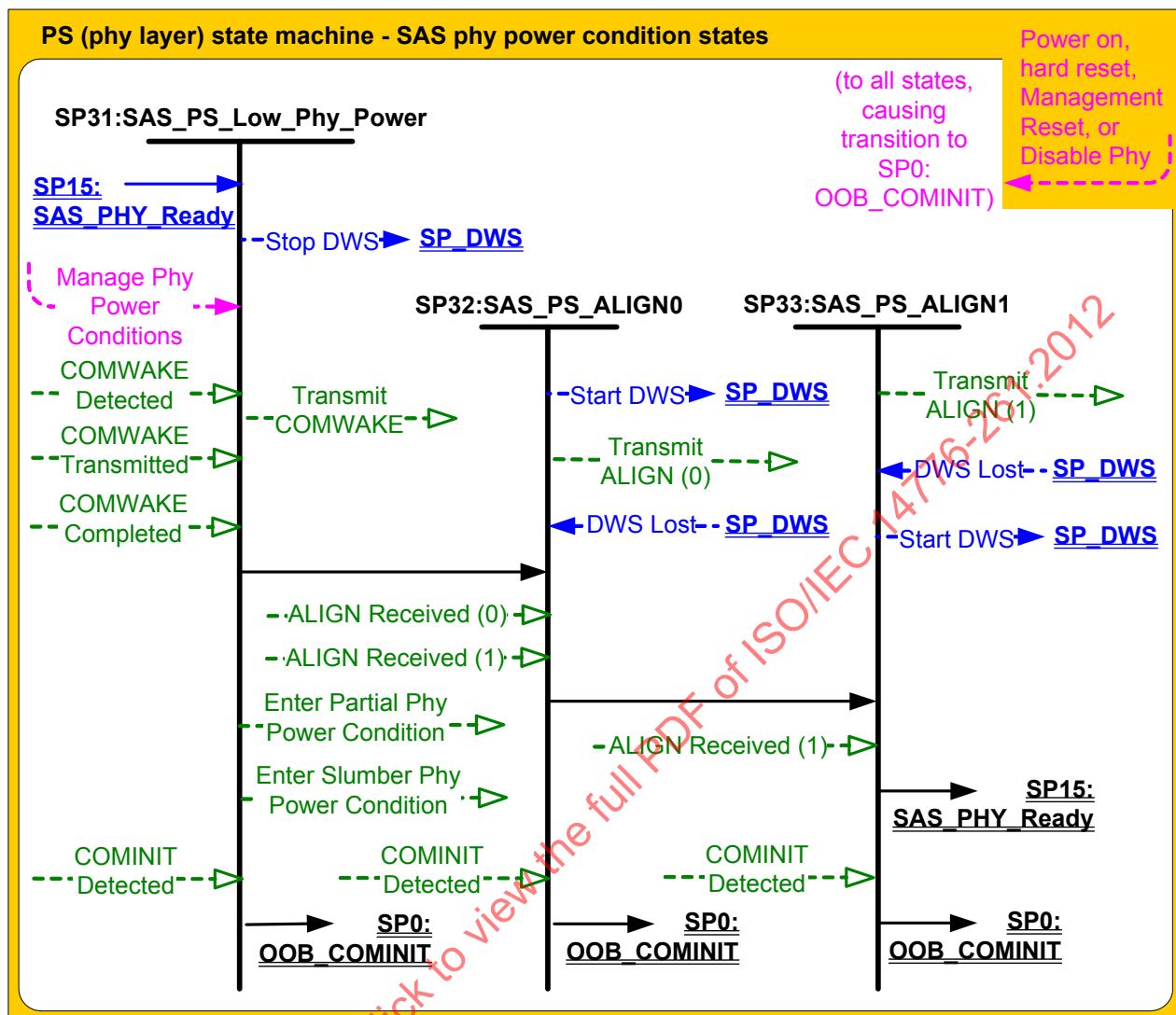


Figure 82 — SP (phy layer) state machine - SAS phy power condition states

5.9.5.2 SP31:SAS_PS_Low_Phy_Power state

5.9.5.2.1 State description

Upon entry into this state, this state shall:

- send a Stop DWS message to the SP_DWS state machine; and
- if applicable, save any vendor specific information for the SP transmitter and SP receiver (e.g., determined from the previous Train-SNW with the arguments set to the same values as those for the previous entry into the SP28:SAS_TrainSetup state (see 5.9.4.12)).

If this state is entered with a Partial argument, then:

- this state shall send an Enter Partial Phy Power Condition message to the SP transmitter and SP receiver;
- the phy shall enter the partial phy power condition (see 4.10.1.3); and
- this state shall set the SASPhyPwrCond state machine variable to Partial.

If this state is entered with a Slumber argument, then:

- a) this state shall send an Enter Slumber Phy Power Condition message to the SP transmitter and SP receiver;
- b) the phy shall enter the slumber phy power condition (see 4.10.1.4); and
- c) this state shall set the SASPhyPwrCond state machine variable to Slumber.

If this state receives:

- a) a Manage Phy Power Conditions (Exit) request or a COMWAKE Detected message; and
- b) the phy is in a partial phy power condition,

then this state shall send a Transmit COMWAKE message to the SP transmitter within a phy wakeup partial timeout (see table 44).

If this state receives:

- a) a Manage Phy Power Conditions (Exit) request or a COMWAKE Detected message; and
- b) the phy is in a slumber phy power condition,

then this state shall send a Transmit COMWAKE message to the SP transmitter within a phy wakeup slumber timeout (see table 44).

While in this state, if a Manage Phy Power Conditions (Exit) request and a COMWAKE Detected message are received, then this state shall only send one Transmit COMWAKE message to the SP transmitter.

5.9.5.2.2 Transition SP31:SAS_PS_Low_Phy_Power to SP0:OOB_COMINIT

This transition shall occur if:

- a) this state:
 - A) receives a COMWAKE Transmitted message; and
 - B) does not receive a COMWAKE Completed message within a hot-plug timeout (see table 44);
 or
- b) this state receives a COMINIT Detected message.

5.9.5.2.3 Transition SP31:SAS_PS_Low_Phy_Power to SP32:SAS_PS_ALIGN0

This transition shall occur after this state:

- a) receives a COMWAKE Transmitted message; and
- b) receives a COMWAKE Completed message.

5.9.5.3 SP32:SAS_PS_ALIGN0 state

5.9.5.3.1 State description

Upon entry into this state, the phy shall:

- 1) initialize and start the SNTT timer and the SNLT timer;
- 2) send a Set Physical Link Rate message to the SP transmitter and to the SP receiver and send a Set SSC message to the SP transmitter with the arguments set to those determined from the last SNW;
- 3) if applicable, restore any vendor specific information for the SP transmitter and SP receiver (see 5.9.5.2.1);
- 4) send a Start DWS message to the SP_DWS state machine; and
- 5) repeatedly send Transmit ALIGN (0) messages to the SP transmitter.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to re-acquire dword synchronization without running a new link reset sequence.

5.9.5.3.2 Transition SP32:SAS_PS_ALIGN0 state to SP0:OOB_COMINIT

This transition shall occur after this state:

- a) receives a DWS Lost message, if this state does not send a Start DWS message;
- b) receives a COMINIT Detected message; or
- c) the SNTT timer expires.

Before the transition, this state shall stop the SNLT timer and the SNTT timer.

5.9.5.3.3 Transition SP32:SAS_PS_ALIGN0 to SP33:SAS_PS_ALIGN1

This transition shall occur:

- a) if this state receives an ALIGN Received (0) message or an ALIGN Received (1) message before the SNLT timer expires;
- b) after this state has sent at least three Transmit ALIGN (0) messages; and
- c) after this state has stopped the SNLT timer.

5.9.5.4 SP33:SAS_PS_ALIGN1 state

5.9.5.4.1 State description

Upon entry into this state, the phy shall repeatedly send Transmit ALIGN (1) messages to the SP transmitter.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to re-acquire dword synchronization without running a new link reset sequence.

Receipt of the ALIGN Received (1) message indicates that the attached phy has been able to achieve dword synchronization with the previously negotiated settings.

If this state receives an ALIGN Received (1) message before the SNTT timer expires, then this state shall set the SASPhyPwrCond state machine variable to Active.

5.9.5.4.2 Transition SP33:SAS_PS_ALIGN1 state to SP0:OOB_COMINIT

This transition shall occur after this state:

- a) receives a DWS Lost message, if this state does not send a Start DWS message;
- b) receives a COMINIT Detected message; or
- c) the SNTT timer expires.

Before the transition, this state shall stop the SNLT timer and the SNTT timer.

5.9.5.4.3 Transition SP33:SAS_PS_ALIGN1 state to SP15:SAS_PHY_Ready

This transition shall occur:

- a) if this state receives an ALIGN Received (1) message before the SNTT timer expires;
- b) after this state has sent at least three Transmit ALIGN (1) messages;
- c) after this state has set the SASPhyPwrCond state machine variable to Active; and
- d) after this state has stopped the SNLT timer and the SNTT timer.

5.9.6 SATA host emulation states

5.9.6.1 SATA host emulation states overview

Figure 83 shows the SATA host emulation states, in which the phy has detected that it is attached to a SATA phy and behaves as if it were a SATA host phy initiating the SATA speed negotiation sequence. These states are indicated by state names with a prefix of SATA.

The power management states defined in this standard are for SAS initiator phys or SAS expander phys that support SATA. SATA low phy power conditions may be enabled in expander phys using the SMP PHY CONTROL function (see 9.4.3.28).

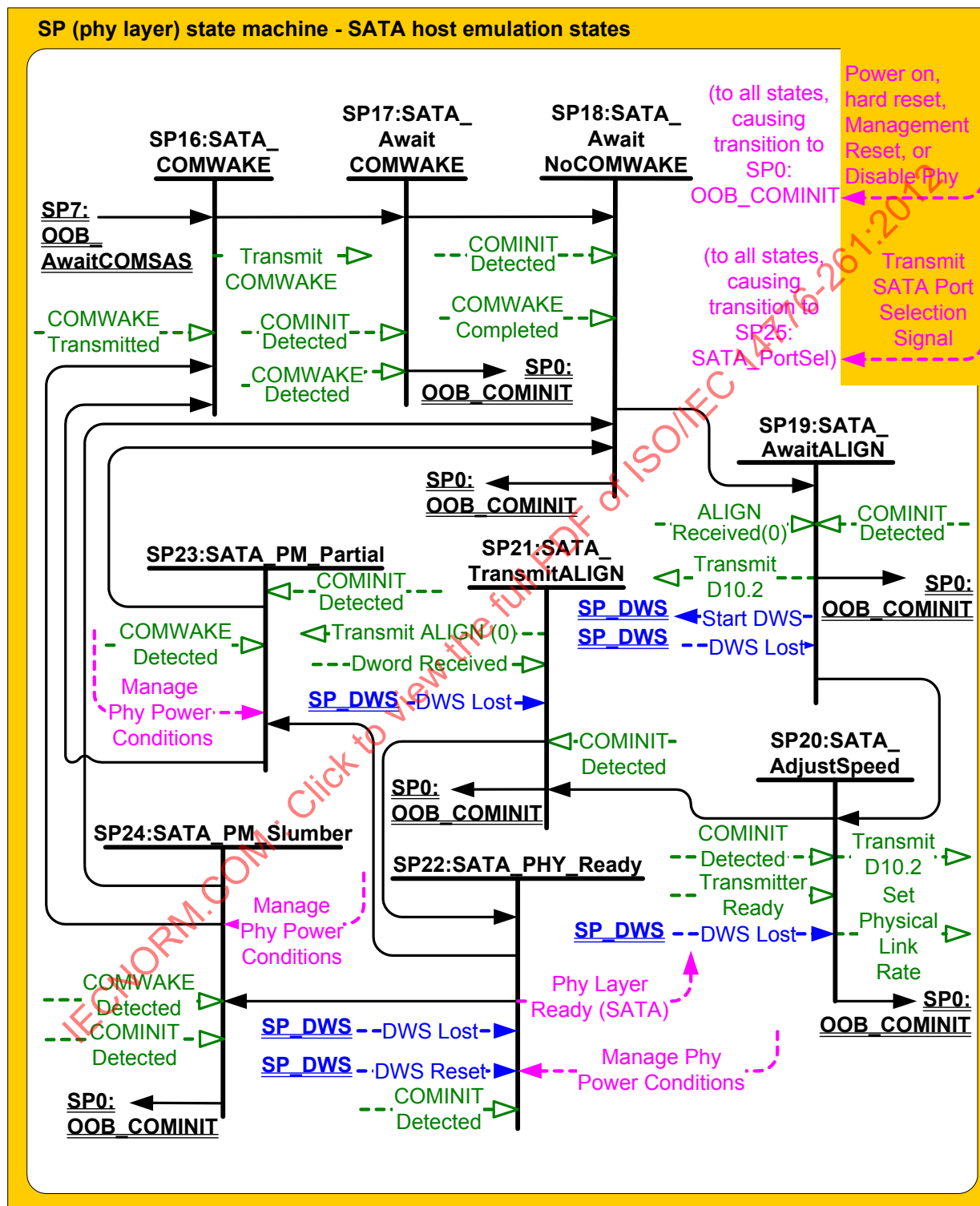


Figure 83 — SP (phy layer) state machine - SATA host emulation states

5.9.6.2 SP16:SATA_COMWAKE state

5.9.6.2.1 State description

This state shall send a Transmit COMWAKE message to the SP transmitter and wait for a COMWAKE Transmitted message.

5.9.6.2.2 Transition SP16:SATA_COMWAKE to SP17:SATA_AwaitCOMWAKE

This transition shall occur after receiving a COMWAKE Transmitted message.

5.9.6.3 SP17:SATA_AwaitCOMWAKE state

5.9.6.3.1 State description

This state waits for COMWAKE to be received.

5.9.6.3.2 Transition SP17:SATA_AwaitCOMWAKE to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

5.9.6.3.3 Transition SP17:SATA_AwaitCOMWAKE to SP18:SATA_AwaitNoCOMWAKE

This transition shall occur after receiving a COMWAKE Detected message.

5.9.6.4 SP18:SATA_AwaitNoCOMWAKE state

5.9.6.4.1 State description

This state waits for a COMWAKE Completed message.

5.9.6.4.2 Transition SP18:SATA_AwaitNoCOMWAKE to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.6.4.3 Transition SP18:SATA_AwaitNoCOMWAKE to SP19:SATA_AwaitALIGN

This transition shall occur after receiving a COMWAKE Completed message.

5.9.6.5 SP19:SATA_AwaitALIGN state

5.9.6.5.1 State description

Upon entry into this state, the phy shall send a Start DWS message to the SP_DWS state machine.

This state shall:

- a) repeatedly send Transmit D10.2 messages to the SP transmitter;
- b) initialize and start the Await ALIGN Timeout timer; and
- c) wait for an ALIGN Received (0) message to be received or for the Await ALIGN Timeout timer to expire.

The phy shall start transmitting D10.2 characters no later than a COMWAKE response time (see 5.7.2.2) after entry into this state.

5.9.6.5.2 Transition SP19:SATA_AwaitALIGN to SP0:OOB_COMINIT

This transition shall occur:

- a) if the Await ALIGN Timeout timer expires;
- b) after receiving a DWS Lost message; or
- c) after receiving a COMINIT Detected message.

Before the transition, this state shall:

- a) if the Await ALIGN Timeout timer expires, then set the ResetStatus state machine variable to UNSUPPORTED_PHY_ATTACHED;
- b) after receiving a DWS Lost message, set the ResetStatus state machine variable to UNKNOWN; or
- c) after receiving a COMINIT Detected message, set the ResetStatus state machine variable to UNKNOWN.

5.9.6.5.3 Transition SP19:SATA_AwaitALIGN to SP20:SATA_AdjustSpeed

This transition shall occur if this state receives an ALIGN Received (0) message before the Await ALIGN Timeout timer expires. The ALIGN Received (0) message indicates an ALIGN (0) was received at any of the physical link rates supported by this phy.

5.9.6.6 SP20:SATA_AdjustSpeed state

5.9.6.6.1 State description

This state waits for the SP transmitter to adjust to the same physical link rate of the ALIGNs that were detected by the receiver circuitry.

This state shall:

- 1) send a Set Physical Link Rate message to the SP transmitter with an argument specifying the physical link rate of the ALIGNs that were detected by the receiver circuitry; and
- 2) repeatedly send Transmit D10.2 messages to the SP transmitter.

5.9.6.6.2 Transition SP20:SATA_AdjustSpeed to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message or a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.6.6.3 Transition SP20:SATA_AdjustSpeed to SP21:SATA_TransmitALIGN

This transition shall occur after receiving a Transmitter Ready message.

5.9.6.7 SP21:SATA_TransmitALIGN state

5.9.6.7.1 State description

This state shall repeatedly send Transmit ALIGN (0) messages to the SP transmitter.

5.9.6.7.2 Transition SP21:SATA_TransmitALIGN to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message or a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.6.7.3 Transition SP21:SATA_TransmitALIGN to SP22:SATA_PHY_Ready

This transition shall occur after receiving three consecutive Dword Received messages containing primitives other than ALIGN (0).

5.9.6.8 SP22:SATA_PHY_Ready state

5.9.6.8.1 State description

While in this state dwords from the link layer are transmitted at the negotiated physical link rate (i.e., the rate established in the previous state).

Upon entry into this state, this state shall:

- a) if the SP transmitter is transmitting at 1.5 Gbps, then set the ResetStatus state machine variable to G1;
- b) if the SP transmitter is transmitting at 3 Gbps, then set the ResetStatus state machine variable to G2;
- or
- c) if the SP transmitter is transmitting at 6 Gbps, then set the ResetStatus state machine variable to G3.

This state shall send a Phy Layer Ready (SATA) confirmation to the link layer.

This state waits for a COMINIT Detected message, a DWS Lost message, or a DWS Reset message.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

5.9.6.8.2 Transition SP22:SATA_PHY_Ready to SP0:OOB_COMINIT

If optical mode is enabled or D.C. mode is enabled, then this transition shall occur after receiving:

- a) a DWS Lost message, if this state does not send a Start DWS message;
- b) a DWS Lost message followed by a COMINIT Detected message, if this state does not send a Start DWS message; or
- c) a DWS Reset message.

If optical mode is enabled, then this transition shall occur after receiving:

- a) a COMINIT Detected message before receiving a DWS Lost message; or
- b) a COMINIT Detected message after sending a Start DWS message.

5.9.6.8.3 Transition SP22:SATA_PHY_Ready to SP23:SATA_PM_Partial

This transition shall occur after receiving a Manage Phy Power Conditions (Enter Partial) request.

5.9.6.8.4 Transition SP22:SATA_PHY_Ready to SP24:SATA_PM_Slumber

This transition shall occur after receiving a Manage Phy Power Conditions (Enter Slumber) request.

5.9.6.9 SP23:SATA_PM_Partial state

5.9.6.9.1 State description

Upon entry into this state, this state shall set the SASPhyPwrCond state machine variable to Partial.

This state waits for a COMWAKE Detected message or a Manage Phy Power Conditions (Exit) request.

5.9.6.9.2 Transition SP23:SATA_PM_Partial to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.6.9.3 Transition SP23:SATA_PM_Partial to SP16:SATA_COMWAKE

This transition shall occur after receiving a Manage Phy Power Conditions (Exit) request.

5.9.6.9.4 Transition SP23:SATA_PM_Partial to SP18:SATA_AwaitNoCOMWAKE

This transition shall occur after receiving a COMWAKE Detected message.

5.9.6.10 SP24:SATA_PM_Slumber state

5.9.6.10.1 State description

Upon entry into this state, this state shall set the SASPhyPwrCond state machine variable to Slumber.

This state waits for a COMWAKE Detected message or a Manage Phy Power Conditions (Exit) request.

5.9.6.10.2 Transition SP24:SATA_PM_Slumber to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

Before the transition, this state shall set the ResetStatus state machine variable to UNKNOWN.

5.9.6.10.3 Transition SP24:SATA_PM_Slumber to SP16:SATA_COMWAKE

This transition shall occur after receiving a Manage Phy Power Conditions (Exit) request.

5.9.6.10.4 Transition SP24:SATA_PM_Slumber to SP18:SATA_AwaitNoCOMWAKE

This transition shall occur after receiving a COMWAKE Detected message.

5.9.7 SATA port selector state SP25:SATA_PortSel

5.9.7.1 State description

Figure 84 shows the SP25:SATA_PortSel state. This state controls transmission of the SATA port selection signal when a specified phy processes a Transmit SATA Port Selection Signal request.

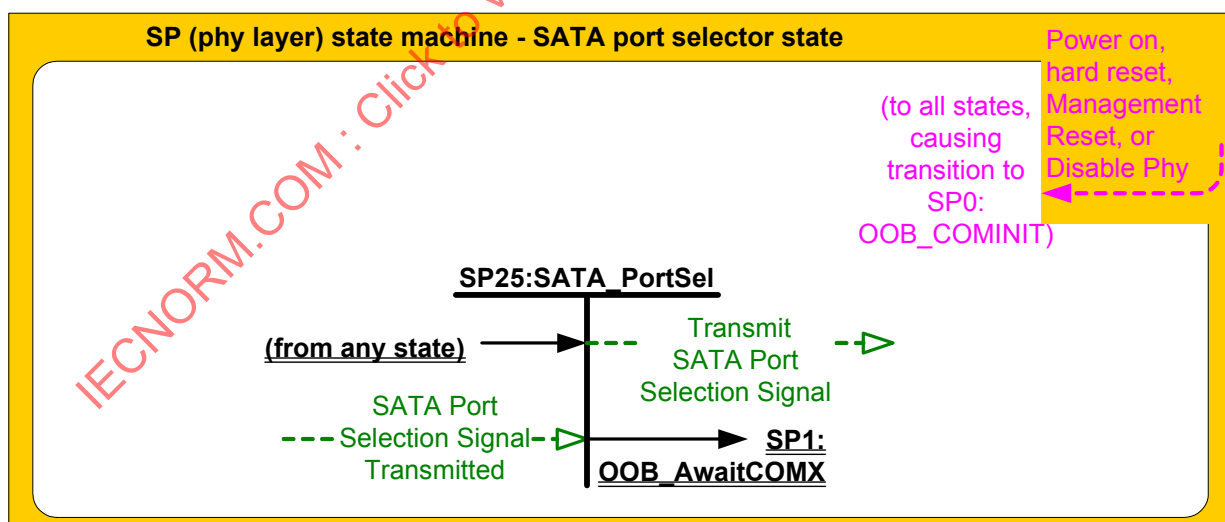


Figure 84 — SP (phy layer) state machine – SATA port selector state

Upon entry into this state, the phy shall:

- set the ResetStatus state machine variable to UNKNOWN;
- send a Transmit SATA Port Selection Signal message to the SP transmitter;
- set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response (see 9.4.3.10); and

- ### 5.9.7.2 Transition SP25:SATA_PortSel to SP1:OOB_AwaitCOMX

5.9.8 SATA spinup hold state SP26:SATA_SpinupHold

5.9.8.1 State description

SP (phy layer) state machine - SATA spinup hold state

SP26:SATA_SpinupHold

(in expander phys)
SATA Spinup Hold

SP7:OOB_AwaitCOMSAS

---COMINIT Detected---

SP0:OOB_COMINIT

(to all states, causing transition to SP0: OOB_COMINIT)

(to all states, causing transition to SP25:SATA_PortSel)

Power-on, hard reset, Management Reset, or Disable Phy

Transmit SATA Port Selection Signal

Figure 85 — SP (phy layer) state machine - SATA spinup hold state

- a) if the ResetStatus state machine variable is set to SPINUP_HOLD, then not change the ResetStatus state machine variable; or
- b) if the ResetStatus state machine variable is not set to SPINUP_HOLD, then:
 - A) set the ResetStatus state machine variable to SPINUP_HOLD; and
 - B) if this state machine is in an expander phy, then send a SATA Spinup Hold confirmation to the link layer.

5.9.8.2 Transition SP26:SATA_SpinupHold to SP0:OOB_COMINIT

This transition shall occur if this state receives a COMINIT Detected message.

5.10 SP DWS (phy layer dword synchronization) state machine

5.10.1 SP_DWS state machine overview

Each phy includes an SP DWS state machine and an SP DWS receiver.

The SP_DWS state machine establishes the same dword boundaries at the receiver as at the attached transmitter by searching for control characters. The SP_DWS receiver monitors and decodes the incoming data stream and forces K28.5 characters into the first character position to perform dword alignment when requested by the SP_DWS state machine. K28.5 characters with either positive or negative disparity shall be

accepted. The SP_DWS receiver continues to reestablish dword alignment by forcing received K28.5 characters into the first character position until a K28.5-based primitive (i.e., K28.5, Dxx.y, Dxx.y, Dxx.y) with correct disparity on each data character is detected. The resultant primitives, dwords and valid dword indicators (e.g., encoding error indicators) are sent to this state machine to enable it to determine the dword synchronization policy.

After dword synchronization has been achieved, this state machine evaluates dwords that are received. When an invalid dword is detected, receipt of two valid dwords are required to nullify the effect of receiving the invalid dword. When four invalid dwords are detected without nullification, dword synchronization is considered lost.

While dword synchronization is lost, the data stream received is invalid and dwords shall not be passed to the link layer.

This state machine consists of the following states:

- a) SP_DWS0:AcquireSync (see 5.10.3) (initial state);
- b) SP_DWS1:Valid1 (see 5.10.4);
- c) SP_DWS2:Valid2 (see 5.10.5);
- d) SP_DWS3:SyncAcquired (see 5.10.6);
- e) SP_DWS4:Lost1 (see 5.10.7);
- f) SP_DWS5:Lost1Recovered (see 5.10.8);
- g) SP_DWS6:Lost2 (see 5.10.9);
- h) SP_DWS7:Lost2Recovered (see 5.10.10);
- i) SP_DWS8:Lost3 (see 5.10.11); and
- j) SP_DWS9:Lost3Recovered (see 5.10.12).

This state machine receives the following requests from the management application layer:

- a) Management Reset; and
- b) Disable Phy.

This state machine shall start in the SP_DWS0:AcquireSync state after:

- a) power on;
- b) hard reset;
- c) receiving a Management Reset request (e.g., as a result of processing an SMP PHY CONTROL function requesting a phy operation of LINK RESET or HARD RESET);
- d) receiving a Disable Phy request (e.g., as a result of processing an SMP PHY CONTROL function requesting a phy operation of DISABLE); or
- e) receiving a Stop DWS message from the SP state machine.

This state machine receives the following messages from the SP state machine (see 5.9):

- a) Start DWS; and
- b) Stop DWS.

This state machine sends the following messages to the SP state machine:

- a) DWS Lost; and
- b) DWS Reset.

This state machine shall maintain the timers listed in table 57.

Table 57 — SP_DWS state machine timers

Timer	Initial value
DWS Reset Timeout timer	1 ms

Figure 86 shows the SP_DWS state machine.

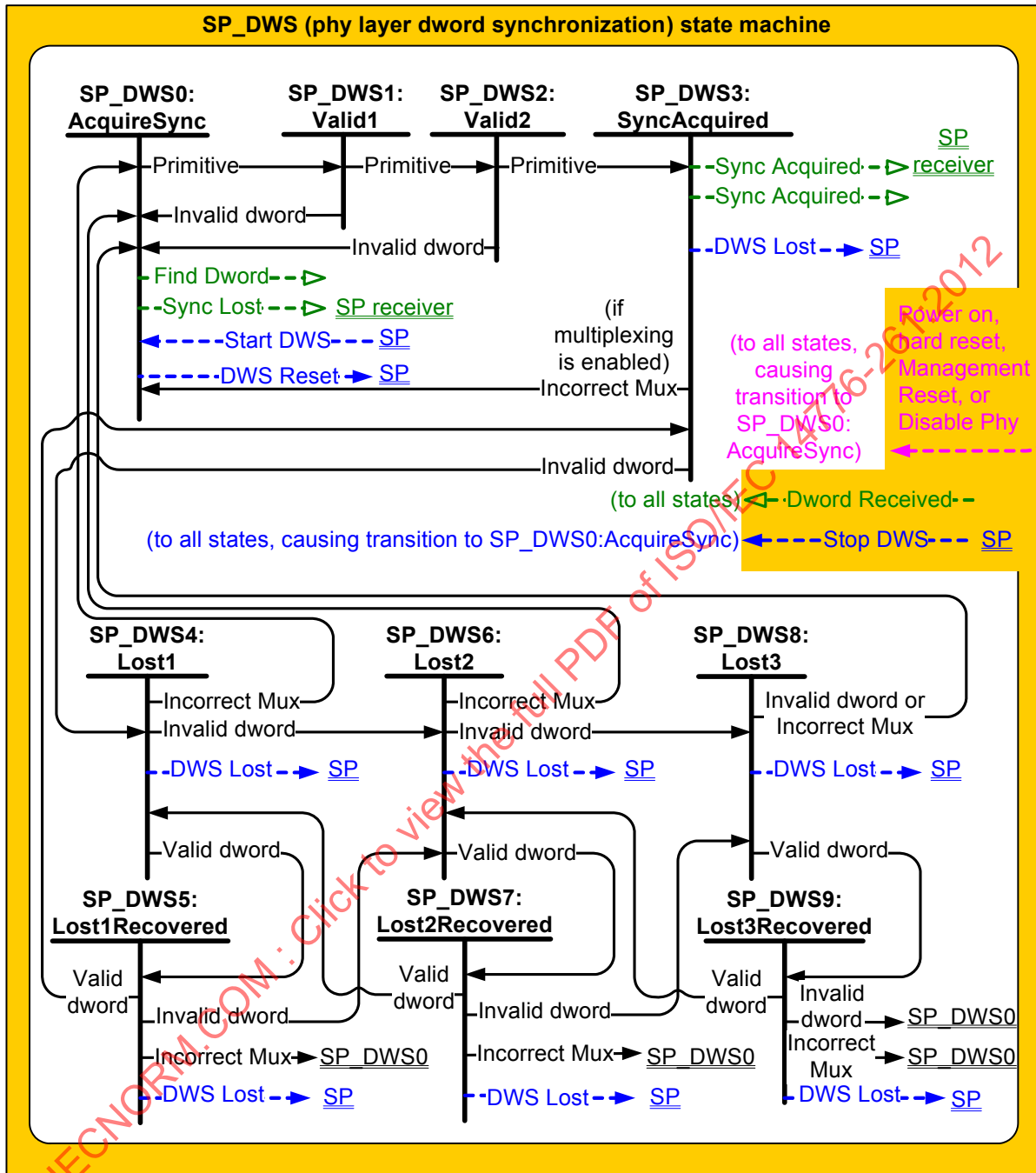


Figure 86 — SP_DWS (phy layer dword synchronization) state machine

5.10.2 SP_DWS receiver

The SP_DWS receiver receives the following message from the SP_DWS state machine:

- Find Dword; and
- Sync Acquired.

The SP_DWS receiver sends the following messages to the SP_DWS state machine indicating dwords received from the physical link:

- Dword Received (Primitive);

- b) Dword Received (Data Dword);
- c) Dword Received (Invalid); and
- d) Incorrect Mux Received.

When the SP_DWS receiver receives a Sync Acquired message, the SP_DWS receiver shall send the most recently received primitive and all subsequent dwords to the link layer state machine receivers (e.g., SL_IR, SL, SSP, SMP, and XL) through the elasticity buffer (see 6.3) as Dword Received confirmations. If multiplexing is enabled (see table 51 in 5.7.4.2.3.3), then the SP_DWS receiver shall use the first incoming MUX to determine the logical phy to which it sends each Dword Received confirmation and shall not send an Dword Received confirmations until it receives the first incoming MUX.

Upon receiving a Find Dword message, the SP_DWS receiver shall monitor the input data stream and force each K28.5 character detected into the first character position of a possible dword. If the next three characters are data characters with correct disparity, then the SP_DWS receiver shall send the dword as a Dword Received (Primitive) message to the SP_DWS state machine. Until the SP_DWS receiver receives another Find Dword message, for every four characters that it receives it shall:

- a) send a Dword Received (Invalid) message to the SP_DWS state machine if the dword is an invalid dword (see 3.1.94);
- b) send a Dword Received (Primitive) message to the SP_DWS state machine if the dword is a primitive (i.e., the dword contains a K28.5 character in the first character position followed by three data characters); or
- c) send a Dword Received (Data Dword) message to the SP_DWS state machine if the dword is a data dword (i.e., it is not an invalid dword or a primitive).

The SP_DWS receiver relationship to other receivers is defined in 4.3.3.

5.10.3 SP_DWS0:AcquireSync state

5.10.3.1 State description

This is the initial state of this state machine.

After receiving a Start DWS message, this state shall:

- a) send a Find Dword message to the SP_DWS receiver;
- b) send a Sync Lost message to the SP receiver; and
- c) initialize and start the DWS Reset Timeout timer;

If this state is entered from SP_DWS1:Valid1 or SP_DWS2:Valid2, then:

- a) this state shall send a Find Dword message to the SP_DWS receiver;
- b) this state shall send a Sync Lost message to the SP receiver; and
- c) the DWS Reset Timeout timer shall continue running.

If the DWS Reset Timeout timer expires, then this state may send a DWS Reset message to the SP state machine (e.g., if the phy chooses to initiate a new link reset sequence because dword synchronization has been lost for too long).

This state shall not send a DWS Reset message to the SP until the DWS Reset Timeout timer expires.

5.10.3.2 Transition SP_DWS0:AcquireSync to SP_DWS1:Valid1

This transition shall occur after sending a Find Dword message and receiving a Dword Received (Primitive) message.

5.10.4 SP_DWS1:Valid1 state

5.10.4.1 State description

This state is reached after one valid primitive has been received. This state waits for a second valid primitive or an invalid dword.

The DWS Reset Timeout timer shall continue running.

5.10.4.2 Transition SP_DWS1:Valid1 to SP_DWS0:AcquireSync

This transition shall occur after receiving a Dword Received (Invalid) message or after the DWS Reset Timeout timer expires.

5.10.4.3 Transition SP_DWS1:Valid1 to SP_DWS2:Valid2

This transition shall occur after receiving a Dword Received (Primitive) message.

5.10.5 SP_DWS2:Valid2 state

5.10.5.1 State description

This state is reached after two valid primitives have been received without adjusting the dword synchronization. This state waits for a third valid primitive or an invalid dword.

The DWS Reset Timeout timer shall continue running.

5.10.5.2 Transition SP_DWS2:Valid2 to SP_DWS0:AcquireSync

This transition shall occur after receiving a Dword Received (Invalid) message or after the DWS Reset Timeout timer expires.

5.10.5.3 Transition SP_DWS2:Valid2 to SP_DWS3:SyncAcquired

This transition shall occur after receiving a Dword Received (Primitive) message.

5.10.6 SP_DWS3:SyncAcquired state

5.10.6.1 State description

This state is reached after three valid primitives have been received without adjusting the dword synchronization.

Upon entry into this state, the phy shall send a Sync Acquired message to the SP_DWS receiver and the SP receiver. The most recently received primitive and all subsequent dwords shall be forwarded through the elasticity buffer (see 6.3) for processing by the link layer.

This state waits for a Dword Received (Invalid) message, which indicates a potential loss of dword synchronization, or an Incorrect Mux Received message, which indicates that dword synchronization is lost.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.6.2 Transition SP_DWS3:SyncAcquired to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.6.3 Transition SP_DWS3:SyncAcquired to SP_DWS4:Lost1

This transition shall occur after receiving a Dword Received (Invalid) message.

5.10.7 SP_DWS4:Lost1 state

5.10.7.1 State description

This state is reached when one invalid dword has been received and not nullified. This state waits for a Dword Received message or an Incorrect Mux Received message.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.7.2 Transition SP_DWS4:Lost1 to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.7.3 Transition SP_DWS4:Lost1 to SP_DWS5:Lost1Recovered

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

5.10.7.4 Transition SP_DWS4:Lost1 to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Invalid) message.

5.10.8 SP_DWS5:Lost1Recovered state

5.10.8.1 State description

This state is reached when a valid dword has been received after one invalid dword had been received. This state waits for a Dword Received message or an Incorrect Mux Received message.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.8.2 Transition SP_DWS5:Lost1Recovered to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.8.3 Transition SP_DWS5:Lost1Recovered to SP_DWS3:SyncAcquired

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

5.10.8.4 Transition SP_DWS5:Lost1Recovered to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Invalid) message.

5.10.9 SP_DWS6:Lost2 state

5.10.9.1 State description

This state is reached when two invalid dwords have been received and not nullified. This state waits for a Dword Received message or an Incorrect Mux Received message.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.9.2 Transition SP_DWS6:Lost2 to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.9.3 Transition SP_DWS6:Lost2 to SP_DWS7:Lost2Recovered

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

5.10.9.4 Transition SP_DWS6:Lost2 to SP_DWS8:Lost3

This transition shall occur after receiving a Dword Received (Invalid) message.

5.10.10 SP_DWS7:Lost2Recovered state

5.10.10.1 State description

This state is reached when a valid dword has been received after two invalid dwords had been received. This state waits for a Dword Received message or an Incorrect Mux Received message.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.10.2 Transition SP_DWS7:Lost2Recovered to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.10.3 Transition SP_DWS7:Lost2Recovered to SP_DWS4:Lost1

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

5.10.10.4 Transition SP_DWS7:Lost2Recovered to SP_DWS8:Lost3

This transition shall occur after receiving a Dword Received (Invalid) message.

5.10.11 SP_DWS8:Lost3 state

5.10.11.1 State description

This state is reached when three invalid dwords have been received and not nullified. This state waits for a Dword Received message or an Incorrect Mux Received message.

If a Dword Received (Invalid) message is received (i.e., the fourth non-nullified invalid dword is received), then this state shall send a DWS Lost message to the SP state machine.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.11.2 Transition SP_DWS8:Lost3 to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.11.3 Transition SP_DWS8:Lost3 to SP_DWS9:Lost3Recovered

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

5.10.12 SP_DWS9:Lost3Recovered state

5.10.12.1 State description

This state is reached when a valid dword has been received after three invalid dwords had been received. This state waits for a Dword Received message or an Incorrect Mux Received message.

If a Dword Received (Invalid) message is received (i.e., the fourth non-nullified invalid dword is received), then this state shall send a DWS Lost message to the SP state machine.

If an Incorrect Mux Received message is received, then this state shall send a DWS Lost message to the SP state machine.

5.10.12.2 Transition SP_DWS9:Lost3Recovered to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

5.10.12.3 Transition SP_DWS9:Lost3Recovered to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

5.11 Multiplexing

If SNW-3 indicates multiplexing is enabled (see table 51 in 5.7.4.2.3.3), then the phy shall begin multiplexing immediately after the multiplexing sequence (see 5.7.4.3).

Figure 87 shows multiplexing disabled (i.e., one logical link).

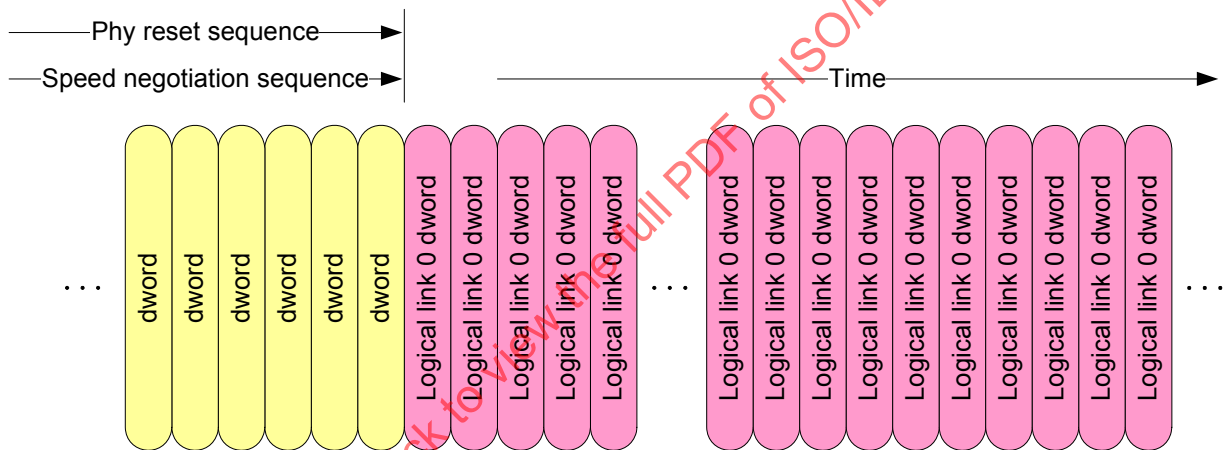


Figure 87 — Multiplexing disabled

Figure 88 shows multiplexing enabled (i.e., two logical links).

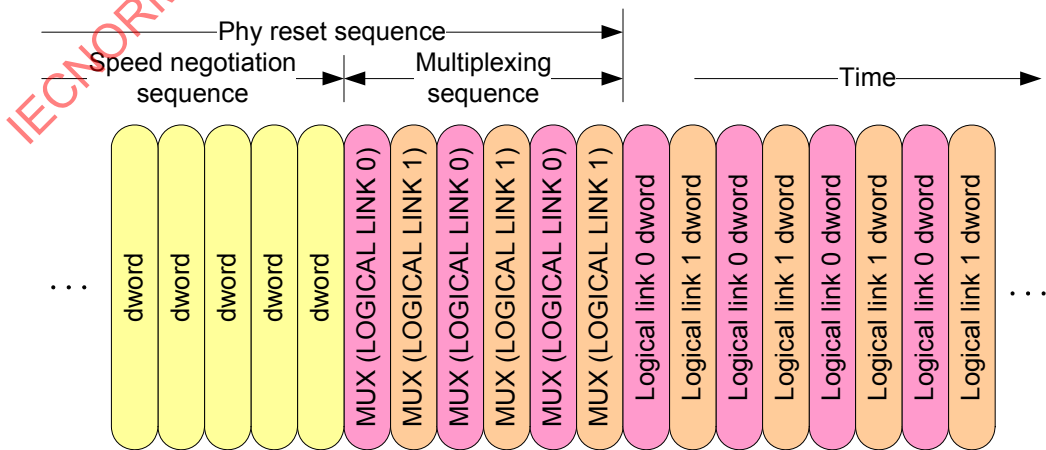


Figure 88 — Multiplexing enabled

After the multiplexing sequence completes, each logical phy shall honor the deletable primitive insertion requirements for physical link rate tolerance management defined in 6.3. The logical phys shall ignore MUXs.

If a phy with multiplexing enabled ever loses dword synchronization, then it shall restart the link reset sequence rather than attempt to reestablish dword synchronization.

Once the multiplexing sequence is complete, the phy shall not perform another multiplexing sequence until a new link reset sequence.

Once the multiplexing sequence is complete:

- a) a logical phy originating dwords shall transmit MUX as a deletable primitive (e.g., substituted in place of an ALIGN) at least once every millisecond; and
- b) a logical phy forwarding dwords should transmit MUX as a deletable primitive at least once every millisecond to confirm the logical link numbers.

Transmitting NOTIFY has higher priority than transmitting MUX.

NOTE 22 Periodic MUX transmission is for the convenience of logic analyzers and to provide additional assurance that the receiving phy is in agreement with the transmitting phy.

If a phy ever receives a MUX that does not match the MUX expected in that position (i.e., it receives MUX (LOGICAL LINK 1) on logical link 0 or receives MUX (LOGICAL LINK 0) on logical link 1), then it shall perform a link reset sequence.

5.12 Spinup

If a SAS target device receives COMSAS during the OOB sequence, then it shall not temporarily consume additional power (e.g., to spin up rotating media) until allowed by the SA_PC state machine (see 9.2.10).

Expander devices that detect an attached SATA phy may halt the automatic phy reset sequence after the COMSAS Detect Timeout (see 5.9) to delay temporary consumption of additional power. The resulting SATA spinup hold is reported in the NEGOTIATED PHYSICAL LINK RATE field in the SMP DISCOVER response (see 9.4.3.10) and is released with the SMP PHY CONTROL function (see 9.4.3.28).

NOTE 23 Enclosures supporting both SATA devices and SAS target devices may need to sequence power to each attached device to avoid excessive power consumption during power on, since the SATA devices may temporarily consume additional power automatically after power on if staggered spin-up is not implemented (see SATA).

6 Link layer

6.1 Link layer overview

The link layer defines primitives, address frames, and connections. Link layer state machines interface to the port layer and the phy layer and perform the identification and hard reset sequences, connection management, and SSP, STP, and SMP specific frame transmission and reception.

6.2 Primitives

6.2.1 Primitives overview

Primitives are dwords whose first character is a K28.3 or K28.5. Primitives are not considered big-endian or little-endian. Primitives are interpreted as first, second, third, and last characters. Table 58 defines the primitive format.

Table 58 — Primitive format

Character	Description
First	K28.5 control character (for primitives defined in this standard) or K28.3 control character (for primitives defined by SATA).
Second	Constant data character.
Third	Constant data character.
Last	Constant data character.

6.2.2 Primitive summary

Table 59 defines the deletable primitives.

Table 59 — Deletable primitives

Primitive	Use ^a	From ^b			To ^b			Primitive sequence type ^c
		I	E	T	I	E	T	
ALIGN (0)	All, SpNeg							Single
ALIGN (1)	SAS, SpNeg	I	E	T	I	E	T	
ALIGN (2)	SAS							
ALIGN (3)								
MUX (LOGICAL LINK 0)	SAS	I	E	T	I	E	T	Single
MUX (LOGICAL LINK 1)								
NOTIFY (ENABLE SPINUP)	SAS	I	E				T	Single
NOTIFY (POWER LOSS EXPECTED)		I	E				T	
NOTIFY (RESERVED 1)					I	E	T	
OOB_IDLE	SpNeg	I	E	T	I	E	T	Single

^a The Use column indicates when the primitive is used:

- a) All: SAS logical links and SATA physical links;
- b) SAS: SAS logical links, both outside connections or inside any type of connection;
- c) NoConn: SAS logical links, outside connections;
- d) Conn: SAS logical links, inside connections;
- e) STP: SAS logical links, inside STP connections; or
- f) SpNeg: SAS physical links, during speed negotiation.

^b The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:

- a) I for SAS initiator ports;
- b) E for expander ports; and
- c) T for SAS target ports.

Expander ports are not considered originators of primitives that are being forwarded from expander port to expander port.

^c The Primitive sequence type columns indicate whether the primitive is a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 6.2.4).

Table 60 defines the primitives not specific to the type of connection.

Table 60 — Primitives not specific to type of connection (part 1 of 2)

Primitive	Use ^a	From ^b			To ^b			Primitive sequence type ^c
		I	E	T	I	E	T	
AIP (NORMAL)	NoConn		E					Extended
AIP (RESERVED 0)								
AIP (RESERVED 1)								
AIP (RESERVED 2)								
AIP (RESERVED WAITING ON PARTIAL)					I	E	T	
AIP (WAITING ON CONNECTION)			E					
AIP (WAITING ON DEVICE)			E					
AIP (WAITING ON PARTIAL)			E					
BREAK	SAS	I	E	T	I	E	T	Redundant
BREAK_REPLY	SAS	I	E	T	I	E	T	Redundant
BROADCAST (CHANGE)	NoConn	I	E		I			Redundant
BROADCAST (SES)				T	I			
BROADCAST (EXPANDER)			E		I			
BROADCAST (ASYNCHRONOUS EVENT)				T	I			
BROADCAST (RESERVED 3)								
BROADCAST (RESERVED 4)								
BROADCAST (RESERVED CHANGE 0)					I			
BROADCAST (RESERVED CHANGE 1)					I			
CLOSE (CLEAR AFFILIATION)	STP	I					T	Triple
CLOSE (NORMAL)	Conn	I		T				
CLOSE (RESERVED 0)					I		T	
CLOSE (RESERVED 1)								
EOAF	NoConn	I	E	T	I	E	T	Single
ERROR	SAS		E		I	E	T	Single
HARD_RESET	NoConn	I	E		I	E	T	Redundant
OPEN_ACCEPT	NoConn	I		T	I		T	Single

^a The Use column indicates when the primitive is used:

- a) All: SAS logical links and SATA physical links;
- b) SAS: SAS logical links, both outside connections or inside any type of connection;
- c) NoConn: SAS logical links, outside connections;
- d) Conn: SAS logical links, inside connections;
- e) STP: SAS logical links, inside STP connections; or
- f) SpNeg: SAS physical links, during speed negotiation.

^b The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:

- a) I for SAS initiator ports;
- b) E for expander ports; and
- c) T for SAS target ports.

Expander ports are not considered originators of primitives that are being forwarded from expander port to expander port.

^c The Primitive sequence type columns indicate whether the primitive is a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, an extended primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 6.2.4).

Table 60 — Primitives not specific to type of connection (part 2 of 2)

Primitive	Use ^a	From ^b			To ^b			Primitive sequence type ^c
		I	E	T	I	E	T	
OPEN_REJECT (BAD DESTINATION)	NoConn		E					Single
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)		I	E	T				
OPEN_REJECT (NO DESTINATION)			E					
OPEN_REJECT (PATHWAY BLOCKED)			E					
OPEN_REJECT (PROTOCOL NOT SUPPORTED)		I	E	T				
OPEN_REJECT (RESERVED ABANDON 1)								
OPEN_REJECT (RESERVED ABANDON 2)					I		T	
OPEN_REJECT (RESERVED ABANDON 3)								
OPEN_REJECT (RESERVED CONTINUE 0)								
OPEN_REJECT (RESERVED CONTINUE 1)								
OPEN_REJECT (RESERVED INITIALIZE 0)								
OPEN_REJECT (RESERVED INITIALIZE 1)								
OPEN_REJECT (RESERVED STOP 0)								
OPEN_REJECT (RESERVED STOP 1)								
OPEN_REJECT (RETRY)		I	E	T				
OPEN_REJECT (STP RESOURCES BUSY)			E	T	I			
OPEN_REJECT (WRONG DESTINATION)		I		T			T	
OPEN_REJECT (ZONE VIOLATION)			E		I			
PS_ACK	NoConn	I	E	T	I	E	T	Redundant
PS_NAK	NoConn	I	E	T	I	E	T	Redundant
PS_REQ (PARTIAL)	NoConn	I	E	T	I	E	T	Redundant
PS_REQ (SLUMBER)	NoConn	I	E	T	I	E	T	Redundant
SOAF	NoConn	I	E	T	I	E	T	Single
TRAIN	SpNeg	I	E	T	I	E	T	Redundant
TRAIN_DONE								

^a The Use column indicates when the primitive is used:

- a) All: SAS logical links and SATA physical links;
- b) SAS: SAS logical links, both outside connections or inside any type of connection;
- c) NoConn: SAS logical links, outside connections;
- d) Conn: SAS logical links, inside connections;
- e) STP: SAS logical links, inside STP connections; or
- f) SpNeg: SAS physical links, during speed negotiation.

^b The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:

- a) I for SAS initiator ports;
- b) E for expander ports; and
- c) T for SAS target ports.

Expander ports are not considered originators of primitives that are being forwarded from expander port to expander port.

^c The Primitive sequence type columns indicate whether the primitive is a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, an extended primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 6.2.4).

Table 61 defines the primitives used only inside SSP and SMP connections.

Table 61 — Primitives used inside SSP and SMP connections

Primitive	Use ^a	From ^b			To ^b			Primitive sequence type ^c
		I	E	T	I	E	T	
ACK	SSP	I		T	I		T	Single
CREDIT_BLOCKED	SSP	I		T	I		T	Single
DONE (ACK/NAK TIMEOUT)	SSP	I		T				Single
DONE (CREDIT TIMEOUT)		I		T				
DONE (NORMAL)		I		T				
DONE (RESERVED 0)					I		T	
DONE (RESERVED 1)								
DONE (RESERVED TIMEOUT 0)								
DONE (RESERVED TIMEOUT 1)								
EOF	SSP, SMP	I		T	I		T	Single
NAK (CRC ERROR)	SSP	I		T				Single
NAK (RESERVED 0)					I		T	
NAK (RESERVED 1)								
NAK (RESERVED 2)								
RRDY (NORMAL)	SSP	I		T				Single
RRDY (RESERVED 0)					I		T	
RRDY (RESERVED 1)								
SOF	SSP, SMP	I		T	I		T	Single

^a The Use column indicates when the primitive is used:
a) SSP: SAS logical links, inside SSP connections; or
b) SMP: SAS logical links, inside SMP connections.

^b The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:
a) I for SSP initiator ports and SMP initiator ports;
b) E for expander ports; and
c) T for SSP target ports and SMP target ports.
Expander ports are not considered originators of primitives that are being forwarded from expander port to expander port.

^c The Primitive sequence type columns indicate whether the primitive is a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, an extended primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 6.2.4).

Table 62 lists the primitives used only inside STP connections and on SATA physical links.

Table 62 — Primitives used inside STP connections and on SATA physical links

Primitive	Use ^a	From ^b			To ^b			Primitive sequence type ^c
		I	E	T	I	E	T	
SATA_CONT	STP, SATA	I		T	I		T	Single
SATA_DMAT	STP, SATA	I		T	I		T	Single
SATA_EOF	STP, SATA	I		T	I		T	Single
SATA_ERROR ^d	SATA		E				T	Single
SATA_HOLD	STP, SATA	I		T	I		T	Continued
SATA_HOLDA	STP, SATA	I		T	I		T	Continued
SATA_PMACK	STP, SATA							Repeated
SATA_PMNAK	STP, SATA	I	E				T	Repeated
SATA_PMREQ_P	STP, SATA							Continued
SATA_PMREQ_S	STP, SATA							Continued
SATA_R_ERR	STP, SATA	I		T	I		T	Continued
SATA_R_IP	STP, SATA	I		T	I		T	Continued
SATA_R_OK	STP, SATA	I		T	I		T	Continued
SATA_R_RDY	STP, SATA	I		T	I		T	Continued
SATA_SOF	STP, SATA	I		T	I		T	Single
SATA_SYNC	STP, SATA	I		T	I		T	Continued
SATA_WTRM	STP, SATA	I		T	I		T	Continued
SATA_X_RDY	STP, SATA	I		T	I		T	Continued

^a The Use column indicates when the primitive is used:

- a) STP: SAS logical links, inside STP connections; or
- b) SATA: SATA physical links.

^b The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:

- a) I for STP initiator ports and SATA host ports;
- b) E for expander ports; and
- c) T for STP target ports and SATA device ports.

Expander ports are not considered originators of primitives that are being forwarded from expander port to expander port.

^c The Primitive sequence type columns indicate whether the primitive is a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, an extended primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 6.2.4).

^d Although included in this table, SATA_ERROR is not a primitive (see 3.1.146) since it starts with K28.6. It does not appear inside STP connections. It is an invalid dword, used by expander devices forwarding an error onto a SATA physical link (see 6.2.8.1).

6.2.3 Primitive encodings

Table 63 defines the primitive encoding for deletable primitives.

Table 63 — Primitive encoding for deletable primitives

Primitive	Character				Hexadecimal
	1 st	2 nd	3 rd	4 th (last)	
ALIGN (0)	K28.5	D10.2	D10.2	D27.3	BC4A4A7Bh
ALIGN (1)	K28.5	D07.0	D07.0	D07.0	BC070707h
ALIGN (2)	K28.5	D01.3	D01.3	D01.3	BC616161h
ALIGN (3)	K28.5	D27.3	D27.3	D27.3	BC7B7B7Bh
MUX (LOGICAL LINK 0)	K28.5	D02.0	D16.7	D31.4	BC02F09Fh
MUX (LOGICAL LINK 1)	K28.5	D04.7	D31.4	D27.4	BCE49F9Bh
NOTIFY (ENABLE SPINUP)	K28.5	D31.3	D31.3	D31.3	BC7F7F7Fh
NOTIFY (POWER LOSS EXPECTED)	K28.5	D31.3	D07.0	D01.3	BC7F0761h
NOTIFY (RESERVED 1)	K28.5	D31.3	D01.3	D07.0	BC7F6107h
Obsolete	K28.5	D31.3	D10.2	D10.2	BC7F4A4Ah
OOB_IDLE	K28.5	D07.0	D03.4	D13.4	BC07838Dh

Table 64 defines the primitive encoding for primitives not specific to type of connection.

Table 64 — Primitive encoding for primitives not specific to type of connection (part 1 of 2)

Primitive	Character				Hexadecimal
	1 st	2 nd	3 rd	4 th (last)	
AIP (NORMAL)	K28.5	D27.4	D27.4	D27.4	BC9B9B9Bh
AIP (RESERVED 0)	K28.5	D27.4	D31.4	D16.7	BC9B9FF0h
AIP (RESERVED 1)	K28.5	D27.4	D16.7	D30.0	BC9BF01Eh
AIP (RESERVED 2)	K28.5	D27.4	D29.7	D01.4	BC9BFD81h
AIP (RESERVED WAITING ON PARTIAL)	K28.5	D27.4	D01.4	D07.3	BC9B8167h
AIP (WAITING ON CONNECTION)	K28.5	D27.4	D07.3	D24.0	BC9B6718h
AIP (WAITING ON DEVICE)	K28.5	D27.4	D30.0	D29.7	BC9B1EFDh
AIP (WAITING ON PARTIAL)	K28.5	D27.4	D24.0	D04.7	BC9B18E4h
BREAK	K28.5	D02.0	D24.0	D07.3	BC021867h
BREAK_REPLY	K28.5	D02.0	D29.7	D16.7	BC02FDF0h
BROADCAST (CHANGE)	K28.5	D04.7	D02.0	D01.4	BCE40281h
BROADCAST (SES)	K28.5	D04.7	D07.3	D29.7	BCE467FDh
BROADCAST (EXPANDER)	K28.5	D04.7	D01.4	D24.0	BCE48118h
BROADCAST (ASYNCHRONOUS EVENT)	K28.5	D04.7	D04.7	D04.7	BCE4E4E4h
BROADCAST (RESERVED 3)	K28.5	D04.7	D16.7	D02.0	BCE4F002h
BROADCAST (RESERVED 4)	K28.5	D04.7	D29.7	D30.0	BCE4FD1Eh
BROADCAST (RESERVED CHANGE 0)	K28.5	D04.7	D24.0	D31.4	BCE4189Fh
BROADCAST (RESERVED CHANGE 1)	K28.5	D04.7	D27.4	D07.3	BCE49B67h
CLOSE (CLEAR AFFILIATION)	K28.5	D02.0	D07.3	D04.7	BC0267E4h
CLOSE (NORMAL)	K28.5	D02.0	D30.0	D27.4	BC021E9Bh
CLOSE (RESERVED 0)	K28.5	D02.0	D31.4	D30.0	BC029F1Eh
CLOSE (RESERVED 1)	K28.5	D02.0	D04.7	D01.4	BC02E481h
EOAF	K28.5	D24.0	D07.3	D31.4	BC18679Fh
ERROR	K28.5	D02.0	D01.4	D29.7	BC0281FDh
HARD_RESET	K28.5	D02.0	D02.0	D02.0	BC020202h
OPEN_ACCEPT	K28.5	D16.7	D16.7	D16.7	BCF0F0F0h
OPEN_REJECT (BAD DESTINATION)	K28.5	D31.4	D31.4	D31.4	BC9F9F9Fh

Table 64 — Primitive encoding for primitives not specific to type of connection (part 2 of 2)

Primitive	Character				Hexadecimal
	1 st	2 nd	3 rd	4 th (last)	
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	K28.5	D31.4	D04.7	D29.7	BC9FE4FDh
OPEN_REJECT (NO DESTINATION)	K28.5	D29.7	D29.7	D29.7	BCFDFDFDh
OPEN_REJECT (PATHWAY BLOCKED)	K28.5	D29.7	D16.7	D04.7	BCFDF0E4h
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	K28.5	D31.4	D29.7	D07.3	BC9FFD67h
OPEN_REJECT (RESERVED ABANDON 1)	K28.5	D31.4	D30.0	D16.7	BC9F1EF0h
OPEN_REJECT (RESERVED ABANDON 2)	K28.5	D31.4	D07.3	D02.0	BC9F6702h
OPEN_REJECT (RESERVED ABANDON 3)	K28.5	D31.4	D01.4	D30.0	BC9F811Eh
OPEN_REJECT (RESERVED CONTINUE 0)	K28.5	D29.7	D02.0	D30.0	BCFD021Eh
OPEN_REJECT (RESERVED CONTINUE 1)	K28.5	D29.7	D24.0	D01.4	BCFD1881h
OPEN_REJECT (RESERVED INITIALIZE 0)	K28.5	D29.7	D30.0	D31.4	BCFD1E9Fh
OPEN_REJECT (RESERVED INITIALIZE 1)	K28.5	D29.7	D07.3	D16.7	BCFD67F0h
OPEN_REJECT (RESERVED STOP 0)	K28.5	D29.7	D31.4	D07.3	BCFD9F67h
OPEN_REJECT (RESERVED STOP 1)	K28.5	D29.7	D04.7	D27.4	BCFDE49Bh
OPEN_REJECT (RETRY)	K28.5	D29.7	D27.4	D24.0	BCFD9B18h
OPEN_REJECT (STP RESOURCES BUSY)	K28.5	D31.4	D27.4	D01.4	BC9F9B81h
OPEN_REJECT (WRONG DESTINATION)	K28.5	D31.4	D16.7	D24.0	BC9FF018h
OPEN_REJECT (ZONE VIOLATION)	K28.5	D31.4	D02.0	D27.4	BC9F029Bh
PS_ACK	K28.5	D16.7	D27.4	D30.0	BCF09B1Eh
PS_NAK	K28.5	D24.0	D27.4	D02.0	BC189B02h
PS_REQ (PARTIAL)	K28.5	D07.3	D02.0	D04.7	BC6702E4h
PS_REQ (SLUMBER)	K28.5	D30.0	D24.0	D02.0	BC1E1802h
SOAF	K28.5	D24.0	D30.0	D01.4	BC181E81h
TRAIN	K28.5	D30.3	D30.3	D30.3	BC7E7E7Eh
TRAIN_DONE	K28.5	D30.3	D30.3	D10.2	BC7E7E4Ah

Table 65 defines the primitive encodings for primitives used only inside SSP and SMP connections.

Table 65 — Primitive encoding for primitives used only inside SSP and SMP connections

Primitive	Character				Hexadecimal
	1 st	2 nd	3 rd	4 th (last)	
ACK	K28.5	D01.4	D01.4	D01.4	BC818181h
CREDIT_BLOCKED	K28.5	D01.4	D07.3	D30.0	BC81671Eh
DONE (ACK/NAK TIMEOUT)	K28.5	D30.0	D01.4	D04.7	BC1E81E4h
DONE (CREDIT TIMEOUT)	K28.5	D30.0	D07.3	D27.4	BC1E679Bh
DONE (NORMAL)	K28.5	D30.0	D30.0	D30.0	BC1E1E1Eh
DONE (RESERVED 0)	K28.5	D30.0	D16.7	D01.4	BC1EF081h
DONE (RESERVED 1)	K28.5	D30.0	D29.7	D31.4	BC1EFD9Fh
DONE (RESERVED TIMEOUT 0)	K28.5	D30.0	D27.4	D29.7	BC1E9BFDh
DONE (RESERVED TIMEOUT 1)	K28.5	D30.0	D31.4	D24.0	BC1E9F18h
EOF	K28.5	D24.0	D16.7	D27.4	BC18F09Bh
NAK (CRC ERROR)	K28.5	D01.4	D27.4	D04.7	BC819BE4h
NAK (RESERVED 0)	K28.5	D01.4	D31.4	D29.7	BC819FFDh
NAK (RESERVED 1)	K28.5	D01.4	D04.7	D24.0	BC81E418h
NAK (RESERVED 2)	K28.5	D01.4	D16.7	D07.3	BC81F067h
RRDY (NORMAL)	K28.5	D01.4	D24.0	D16.7	BC8118F0h
RRDY (RESERVED 0)	K28.5	D01.4	D02.0	D31.4	BC81029Fh
RRDY (RESERVED 1)	K28.5	D01.4	D30.0	D02.0	BC811E02h
SOF	K28.5	D24.0	D04.7	D07.3	BC18E467h

Table 66 lists the primitive encodings for primitives used only inside STP connections and on SATA physical links.

Table 66 — Primitive encoding for primitives used only inside STP connections and only on SATA physical links

Primitive	Character				Hexadecimal
	1 st	2 nd	3 rd	4 th (last)	
SATA_CONT	K28.3	D10.5	D25.4	D25.4	7CAA9999h
SATA_DMAT	K28.3	D21.5	D22.1	D22.1	7CB53636h
SATA_EOF	K28.3	D21.5	D21.6	D21.6	7CB5D5D5h
SATA_ERROR ^{a, b}	K28.6	D02.0	D01.4	D29.7	DC0281FDh
SATA_HOLD	K28.3	D10.5	D21.6	D21.6	7CAAD5D5h
SATA_HOLDA	K28.3	D10.5	D21.4	D21.4	7CAA9595h
SATA_PMACK	K28.3	D21.4	D21.4	D21.4	7C959595h
SATA_PMNAK	K28.3	D21.4	D21.7	D21.7	7C95F5F5h
SATA_PMREQ_P	K28.3	D21.5	D23.0	D23.0	7CB51717h
SATA_PMREQ_S	K28.3	D21.4	D21.3	D21.3	7C957575h
SATA_R_ERR	K28.3	D21.5	D22.2	D22.2	7CB55656h
SATA_R_IP	K28.3	D21.5	D21.2	D21.2	7CB55555h
SATA_R_OK	K28.3	D21.5	D21.1	D21.1	7CB53535h
SATA_R_RDY	K28.3	D21.4	D10.2	D10.2	7C954A4Ah
SATA_SOF	K28.3	D21.5	D23.1	D23.1	7CB53737h
SATA_SYNC	K28.3	D21.4	D21.5	D21.5	7C95B5B5h
SATA_WTRM	K28.3	D21.5	D24.2	D24.2	7CB55858h
SATA_X_RDY	K28.3	D21.5	D23.2	D23.2	7CB55757h
^a Except for SATA_ERROR, all values are defined by SATA. ^b Although included in this table, SATA_ERROR is not a primitive (see 3.1.146) since it starts with K28.6. It does not appear inside STP connections. It is an invalid dword, used by expander devices forwarding an error onto a SATA physical link (see 6.2.8.1).					

6.2.4 Primitive sequences

6.2.4.1 Primitive sequences overview

Table 67 summarizes the types of primitive sequences.

Table 67 — Primitive sequences

Primitive sequence type	Transmit ^a	Receive ^b	Reference
Single	1	1	6.2.4.2
Repeated	1 or more	1	6.2.4.3
Continued	2 followed by SATA_CONT	1	6.2.4.4
Extended	3	1	6.2.4.5
Triple	3	3	6.2.4.6
Redundant	6	3	6.2.4.7
^a Number of times the transmitter transmits the primitive to transmit the primitive sequence. ^b Number of times the receiver receives the primitive to detect the primitive sequence.			

Any number of deletable primitives may be transmitted inside primitive sequences without affecting the count or breaking the consecutiveness requirements unless prohibited by the primitives description (e.g., TRAIN, TRAIN_DONE). Rate matching deletable primitives shall be transmitted inside primitive sequences inside of connections if rate matching is enabled (see 6.14).

6.2.4.2 Single primitive sequence

Primitives labeled as single primitive sequences (e.g., RRDY, SATA_SOF) shall be transmitted one time to form a single primitive sequence.

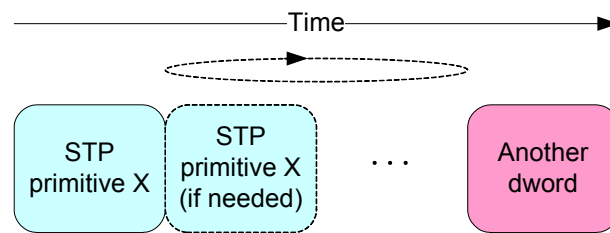
Receivers count each primitive received that is labeled as a single primitive sequence as a distinct single primitive sequence.

ALIGNs, NOTIFYs, and MUXs are deletable primitives (see 6.2.5 and 6.3).

6.2.4.3 Repeated primitive sequence

Primitives that form repeated primitive sequences (e.g., SATA_PMACK) shall be transmitted one or more times. Only STP primitives form repeated primitive sequences. Any number of deletable primitives may be transmitted inside repeated primitive sequences as described in 6.2.4.1.

Figure 89 shows an example of transmitting a repeated primitive sequence.

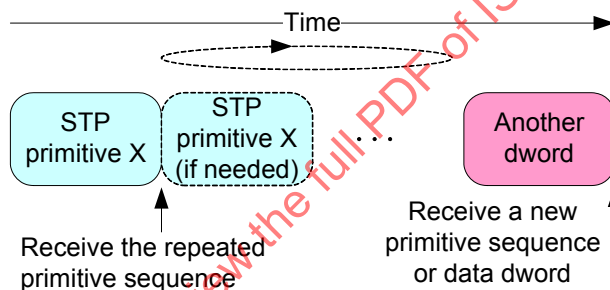


Note: Another dword is a dword other than a deletable primitive or STP primitive X

Figure 89 — Transmitting a repeated primitive sequence

Receivers do not count the number of times a repeated primitive is received. An expander device forwarding a repeated primitive sequence may transmit more repeated primitives than it receives (i.e., expand) or transmit fewer repeated primitives than it receives (i.e., contract). While transmitting a repeated primitive sequence, the expander device is considered to be originating (see 6.3.2) rather than forwarding (see 6.3.3) for purposes of deletable primitive insertion.

Figure 90 shows an example of receiving a repeated primitive sequence.



Note: Another dword is a dword other than a deletable primitive or STP primitive X

Figure 90 — Receiving a repeated primitive sequence

6.2.4.4 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be transmitted as specified in 6.18.3. Any number of deletable primitives may be transmitted inside continued primitive sequences as described in 6.2.4.1.

6.2.4.5 Extended primitive sequence

Primitives that form extended primitive sequences (e.g., AIP) shall be transmitted three times consecutively. Any number of deletable primitives may be transmitted inside extended primitive sequences as described in 6.2.4.1.

A receiver shall detect an extended primitive sequence after the primitive is received one time. The receiver shall process an extended primitive sequence the same as a single primitive sequence (see 6.2.4.2).

Figure 91 shows examples of extended primitive sequences.

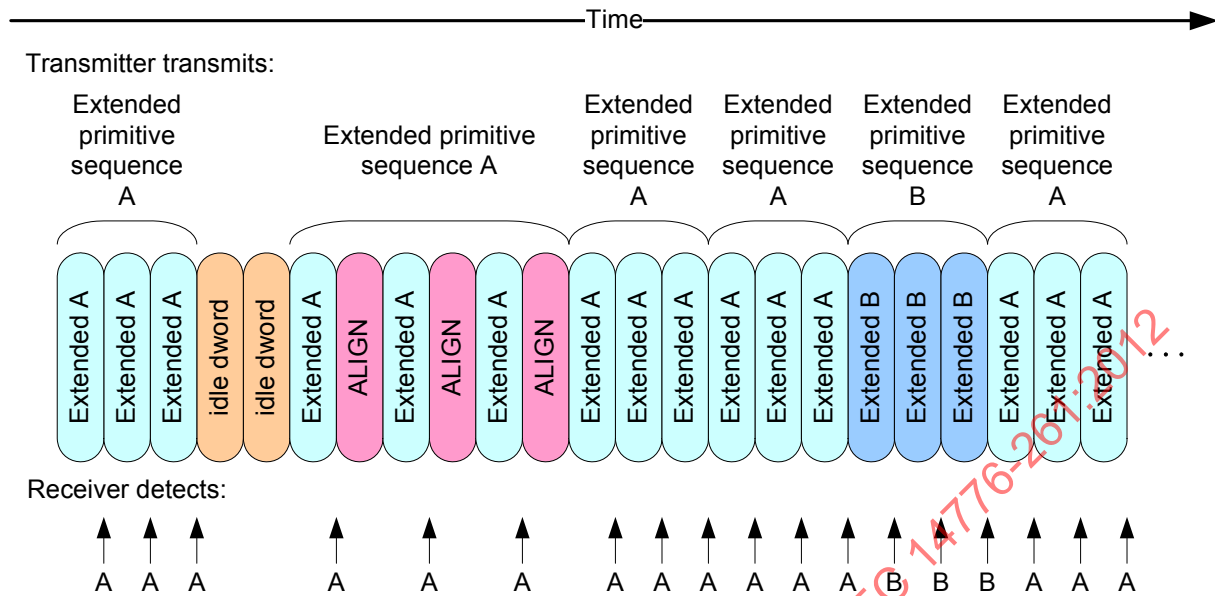


Figure 91 — Extended primitive sequences

6.2.4.6 Triple primitive sequence

Primitives that form triple primitive sequences (e.g., CLOSE (NORMAL)) shall be transmitted three times consecutively. Any number of deletable primitives may be transmitted inside triple primitive sequences as described in 6.2.4.1.

Receivers shall detect a triple primitive sequence after the identical primitive is received in three consecutive dwords. After detecting a triple primitive sequence, a receiver shall not detect a second instance of the same triple primitive sequence until it has received three consecutive dwords that are not any of the following:

- the original primitive; or
- a deletable primitive.

Figure 92 shows examples of triple primitive sequences.

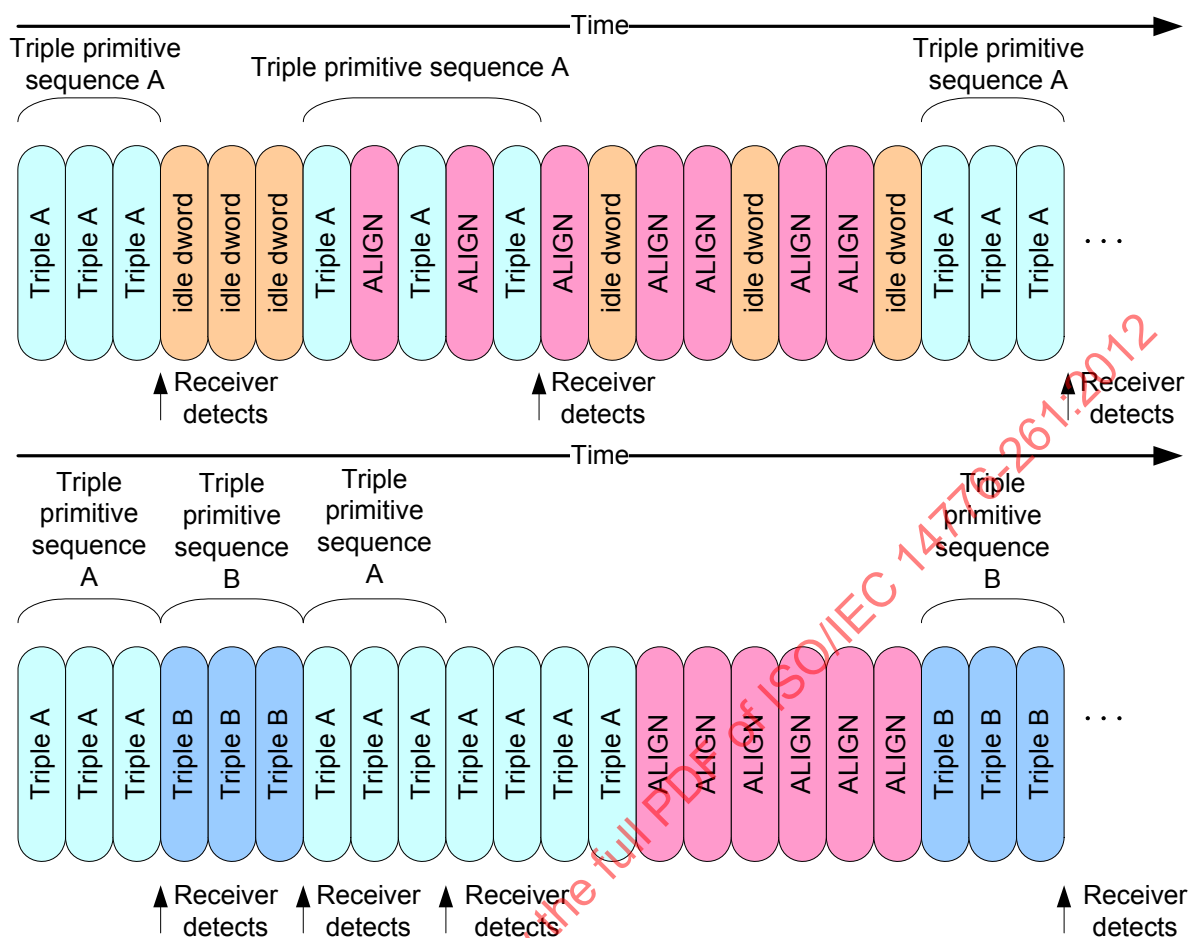


Figure 92—Triple primitive sequences

6.2.4.7 Redundant primitive sequence

Primitives that form redundant primitive sequences (e.g., BROADCAST (CHANGE)) shall be transmitted six times consecutively. Any number of deletable primitives may be transmitted inside redundant primitive sequences as described in 6.2.4.1 unless prohibited by the primitives description (e.g., TRAIN, TRAIN DONE).

A receiver shall detect a redundant primitive sequence after the identical primitive is received in any three of six consecutive dwords. After detecting a redundant primitive sequence, a receiver shall not detect a second instance of the same redundant primitive sequence until it has received six consecutive dwords that are not any of the following:

- a) the original primitive; or
- b) a deletable primitive.

Figure 93 shows examples of redundant primitive sequences.

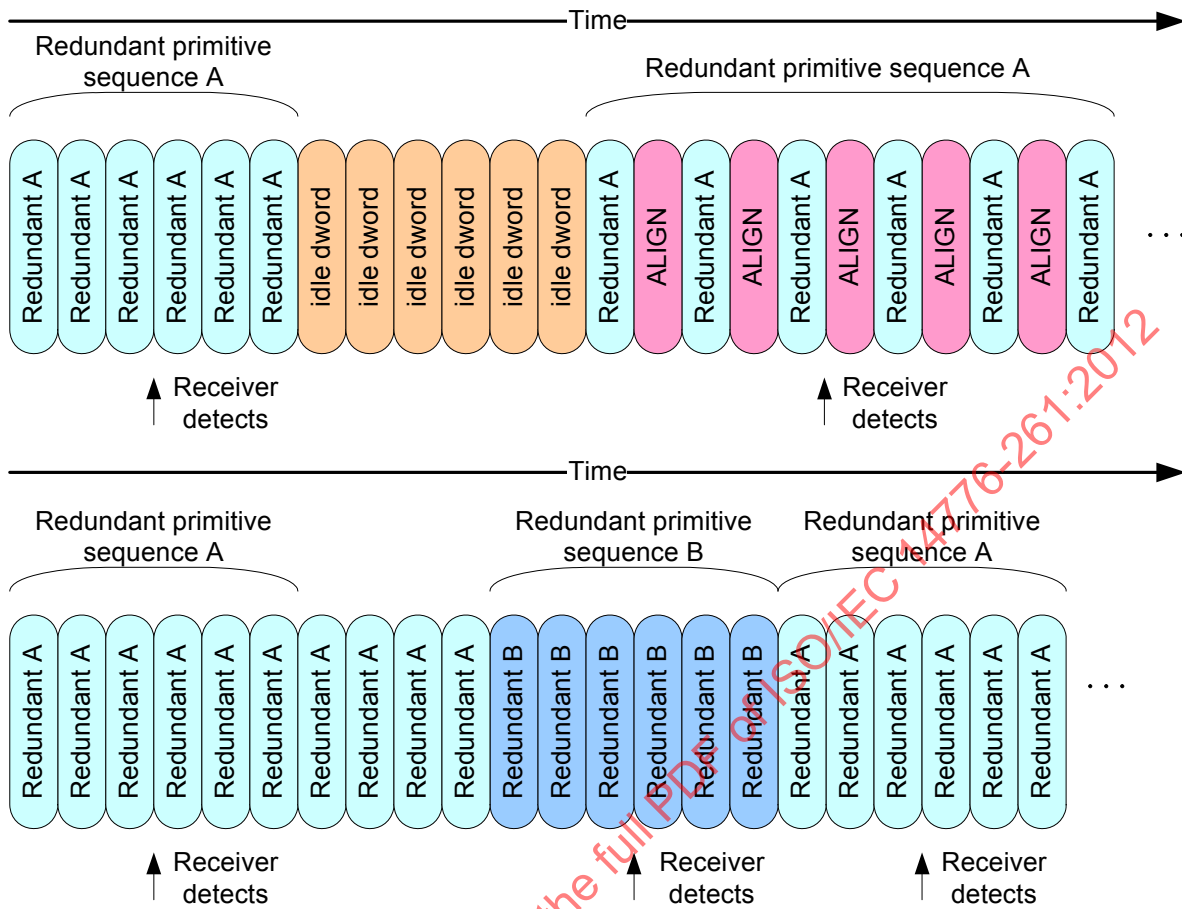


Figure 93 — Redundant primitive sequences

6.2.5 Deletable primitives

6.2.5.1 ALIGN

ALIGNs are used for:

- OOB signals (see 5.6);
- character and dword alignment during the speed negotiation sequence (see 5.7.4.2);
- physical link rate tolerance management after the phy reset sequence (see 6.3); and
- rate matching during connections (see 6.14).

ALIGNs are deletable primitives (see 6.3).

Table 68 defines the different versions of ALIGN primitives.

Table 68 — ALIGN primitives

Primitive	Description
ALIGN (0)	Used for OOB signals when D.C. mode is enabled, the speed negotiation sequence, physical link rate tolerance management, phy power condition sequence, and rate matching.
ALIGN (1)	Used for the speed negotiation sequence, physical link rate tolerance management, phy power condition sequence, and rate matching.
ALIGN (2)	Used for physical link rate tolerance management and rate matching.
ALIGN (3)	Used for OOB signals when optical mode is enabled, physical link rate tolerance management, and rate matching.

If D.C. mode is enabled, then phys may use ALIGN (0) to construct OOB signals (see 5.6 and SAS-2.1). If optical mode is enabled, then phys use ALIGN (3) to construct OOB signals (see 5.6 and SAS-2.1). Phys use ALIGN (0) and ALIGN (1) during the:

- a) speed negotiation sequence (see 5.7.4.2); and
- b) phy power condition sequence (see 5.8).

Phys shall rotate through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) for all ALIGNs transmitted after the phy reset sequence.

NOTE 24 ALIGN rotation is performed on a physical phy basis and is used to reduce radiated emissions.

Phys receiving ALIGNs after the phy reset sequence shall not verify the rotation and shall accept any of the ALIGNs at any time.

Phys shall only detect an ALIGN after decoding all four characters in the primitive.

NOTE 25 SATA devices are allowed to decode every dword starting with a K28.5 as an ALIGN, since ALIGN is the only primitive defined starting with K28.5.

For physical link rate tolerance management and rate matching, ALIGNs may be replaced by NOTIFYs (see 6.2.5.3) or MUXs (see 6.2.5.2). ALIGNs shall not be replaced by NOTIFYs or MUXs during OOB signals or speed negotiation.

6.2.5.2 MUX (Multiplex)

MUX is used if multiplexing (see 5.11) is enabled (see table 51 in 5.7.4.2.3.3) as follows:

- a) transmitted during the multiplexing sequence (see 5.7.4.3); and
- b) substituted in place of an ALIGN (see 6.2.5.1) being transmitted for physical link rate tolerance management (see 6.3) or rate matching (see 6.14) to confirm the logical link number as defined in 5.11.

Substitution of a MUX for an ALIGN may or may not affect the ALIGN rotation (i.e., the MUX may take the place of one of the ALIGNs in the rotation through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3), or may delay the rotation).

MUXs are deletable primitives (see 6.3). A phy supporting multiplexing shall process MUX primitives in logic running off the received clock without using an elasticity buffer rather than logic after the elasticity buffer, because they are not accompanied by additional deletable primitives (e.g., ALIGNs and/or NOTIFYs).

The versions of MUX are defined in table 69.

Table 69 — MUX primitives

Primitive	Description
MUX (LOGICAL LINK 0)	Establishes the position of dwords in logical link 0.
MUX (LOGICAL LINK 1)	Establishes the position of dwords in logical link 1.

See 5.11 for details on multiplexing.

6.2.5.3 NOTIFY

6.2.5.3.1 NOTIFY overview

NOTIFY may be substituted in place of any ALIGN (see 6.2.5.1) being transmitted for physical link rate tolerance management (see 6.3) or rate matching (see 6.14). Substitution of a NOTIFY in place of an ALIGN may or may not affect the ALIGN rotation (i.e., the NOTIFY may take the place of one of the ALIGNs in the rotation through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3), or may delay the rotation). A specific NOTIFY shall not be transmitted in more than three consecutive dwords until at least three other dwords have been transmitted.

NOTIFYs are deletable primitives (see 6.3). If a phy supports a specific NOTIFY primitive, then the phy should decode that NOTIFY in logic running off the received clock without using an elasticity buffer rather than logic after the elasticity buffer to avoid missing detection of important information.

NOTIFY shall not be forwarded through expander devices. Expander devices shall substitute an ALIGN for a NOTIFY if necessary.

SAS target devices are not required to detect every transmitted NOTIFY.

The versions of NOTIFY representing different reasons are defined in table 70.

Table 70 — NOTIFY primitives

Primitive	Description	Reference
NOTIFY (ENABLE SPINUP)	Specify to a SAS target device that it may temporarily consume additional power (see 9.2.10).	6.2.5.3.2
NOTIFY (POWER LOSS EXPECTED)	Specify to a SAS target device that power loss may occur within the time specified by the POWER LOSS TIMEOUT field in the Shared Port Control mode page (see 9.2.7.6).	6.2.5.3.3
NOTIFY (RESERVED 1)	Reserved.	

NOTIFY (RESERVED 1) shall be ignored by all devices.

6.2.5.3.2 NOTIFY (ENABLE SPINUP)

NOTIFY (ENABLE SPINUP) is transmitted by a SAS initiator port or expander port and is used to specify to a SAS target device that it may temporarily consume additional power (e.g., to spin up rotating media) (see 9.2.10). The length of time the SAS target device consumes additional power and the amount of additional power is vendor specific. NOTIFY (ENABLE SPINUP) shall interact with the device's power condition state transitions, controlled by the Power Conditions mode page (see SPC-4) and/or the START STOP UNIT command (see SBC-3), as described in 9.2.10.

SAS initiator devices and expander devices shall use NOTIFY (ENABLE SPINUP) while attached to SSP target devices (i.e., devices that report SSP target port support in their IDENTIFY address frames). SAS initiator ports and expander ports shall transmit one NOTIFY (ENABLE SPINUP) after power on when the enclosure is ready for initial temporary consumption of additional power. After the initial NOTIFY (ENABLE SPINUP), SAS initiator ports and expander ports shall transmit NOTIFY (ENABLE SPINUP) periodically. Otherwise, the selection of when and how often to transmit NOTIFY (ENABLE SPINUP) is outside the scope of this standard.

NOTE 26 The SAS initiator device or expander device uses NOTIFY (ENABLE SPINUP) to avoid exceeding enclosure power supply capabilities during temporary consumption of additional power by multiple SAS target devices. It may choose to rotate transmitting NOTIFY (ENABLE SPINUP) across all of its ports, distributing it to N ports at a time if the enclosure power supply is capable of powering N SAS target devices that are temporarily consuming additional power at the same time. An expander device may allow this timing to be configured by an NVRAM programmed with enclosure-specific sequencing patterns, or may employ more complex, dynamic interaction with the enclosure power supply.

NOTE 27 NOTIFY (ENABLE SPINUP) should be transmitted as frequently as possible to avoid incurring SCSI application layer timeouts.

A SAS target device with multiple SAS target ports shall process a NOTIFY (ENABLE SPINUP) received by any of its SAS target ports (e.g., if a dual-port SAS target device, that powers on in the Stopped state receives a START STOP UNIT command with the START bit set to one through one SSP target port followed by a receipt of a NOTIFY (ENABLE SPINUP) on the other SSP target port, then the SAS target device may temporarily consume additional power (see 9.2.10)).

6.2.5.3.3 NOTIFY (POWER LOSS EXPECTED)

NOTIFY (POWER LOSS EXPECTED) is transmitted by a SAS initiator port or expander port and is used to specify to a SAS target device that power loss may occur within the time specified in the POWER LOSS TIMEOUT field in the Shared Port Control mode page (see 9.2.7.6).

NOTIFY (POWER LOSS EXPECTED) shall be transmitted at least three times by the SAS initiator port or expander port.

If a SAS target device supports NOTIFY (POWER LOSS EXPECTED) and receives NOTIFY (POWER LOSS EXPECTED) on an SSP target port, then:

- a) the device server for each logical unit to which the SSP target port has access shall:
 - 1) stop writing data to the media as soon as possible without creating read errors for future reads (e.g., on a direct-access block device, a physical block boundary is reached);
 - 2) clear all task sets as defined in SAM-5; and
 - 3) establish a unit attention condition for the initiator port associated with every I_T nexus as defined in SAM-5 (e.g., with the additional sense code set to COMMANDS CLEARED BY POWER LOSS NOTIFICATION);

and
- b) the SAS target device shall:
 - A) on each phy that receives NOTIFY (POWER LOSS EXPECTED), if there is an SSP connection, then transmit a BREAK primitive sequence on that connection; and
 - B) on each phy that does not receive NOTIFY (POWER LOSS EXPECTED), if there is an SSP connection, then transmit a BREAK primitive sequence or a CLOSE primitive sequence on that connection.

If the SAS target device receives any frames after receiving NOTIFY (POWER LOSS EXPECTED) before a connection is closed, then it should discard the received frames.

The SCSI application layer that receives a Power Loss Expected event shall:

- a) start the power loss timer;
- b) send an Accept_Reject OPENs (Reject SSP) request to all ST_T state machines (i.e., all SSP connection requests result in OPEN_REJECT (RETRY));

- c) if a SCSI Command Received transport protocol service indication is received, then the device server shall abort that command and send an Accept_Reject OPENs (Reject SSP) request to the ST_T state machine on which the SCSI Command Received transport protocol service indication was received; and
- d) if the power loss timeout timer expires, then the SCSI application layer shall send an Accept_Reject OPENs (Accept SSP) request to all ST_T state machines.

After power on, the power loss timeout timer shall be initialized and stopped until a NOTIFY (POWER LOSS EXPECTED) is received.

6.2.5.4 OOB_IDLE

OOB_IDLE (see 5.9.4.2 and SAS-2.1) is used when a phy is in optical mode for:

- a) OOB signals; and
- b) speed negotiation.

6.2.6 Primitives not specific to type of connections

6.2.6.1 AIP (Arbitration in progress)

AIP is transmitted by an expander device after a connection request to specify that the connection request is being processed and specify the status of the connection request.

The versions of AIP representing different statuses are defined in table 71.

Table 71 — AIP primitives

Primitive	Description
AIP (NORMAL)	Expander device has accepted the connection request. This primitive may be transmitted multiple times (see 6.13.4.3).
AIP (RESERVED 0)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 1)	
AIP (RESERVED 2)	
AIP (WAITING ON CONNECTION)	Expander device has determined the routing for the connection request, but either the destination phys are all being used for connections, or there are insufficient routing resources to complete the connection request. This may be transmitted multiple times (see 6.13.4.3).
AIP (WAITING ON DEVICE)	Expander device has determined the routing for the connection request and forwarded the request to the output physical link. This is transmitted one time (see 6.13.4.3).
AIP (WAITING ON PARTIAL)	Expander device has determined the routing for the connection request, but the destination phys are all busy with other partial pathways. This primitive may be transmitted multiple times (see 6.13.4.3).
AIP (RESERVED WAITING ON PARTIAL)	Reserved. Processed the same as AIP (WAITING ON PARTIAL).

See 6.13 for details on connections.

6.2.6.2 BREAK

BREAK is used to abort a connection request or break a connection.

See 6.13.6 and 6.13.8 for details on breaking connections.

6.2.6.3 BREAK_REPLY

BREAK_REPLY is used to confirm the receipt of a BREAK.

See 6.13.6 and 6.13.8 for details on breaking connections.

6.2.6.4 BROADCAST

BROADCASTs are used to notify SAS ports and expander devices in a SAS domain about certain events.

The versions of BROADCAST representing different Broadcasts (see table 7 in 4.1.13) are defined in table 72.

Table 72 — BROADCAST primitives

Primitive	Description
BROADCAST (CHANGE)	Broadcast (Change)
BROADCAST (RESERVED CHANGE 0)	Broadcast (Reserved Change 0)
BROADCAST (RESERVED CHANGE 1)	Broadcast (Reserved Change 1)
BROADCAST (SES)	Broadcast (SES)
BROADCAST (EXPANDER)	Broadcast (Expander)
BROADCAST (ASYNCHRONOUS EVENT)	Broadcast (Asynchronous Event)
BROADCAST (RESERVED 3)	Broadcast (Reserved 3)
BROADCAST (RESERVED 4)	Broadcast (Reserved 4)

A phy that has not completed the link reset sequence shall not transmit a BROADCAST. A phy shall not transmit a BROADCAST inside a connection. A phy shall ignore any BROADCAST received inside a connection.

A BROADCAST received by a phy that has not completed the link reset sequence shall be ignored.

6.2.6.5 CLOSE

CLOSE is used to close a connection. This primitive may be originated by a SAS initiator port or a SAS target port.

The versions of CLOSE representing different reasons are defined in table 73

Table 73 — CLOSE primitives

Primitive	Description
CLOSE (CLEAR AFFILIATION)	Close an open STP connection and clear the affiliation (see 6.18.4). Processed the same as CLOSE (NORMAL) if: a) the connection is not an STP connection; b) the connection is an STP connection, but affiliations are not implemented by the STP target port; or c) the connection is an STP connection, but an affiliation is not present.
CLOSE (NORMAL)	Close a connection.
CLOSE (RESERVED 0)	Reserved. Processed the same as CLOSE (NORMAL).
CLOSE (RESERVED 1)	

See 6.13.7 for details on closing connections.

6.2.6.6 EOAF (End of address frame)

EOAF specifies the end of an address frame.

See 6.8 for details on address frames.

6.2.6.7 ERROR

ERROR should be transmitted by an expander device while the expander device is forwarding dwords from a SAS physical link or SATA physical link to a SAS physical link and receives an invalid dword or an ERROR.

NOTE 28 Since an 8b10b coding error in one dword is sometimes not detected until the next dword (see table 43 in 5.3.9), expander devices should avoid deleting invalid dwords or ERRORs unless necessary (e.g., if the elasticity buffer is full) to avoid hiding evidence that an error has occurred.

See 6.16 for details on error handling by expander devices.

6.2.6.8 HARD_RESET

HARD_RESET is used to force a phy to generate a hard reset (see 4.4.2) to its port. This primitive is only valid after the phy reset sequence without an intervening identification sequence (see 4.4) and shall be ignored at other times.

6.2.6.9 OPEN_ACCEPT

OPEN_ACCEPT specifies the acceptance of a connection request.

See 6.13 for details on connection requests.

6.2.6.10 OPEN_REJECT

OPEN_REJECT specifies that a connection request has been rejected and specifies the reason for the rejection. The result of some OPEN_REJECTs is to abandon (i.e., not retry) the connection request, and the result of other OPEN_REJECTs is to retry the connection request.

All of the OPEN_REJECT versions defined in table 74 shall result in the originating port abandoning the connection request.

Table 74 — Abandon-class OPEN_REJECT primitives

Primitive	Originator	Description
OPEN_REJECT (BAD DESTINATION)	Expander phy	A connection request routes to a destination expander phy in the same expander port as the source expander phy, and the expander port is using the direct routing method (see 4.6.7.1).
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	Any phy	The requested connection rate is not supported on some physical link on the pathway between the source phy and destination phy. When a SAS initiator phy is directly attached to a SAS target phy, the requested connection rate is not supported by the destination phy. If the connection rate is 1.5 Gbps, then this shall be considered an abandon-class OPEN_REJECT. If the connection rate is greater than 1.5 Gbps, then the connection request shall be modified and reattempted as described in 6.8.3.
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	Any phy	Phy with destination SAS address exists, but the destination phy does not support the requested initiator role, target role, protocol, initiator connection tag, or features (i.e., the values in the INITIATOR PORT bit, the PROTOCOL field, the INITIATOR CONNECTION TAG field, and/or the FEATURES field in the OPEN address frame are not supported).
OPEN_REJECT (RESERVED ABANDON 1)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 2)		
OPEN_REJECT (RESERVED ABANDON 3)		
OPEN_REJECT (STP RESOURCES BUSY)	Destination phy	STP target port with destination SAS address exists, but the STP target port supports affiliations and is not able to establish an affiliation with this STP initiator port (e.g., because it has reached its maximum number of affiliations), or the STP target port does not support affiliations and all of the available affiliation contexts have been allocated to other STP initiator ports (see 6.18.4). Process the same as OPEN_REJECT (WRONG DESTINATION) for non-STP connection requests.
OPEN_REJECT (WRONG DESTINATION)	Destination phy	The destination SAS address does not match the SAS address of the SAS port to which the connection request was delivered.
OPEN_REJECT (ZONE VIOLATION)	Zoning expander phy	The connection request is from a zone group that does not have permission to access the zone group that contains the destination phy according to the zone permission table of an unlocked zoning expander device.

All of the OPEN_REJECT versions defined in table 75 shall result in the originating port retrying the connection request.

Table 75 — Retry-class OPEN_REJECT primitives

Primitive	Originator	Description
OPEN_REJECT (NO DESTINATION) ^a	Expander phy	An expander device in the pathway is not configuring (see 4.7.4) and determines that: a) there is no such destination phy; b) the connection request routes to a destination expander phy in the same expander port as the source expander phy and the expander port is using the subtractive routing method; or c) the SAS address is valid for an STP target port in an STP/SATA bridge, but the initial Register - Device to Host FIS has been received with an error (see 9.4.3.12).
OPEN_REJECT (PATHWAY BLOCKED) ^b	Expander phy	An expander device determined the pathway was blocked by higher priority connection requests.
OPEN_REJECT (RESERVED CONTINUE 0) ^c	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED CONTINUE 1) ^c		
OPEN_REJECT (RESERVED INITIALIZE 0) ^a	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED INITIALIZE 1) ^a		
OPEN_REJECT (RESERVED STOP 0) ^b	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED).
OPEN_REJECT (RESERVED STOP 1) ^b		
OPEN_REJECT (RETRY) ^c	Destination phy or zoning expander phy	Either: a) a phy with the destination SAS address exists but is temporarily not able to accept connections (see 6.17.1, 6.18.5, and 6.19.3); b) an expander device in the pathway is configuring (see 4.7.4) and would otherwise have returned OPEN_REJECT (NO DESTINATION) (see 4.7.4 and 6.13.4.2.5); c) an expander device in the pathway is locked and would otherwise have returned OPEN_REJECT (ZONE VIOLATION) (see 4.9.3.5 and 6.13.4.2.5); d) an expander device in the pathway has reduced functionality (see 4.6.8 and 6.13.4.2.5); or e) a phy with the destination SAS address exists but is in the slumber phy power condition (see 4.10.1.4).
^a If the I_T Nexus Loss timer is already running, it continues running. If it is not already running, it is initialized and started. Stop retrying the connection request if the I_T Nexus Loss timer expires. ^b If the I_T Nexus Loss timer is already running, it continues running. Stop retrying the connection request if the I_T Nexus Loss timer expires. ^c If the I_T Nexus Loss timer (see 7.2.2) is already running, it is stopped.		

NOTE 29 Some SAS logical phys compliant with earlier versions of this standard also transmit OPEN_REJECT (RETRY) if they receive an OPEN address frame while their SL_CC state machines are in the SL_CC5:BreakWait state (see 6.15.4.7).

When a SAS logical phy detects more than one reason to transmit an OPEN_REJECT, the SL_CC state machine determines the priority in the SL_CC2:Selected state (see 6.15.4.4).

When an expander logical phy detects more than one reason to transmit an OPEN_REJECT, the ECM determines the priority (see 6.13.4).

See 6.13 for details on connection requests.

6.2.6.11 PS_ACK

PS_ACK specifies the positive acknowledgement of a PS_REQ.

6.2.6.12 PS_NAK

PS_NAK specifies the negative acknowledgement of a PS_REQ.

A PS_NAK occurs for the following reasons:

- a) a PS_REQ is received and the requested low phy power condition is supported and disabled; or
- b) a PS_REQ is received during an attempt to establish a connection.

6.2.6.13 PS_REQ

PS_REQ is used to request a transition to a specific low phy power condition (see 4.10.1.1).

All versions of PS_REQ representing different low phy power conditions are defined in table 76.

Table 76 — PS_REQ primitives

Primitive	Description
PS_REQ (PARTIAL)	Requests a transition into the partial phy power condition (see 4.10.1.3).
PS_REQ (SLUMBER)	Requests a transition into the slumber phy power condition (see 4.10.1.4).

6.2.6.14 SOAF (Start of address frame)

SOAF specifies the start of an address frame.

See 6.8 for details on address frames.

6.2.6.15 TRAIN

TRAIN is used during Train-SNW during speed negotiation. Deletable primitives shall not be transmitted inside a TRAIN primitive sequence.

See 5.7.4.2.3.4 for details on Train-SNW.

6.2.6.16 TRAIN_DONE

TRAIN_DONE is used during Train-SNW during speed negotiation. Deletable primitives shall not be transmitted inside a TRAIN_DONE primitive sequence.

See 5.7.4.2.3.4 for details on Train-SNW.

6.2.7 Primitives used only inside SSP and SMP connections

6.2.7.1 ACK (Acknowledge)

ACK specifies the positive acknowledgement of an SSP frame.

See 6.17.3 for details on SSP frame transmission.

6.2.7.2 CREDIT_BLOCKED

CREDIT_BLOCKED specifies that no more RRDYs are going to be transmitted during this connection (i.e., credit is not going to be increased).

See 6.17.4 for details on SSP flow control.

6.2.7.3 DONE

DONE is used to start closing an SSP connection and specify a reason for doing so. This primitive may be originated by an SSP initiator port or an SSP target port. DONE is not used during an SMP or STP connection.

The versions of DONE representing different reasons are defined in table 77. The SSP state machines describe when these are used (see 6.17.8).

Table 77 — DONE primitives

Primitive	Description
DONE (ACK/NAK TIMEOUT)	The SSP state machines (see 6.17.8) timed out waiting for an ACK or NAK, and the phy is going to transmit BREAK primitive sequence in 1 ms unless DONE is received within 1 ms of transmitting the DONE (ACK/NAK TIMEOUT).
DONE (RESERVED TIMEOUT 0)	Reserved. Processed the same as DONE (ACK/NAK TIMEOUT).
DONE (RESERVED TIMEOUT 1)	
DONE (NORMAL)	Finished transmitting all frames.
DONE (RESERVED 0)	Reserved. Processed the same as DONE (NORMAL).
DONE (RESERVED 1)	
DONE (CREDIT TIMEOUT)	The SSP state machines (see 6.17.8) timed out waiting for an RRDY or received a CREDIT BLOCKED, and the phy is going to transmit BREAK primitive sequence if it provides transmit frame credit for 1 ms without receiving a frame or a DONE.

See 6.17.7 for details on closing SSP connections.

6.2.7.4 EOF (End of frame)

EOF specifies the end of an SSP or SMP frame.

See 6.17.3 for details on SSP frame transmission and 6.19.1 for details on SMP frame transmission.

6.2.7.5 NAK (Negative acknowledgement)

NAK specifies the negative acknowledgement of an SSP frame and the reason for doing so.

The versions of NAK representing different reasons are defined in table 78.

Table 78 — NAK primitives

Primitive	Description
NAK (CRC ERROR)	The frame had a bad CRC, an invalid dword, or an ERROR was received during frame reception.
NAK (RESERVED 0)	Reserved. Processed the same as NAK (CRC ERROR).
NAK (RESERVED 1)	
NAK (RESERVED 2)	

See 6.17.3 for details on SSP frame transmission.

6.2.7.6 RRDY (Receiver ready)

RRDY is used to increase SSP frame credit.

The versions of RRDY representing different reasons are defined in table 79.

Table 79 — RRDY primitives

Primitive	Description
RRDY (NORMAL)	Increase transmit frame credit by one.
RRDY (RESERVED 0)	Reserved. Processed the same as RRDY (NORMAL).
RRDY (RESERVED 1)	

A phy shall not transmit RRDY after transmitting CREDIT_BLOCKED in a connection. See 6.17.4 for details on SSP flow control.

6.2.7.7 SOF (Start of frame)

SOF specifies the start of an SSP or SMP frame.

See 6.17.3 for details on SSP frame transmission and 6.19.1 for details on SMP frame transmission.

6.2.8 Primitives used only inside STP connections and on SATA physical links

6.2.8.1 SATA_ERROR

SATA_ERROR should be transmitted by an expander device when it is forwarding dwords from a SAS logical link to a SATA physical link and it receives an invalid dword or an ERROR.

NOTE 30 Since an 8b10b coding error in one dword is sometimes not detected until the next dword (see table 43 in 5.3.9), expander devices should avoid deleting invalid dwords or ERRORs unless necessary (e.g., if the elasticity buffer is full) to avoid hiding evidence that an error has occurred.

See 5.10 for details on error handling by expander devices.

Although included in this subclause, SATA_ERROR is not a primitive (see 3.1.146) since it starts with K28.6. SATA_ERROR does not appear inside STP connections. SATA_ERROR is an invalid dword.

6.2.8.2 SATA_PMACK, SATA_PMNAK, SATA_PMREQ_P, and SATA_PMREQ_S (Power management acknowledgements and requests)

SATA_PMREQ_P and SATA_PMREQ_S request entry into the partial interface power management sequence and slumber interface power management sequence (see SATA). SATA_PMACK is used to accept a power management request. SATA_PMNAK is used to reject an interface power management request.

See 6.11 for rules on handling the power management primitives.

6.2.8.3 SATA_HOLD and SATA_HOLD_A (Hold and hold acknowledge)

See 6.18.2 for rules on STP flow control, which uses SATA_HOLD and SATA_HOLD_A.

6.2.8.4 SATA_R_RDY and SATA_X_RDY (Receiver ready and transmitter ready)

When a SATA port has a frame to transmit, it transmits SATA_X_RDY and waits for SATA_R_RDY before transmitting the frame.

6.2.8.5 Other primitives used inside STP connections and on SATA physical links

Other primitives used in STP connections and on SATA physical links are defined in SATA.

6.3 Physical link rate tolerance management

6.3.1 Physical link rate tolerance management overview

A phy may have three clocks:

- a) an internal clock (e.g., based on a PLL clock generator);
- b) a transmit clock (e.g., based on a PLL clock generator with SSC, if SSC is enabled). Used when transmitting dwords on the physical link; and
- c) a receive clock, derived from the input bit stream. Used when receiving dwords from the physical link.

Although the receive clock has the same nominal fixed frequency as the internal clock, the receive clock may differ from the internal clock frequency up to the physical link rate tolerance (see SAS-2.1). Over time:

- a) if the input clock is faster than the internal clock, an overrun occurs if the phy receiver receives a dword and is not able to forward it to an internal receive buffer; or
- b) if the input clock is slower than the internal clock, an underrun occurs if the phy receiver is not able to obtain a dword from an internal transmit buffer when needed.

To avoid overruns and underruns, phy transmitters insert deletable primitives (see 6.2.5) in the dword stream. Phy receivers may pass deletable primitives through to their internal buffers, or may strip them out when an overrun occurs. Phy receivers add deletable primitives when an underrun occurs. The internal logic shall ignore all deletable primitives that arrive in the internal buffers.

Circuitry (e.g., an elasticity buffer) is required to absorb the slight differences in frequencies between the phys. Figure 94 shows an example of an elasticity buffer. The frequency tolerance for a phy is specified in SAS-2.1. The depth of the elasticity buffer is vendor specific but shall accommodate the physical link rate tolerance management deletable primitive insertion requirements in table 80 (see 6.3.2).

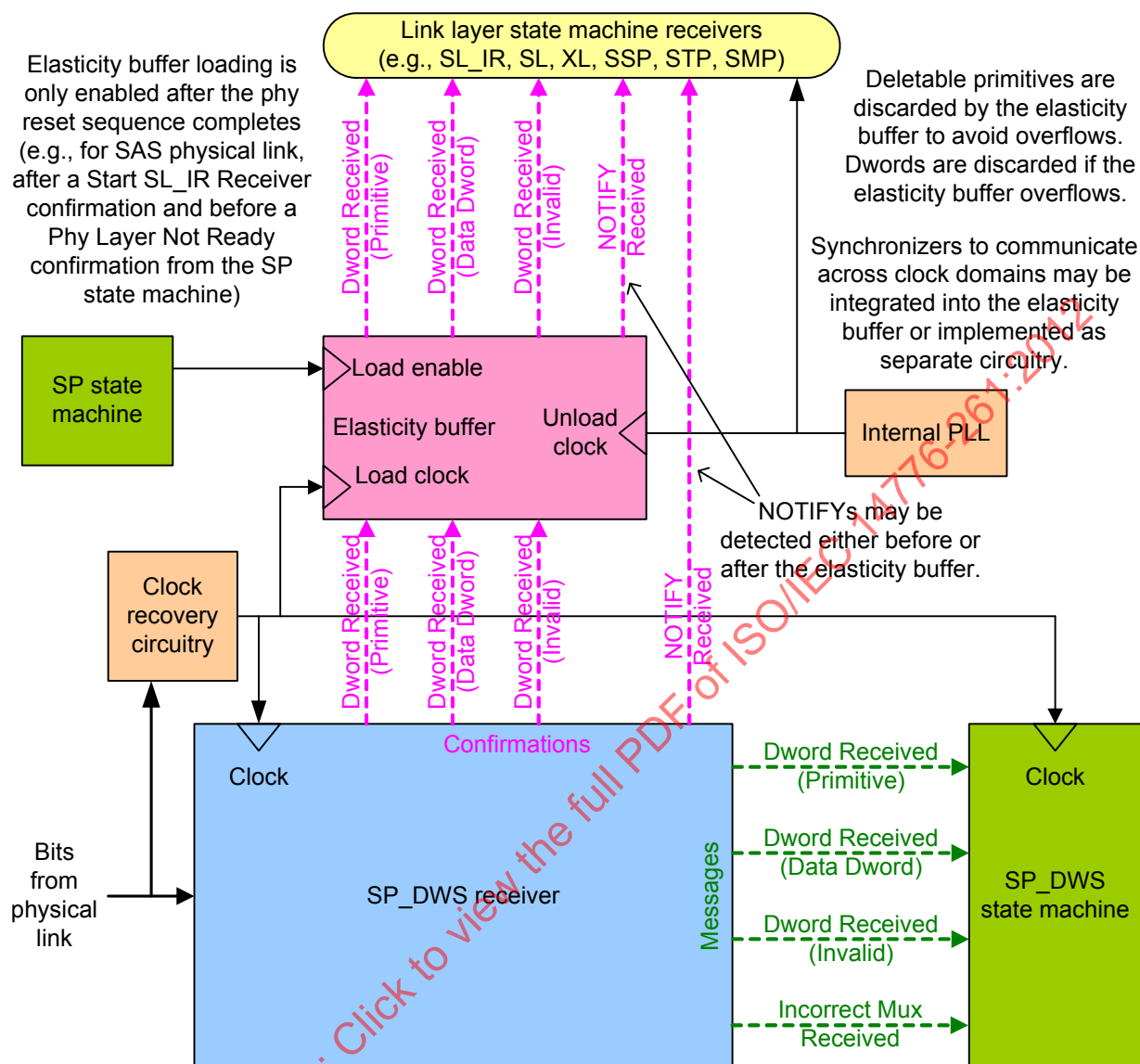


Figure 94 — Elasticity buffer

6.3.2 Phys originating dwords

A logical phy that is originating dwords (i.e., a logical phy that is not an expander logical phy forwarding dwords from another expander logical phy) shall only insert deletable primitives for physical link rate tolerance management after the phy reset sequence (see 5.7) completes as described in table 80.

Table 80 — Physical link rate tolerance management deletable primitive insertion requirement

Physical link rate	Requirement ^a
1.5 Gbps	One deletable primitive within every 128 dwords ^{b, c}
3 Gbps	Two deletable primitives within every 256 dwords ^{b, d}
6 Gbps	Four deletable primitives within every 512 dwords ^b
^a These numbers account for the worst case clock frequency differences between the fastest phy transmitter and the slowest phy receiver (e.g., a center-spreading expander phy originating dwords in an STP connection at +2 400 ppm that are forwarded to a down-spreading SATA device with an internal clock at -5 350 ppm). The difference of 7 750 ppm (i.e., 0.775 % or 1/129) is less than the deletable primitive insertion rate of 1/128 (i.e., 7 813 ppm or 0.781 25 %), ensuring there are enough deletable primitives for the phy receiver to delete without having to buffer dwords. ^b 128 dwords at 1.5 Gbps, 256 dwords at 3 Gbps, and 512 dwords at 6 Gbps are each nominally 3 413.3 ns. ^c Phys compliant with SAS-1.1 were required to insert one deletable primitive within every 2 048 dwords at 1.5 Gbps. ^d Phys compliant with SAS-1.1 were required to insert two deletable primitives within every 4 096 dwords at 3 Gbps.	

Deletable primitives inserted for physical link rate tolerance management are in addition to deletable primitives inserted for rate matching (see 6.14). See Annex G for a summary of their combined requirements.

See 6.2.5.1 for details on rotating through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3). NOTIFYs may also be transmitted in place of ALIGNs (see 6.2.5.3) on SAS logical links. MUXs may also be transmitted in place of ALIGNs on multiplexed SAS physical links.

6.3.3 Expander phys forwarding dwords

An expander device that is forwarding dwords (i.e., is not originating dwords) is allowed to insert or delete as many deletable primitives as required to match the transmit and receive connection rates. An expander device is not required to transmit the number of deletable primitives for physical link rate tolerance management described in table 80 when forwarding dwords to a SAS logical link. An expander device shall increase or reduce the number of deletable primitives based on clock frequency differences between the expander device's receiving phy and the expander device's transmitting phy (e.g., if receiving at -100 ppm and transmitting at +100 ppm, then it transmits more deletable primitives than it receives).

The expander device is also required to insert deletable primitives for rate matching (see 6.14). During an STP connection, the expander device shall:

- preserve the incoming rate of any additional deletable primitives that it receives that are not discarded because of physical link rate tolerance management or rate matching (e.g. the 1/128 deletable primitives received from an originating STP initiator phy compliant with SAS-1.1 for STP initiator phy throttling); or
- transmit one deletable primitive within every 128 dwords, without discarding any data dwords or primitives.

The expander device may reduce the length of repeated primitive sequences (i.e., primitive, SATA_CONT, and data dword sequences).

NOTE 31 One possible implementation for expander devices forwarding dwords is for the expander device to delete all deletable primitives received and to insert deletable primitives at the transmit phy whenever its elasticity buffer is empty.

The STP target port of an STP/SATA bridge is allowed to insert or delete as many deletable primitives as required to match the transmit and receive connection rates. It is not required to transmit any particular number of deletable primitives for physical link rate tolerance management when forwarding to a SAS logical link and is not required to ensure that any deletable primitives it transmits are in pairs.

NOTE 32 Due to physical link rate tolerance management deletable primitive removal, the STP target port may not receive a pair of deletable primitives every 256 dwords, even if the STP initiator port transmitted them in pairs. However, the rate of the dword stream allows for deletable primitive insertion by the STP/SATA bridge. One possible implementation is for the STP/SATA bridge to delete all deletable primitives received by the STP target port and to insert two consecutive ALIGNs at the SATA host port when its elasticity buffer is empty or when 254 non-ALIGN dwords have been transmitted. It may need to buffer up to 2 dwords concurrently being received by the STP target port while it does so.

An expander device supporting the SSC modulation type of center-spreading also includes a center-spreading tolerance buffer (see SAS-2.1).

6.4 Idle physical links

Idle dwords are vendor specific data dwords which are scrambled (see 6.6).

Phys shall transmit idle dwords if there are no other dwords to transmit and:

- a) no connection is open; or
- b) an SSP or SMP connection is open.

SATA_SYNC is a continued primitive sequence which may contain vendor specific data dwords (see 6.2.4.4) which are scrambled (see 6.6) during an STP connection.

6.5 CRC

6.5.1 CRC overview

All frames include cyclic redundancy check (CRC) values to help detect transmission errors.

Frames transmitted in an STP connection shall include a CRC as defined by SATA. Address frames, SSP frames, and SMP frames shall include a CRC as defined by this standard.

Annex C contains information on CRC generation/checker implementation.

Table 81 defines notation used in the following text describing CRC calculation. Arithmetic is modulo 2.

Table 81 — CRC notation and definitions

Notation	Definition
$T(i)$	A transformation over the non-negative integers (i.e., Z^+): $T(i) = i + 7 - 2 \times (i \bmod 8)$, $i \geq 0$, $i \in Z^+$
$F(x)$	A polynomial representing the bits covered by the CRC: $F(x) = b_0x^{(k-1)} + b_1x^{(k-2)} + \dots + b_{(k-2)}x + b_{(k-1)}$ where: k is the number of bits b_i describes a bit, where the bit index i denotes that bit b_i is more significant than bit $b_{(i+1)}$ For example, if the frame, except for the CRC field, contains one data dword set to 516F3019h (i.e., $F(x) = 516F3019h$), then: $F(x) = x^{30} + x^{28} + x^{24} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{13} + x^{12} + x^4 + x^3 + x$ (i.e., $F(x) = 8AF60C98h$)
$F_t(x)$	$F(x)$ with the bit positions of each byte transposed (i.e., bit 7 is bit 0, bit 6 is bit 1, etc.): $F_t(x) = b_{T(0)}x^{(k-1)} + b_{T(1)}x^{(k-2)} + \dots + b_{T(k-2)}x + b_{T(k-1)}$ For example, if the frame, except for the CRC field, contains one data dword set to 516F3019h (i.e., $F(x) = 516F3019h$), then: $F_t(x) = x^{31} + x^{27} + x^{25} + x^{23} + x^{22} + x^{21} + x^{20} + x^{18} + x^{17} + x^{11} + x^{10} + x^7 + x^4 + x^3$ (i.e., $F_t(x) = 8AF60C98h$)
$L(x)$	The identity polynomial of degree 31 (i.e., a polynomial with all of the coefficients set to one): $L(x) = x^{31} + x^{30} + \dots + x + 1$ (i.e., $L(x) = FFFFFFFFh$)
$G(x)$	The CRC generator polynomial (i.e., the divisor polynomial): $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (i.e., $G(x) = 1_04C11DB7h$)
$R(x)$	The remainder polynomial, which is of degree less than 32.
$R_t(x)$	$R(x)$ with the bit positions of each byte transposed.
$Q(x)$	A quotient polynomial resulting from CRC calculation by the transmitter. This value is discarded.
$Q'(x)$	A quotient polynomial resulting from CRC calculation by the receiver. This value is discarded.
$M(x)$	A polynomial representing the transmitted frame including the CRC field, which is of degree $k+31$.
$M'(x)$	A polynomial representing the received frame including the received CRC field. If the received frame has no errors, then $M'(x) = M(x)$ and $M'(x)$ is of degree $k+31$.
$M'_t(x)$	$M'(x)$ with the bit positions of each byte transposed.
$R'(x)$	The result of finding the remainder of an error-free reception of $M(x)$, and also the remainder of $x^{32}L(x) / G(x)$, which is a unique constant polynomial: $R'(x) = x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$ (i.e., $R'(x) = C704DD7Bh$)
$R'_t(x)$	$R'(x)$ with the bit positions of each byte transposed: $R'_t(x) = x^{28} + x^{27} + x^{26} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{14} + x^{10} + x^5 + 1$ (i.e., $R'_t(x) = 1CDF4421h$)

6.5.2 CRC generation

The CRC is calculated from $F(x)$ as follows:

$$x^k \times L(x) + x^{32} \times F_t(x) = Q(x) \times G(x) + R(x)$$

That is:

- 1) the frame $F(x)$, not including the CRC field, is transposed into $F_t(x)$;
- 2) the first 32 bits of the transposed frame are inverted (i.e., $x^k \times L(x)$ is added);
- 3) 32 bits of zero are appended to the end (i.e., $F_t(x)$ is multiplied by x^{32}); and
- 4) this result is divided by the generator polynomial $G(x)$ to find the remainder $R(x)$.

The transmitter shall present $M(x)$ to the 8b10b encoder:

$$M(x) = x^{32} \times F(x) + L(x) + R_t(x)$$

That is, the inverted transposed remainder is appended to the end of the frame, then this result (i.e., $M(x)$) is presented to the 8b10b encoder for transmission.

For the purposes of CRC computation, inverting the first 32 bits of a frame may be performed by one of the following methods:

- a) inverting the first 32 bits of the frame $F(x)$ and seeding the CRC remainder register with 00000000h;
- b) seeding the CRC remainder register with FFFFFFFFh; or
- c) prepending the constant 62F52692h to $F(x)$ and seeding the CRC remainder register with 00000000h.

NOTE 33 The bit order of $F(x)$ used to calculate the CRC is the same order as the bit transmission order (i.e., the bits within each byte encoded into a data dword are transposed to match the implicit transposition in the 8b10b encoding process).

Figure 95 shows the CRC process for an address frame, an SSP frame and SMP frame.

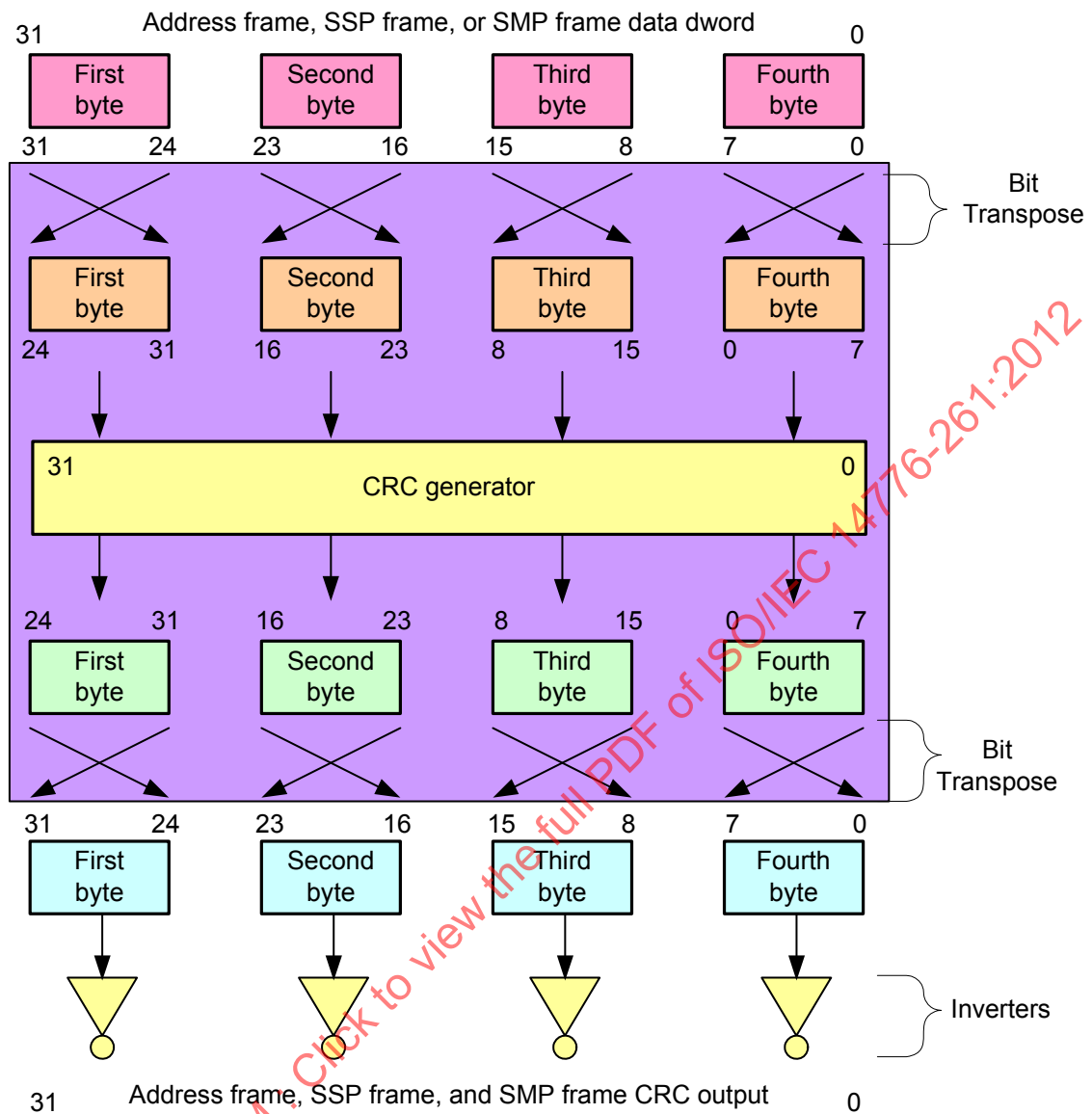


Figure 95 — Address frame, SSP frame, and SMP frame CRC bit ordering

Dwords in STP frames are little-endian and are fed into the STP CRC generator without swapping bits within each byte and without inverting the output like the SAS CRC generator. Figure 96 shows the STP CRC bit ordering.

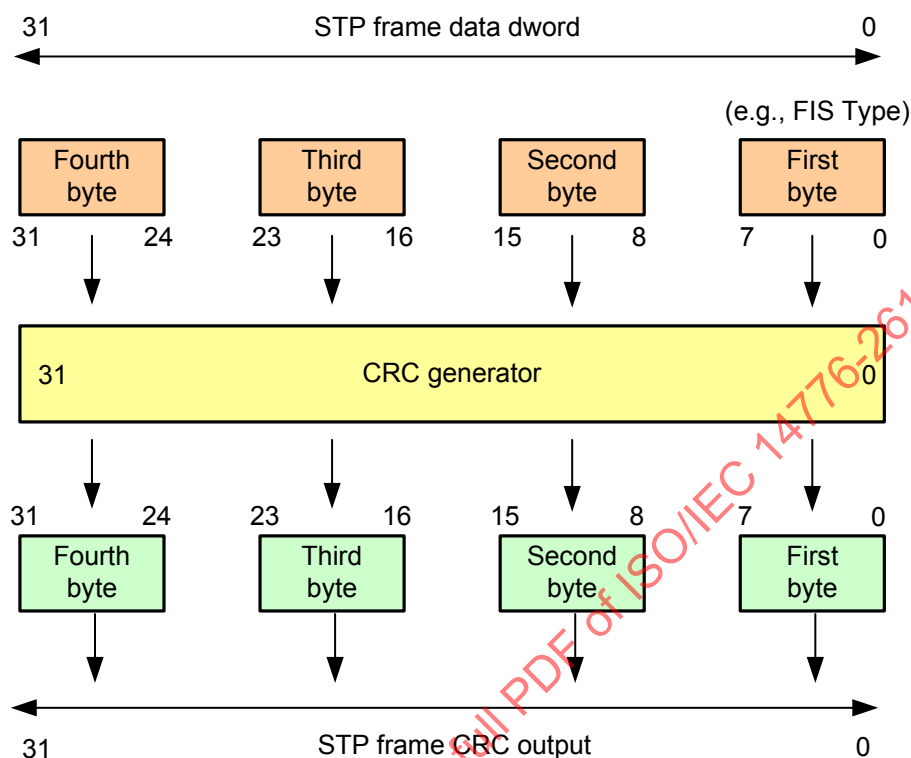


Figure 96 — STP frame CRC bit ordering

Since STP is little-endian, the first byte of a dword is in bits 7:0 rather than 31:24 as in SSP and SMP. As a result, the first byte contains the least-significant bit. In SSP and SMP, the first byte contains the most-significant bit.

See 6.7 for details on how the CRC generator fits into the dword flow along with the scrambler.

6.5.3 CRC checking

The CRC of received frame is calculated by the receiver in the same manner that it is generated by the transmitter.

That is:

- 1) the received frame $M'(x)$, including the CRC field, is transposed into $M_t'(x)$;
- 2) the first 32 bits of the received transposed frame are inverted;
- 3) 32 bits of zero are appended to the end; and
- 4) this result is divided by the generator polynomial $G(x)$ to find the remainder.

A received frame that has not incurred any CRC detectable errors during transmission generates a remainder equal to $R'(x)$.

If there were no transmission errors, then the received frame $M'(x)$ equals $M(x)$:

$$\begin{aligned} M'(x) &= M(x) \\ &= x^{32} \times F(x) + L(x) + R_t(x) \end{aligned}$$

The CRC $R'(x)$ is derived as follows:

$$\begin{aligned} x^{(k+32)} \times L(x) + x^{32} \times M_t'(x) &= x^{(k+32)} \times L(x) + x^{32} \times (x^{32} \times F_t(x) + L(x) + R(x)) \\ &= x^{32} \times Q(x) \times G(x) + x^{32} \times L(x) \end{aligned}$$

However, $G(x)$ divides $x^{32}L(x)$:

$$x^{32} \times L(x) = Q'(x) \times G(x) + R'(x)$$

Since $L(x)$ and $G(x)$ are known and constant, $R'(x)$ is known and constant and is expected by the receiver after calculating the CRC of a received frame.

From the previous two results:

$$x^{(k+32)} \times L(x) + x^{32} \times (x^{32} \times F_t(x) + L(x) + R(x)) = (x^{32} \times Q(x) + Q'(x)) \times G(x) + R'(x)$$

$R'(x)$ is then transposed and inverted, in the same manner as is done by the transmitter, to obtain:

$$R_t'(x) + L(x) = 1CDF4421h$$

As an alternative to this process, the receiver may check the CRC validity of the frame by stripping off the last 32 bits, leaving $F(x)$, and calculating the CRC as defined in 6.5.2. The frame has a valid CRC if the result $L(x) + R_t(x)$ equals the last 32 bits of the frame that were stripped.

See 6.7 for details on where the CRC checker fits into the dword flow along with the descrambler.

6.6 Scrambling

Scrambling is used to reduce the probability of long strings of repeated patterns appearing on the physical link.

All data dwords are scrambled. Table 82 lists the scrambling for different types of data dwords.

Table 82 — Scrambling for different data dword types

Connection state	Data dword type	Description of scrambling
Outside connections	SAS idle dword	When a connection is not open and there are no other dwords to transmit, vendor specific scrambled data dwords shall be transmitted.
	Address frame	After an SOAF, all data dwords shall be scrambled until the EOAF.
Inside SSP connection	SSP frame	After an SOF, all data dwords shall be scrambled until the EOF.
	SSP idle dword	When there are no other dwords to transmit, vendor specific scrambled data dwords shall be transmitted.
Inside SMP connection	SMP frame	After an SOF, all data dwords shall be scrambled until the EOF.
	SMP idle dword	When there are no other dwords to transmit, vendor specific scrambled data dwords shall be transmitted.
Inside STP connection	STP frame	After a SATA_SOF, all data dwords shall be scrambled until the SATA_EOF.
	Continued primitive	After a SATA_CONT, vendor specific scrambled data dwords shall be transmitted until a primitive other than a deletable primitive is transmitted.

Data dwords being transmitted shall be XORed with a defined pattern to produce a scrambled value encoded and transmitted on the physical link. Received data dwords shall be XORed with the same pattern after decoding to produce the original data dword value, provided there are no transmission errors.

The pattern that is XORed with the data dwords is defined by the output of a linear feedback shift register implemented with the following polynomial:

$$G(x) = x^{16} + x^{15} + x^{13} + x^4 + 1.$$

The output of the pattern generator is 16 bits wide. For each data dword, two outputs of the pattern generator are used as follows:

- 1) the first output of the generator is applied to the lower 16 bits (i.e., bits 15 through 0) of the 32-bit data dword being transmitted or received; and
- 2) the second output of the generator is applied to the upper 16 bits (i.e., bits 31 through 16).

NOTE 34 Scrambling is not based on data feedback, so the sequence of values XORed with the data being transmitted is constant.

The value of the linear feedback shift register shall be initialized at each SOF and SOAF to FFFFh.

For detailed requirements about scrambling of data dwords following SATA_SOF and SATA_CONT, see SATA.

NOTE 35 STP scrambling uses two linear feedback shift registers, since continued primitive sequences may occur inside STP frames and the STP frame and the continued primitive sequence have independent scrambling patterns.

Annex E contains information on scrambling implementations.

6.7 Bit order of CRC and scrambler

Figure 97 shows how data dwords and primitives are routed to the bit transmission logic in figure 57 (see 5.5). Data dwords go through the CRC generator and scrambler.

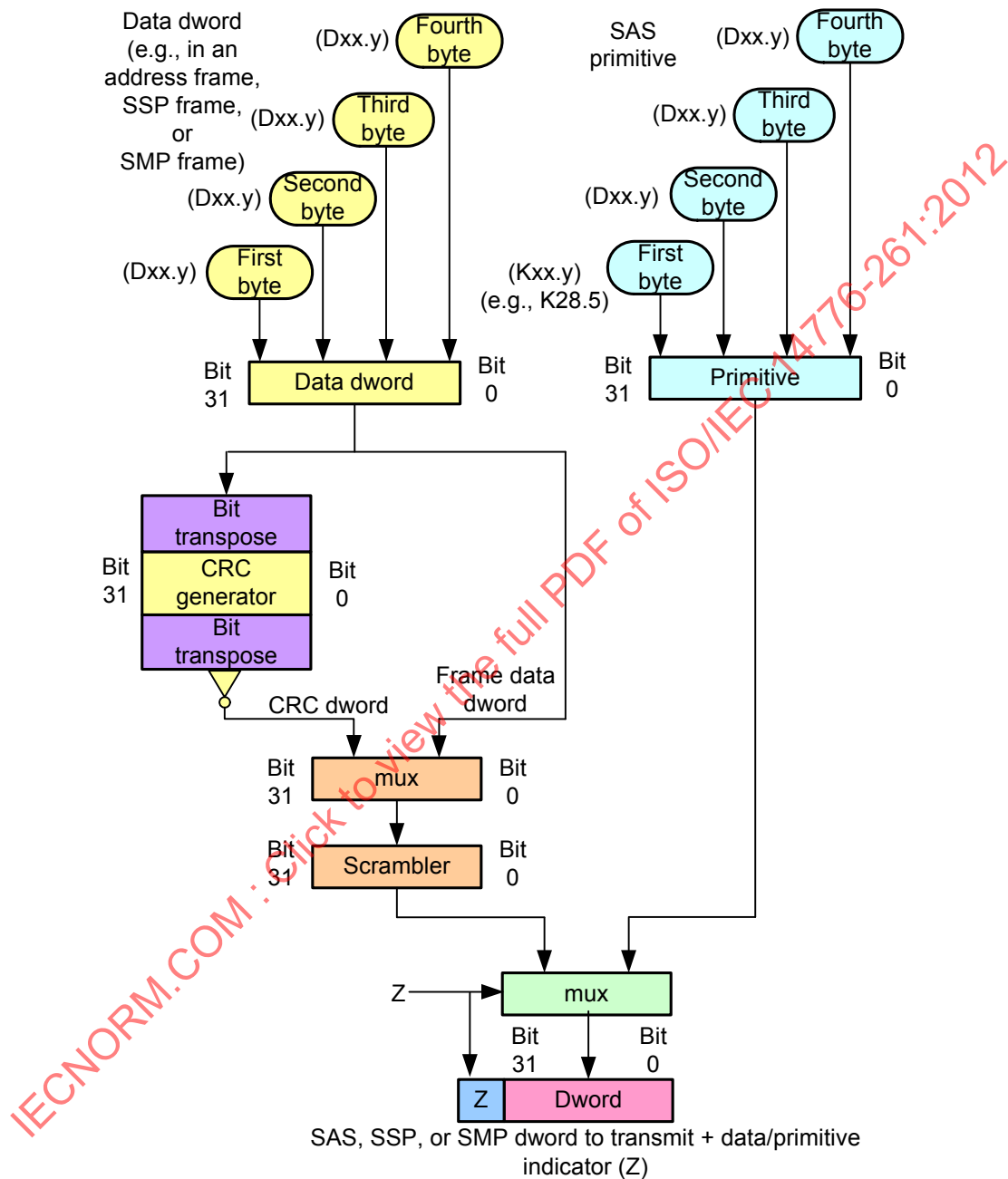


Figure 97 — Transmit path bit ordering

Figure 98 shows the routing of dwords received from the bit reception logic in figure 58 (see 5.5).

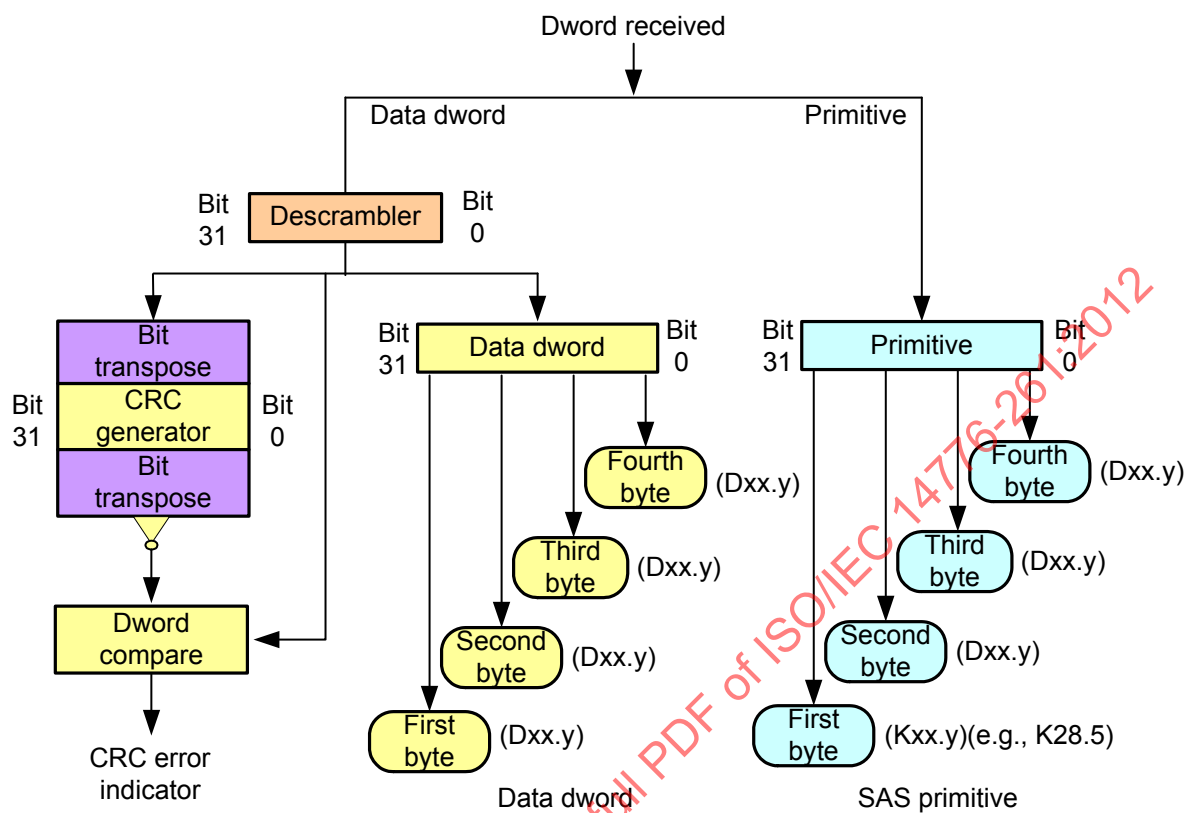


Figure 98 — Receive path bit ordering

Figure 99 shows the STP transmit path bit ordering.

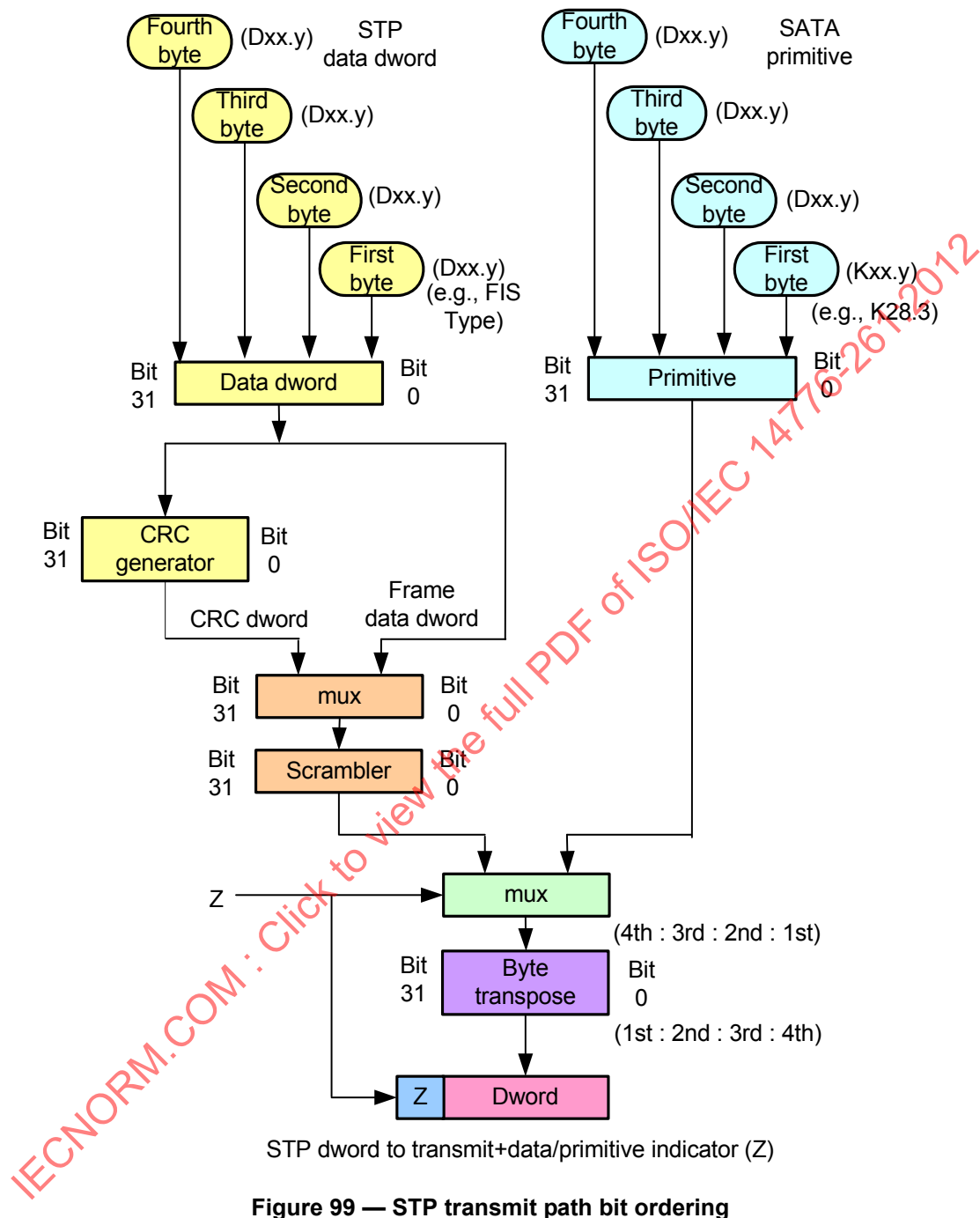


Figure 99 — STP transmit path bit ordering

Figure 100 shows the STP receive path bit ordering.

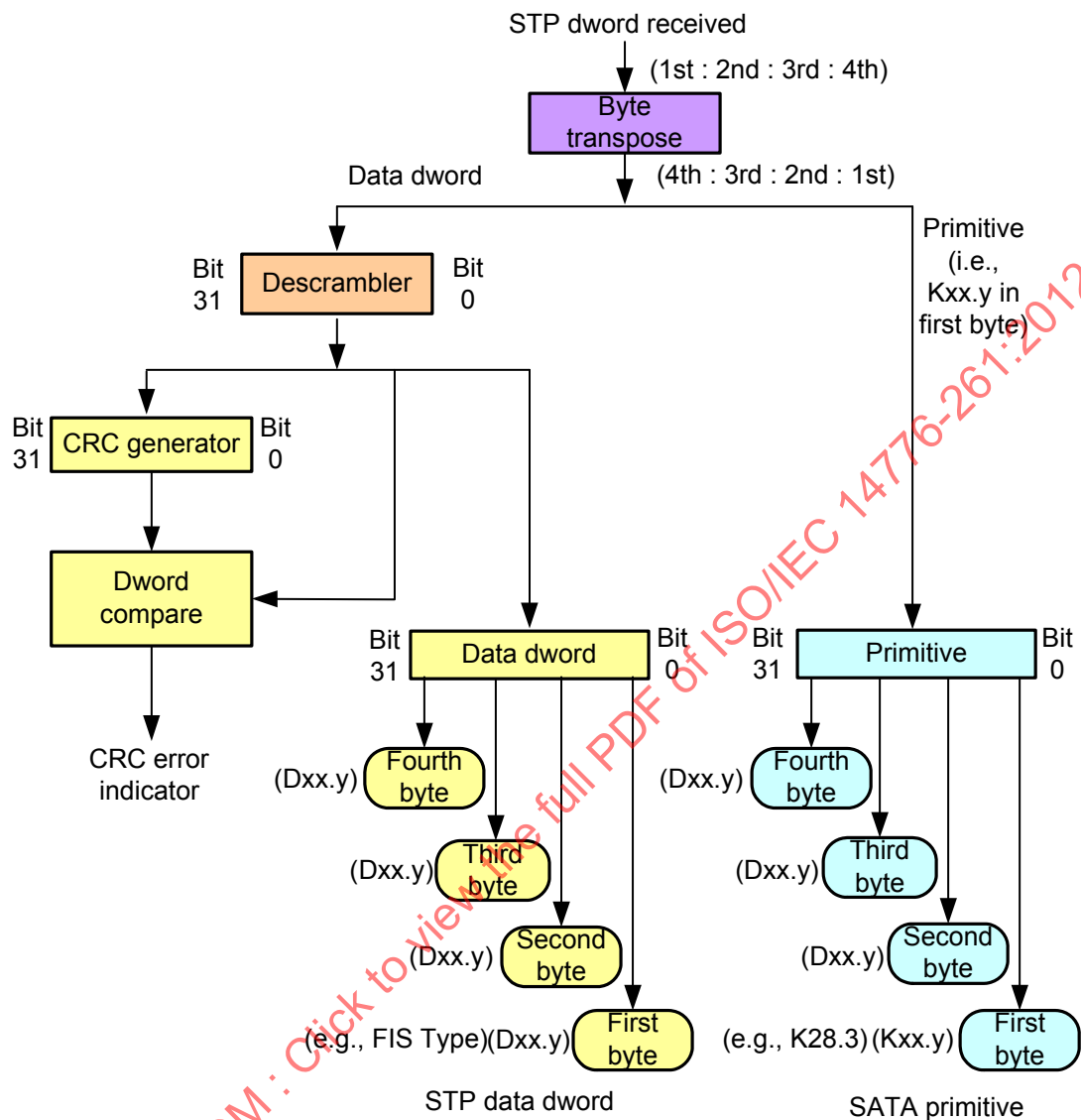


Figure 100 — STP receive path bit ordering

6.8 Address frames

6.8.1 Address frames overview

Address frames are used for the identification sequence (see 6.9) and for connection requests (see 6.13). Address frames are preceded by SOAF and followed by EOAF as shows in figure 101.

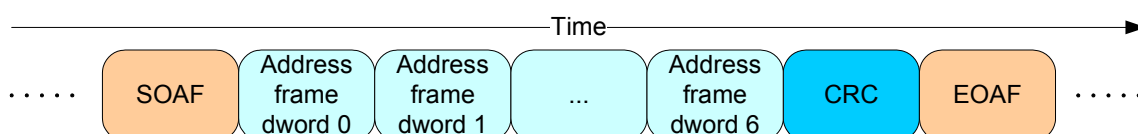


Figure 101 — Address frame transmission

Address frames shall only be transmitted outside connections. Partial address frames (i.e., not containing the number of data dwords defined for the frame) shall not be transmitted. All data dwords in an address frame shall be scrambled.

Table 83 defines the address frame format.

Table 83 — Address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0					ADDRESS FRAME TYPE			
1	Frame type dependent bytes							
27								
28								
31	CRC						(LSB)	

The ADDRESS FRAME TYPE field indicates the type of address frame and is defined in table 84. This field determines the definition of the frame type dependent bytes.

Table 84 — ADDRESS FRAME TYPE field

Code	Address frame type	Description
0h	IDENTIFY	Identification sequence
1h	OPEN	Connection request
All others	Reserved	

The CRC field contains a CRC value (see 6.5) that is computed over the entire address frame prior to the CRC field.

Address frames with unknown address frame types, incorrect lengths, or CRC errors shall be ignored by the recipient.

6.8.2 IDENTIFY address frame

Table 85 defines the IDENTIFY address frame format used for the identification sequence. The IDENTIFY address frame is transmitted by each logical phy after the phy reset sequence completes if the physical link is a SAS physical link. The IDENTIFY address frame transmitted by each logical phy in a physical phy shall be identical.

Table 85 — IDENTIFY address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved	DEVICE TYPE			ADDRESS FRAME TYPE (0h)			
1	Reserved				REASON			
2	Reserved				SSP INITIATOR PORT	STP INITIATOR PORT	SMP INITIATOR PORT	Restricted (for OPEN address frame)
3	Reserved				SSP TARGET PORT	STP TARGET PORT	SMP TARGET PORT	Restricted (for OPEN address frame)
4	DEVICE NAME							
11								
12								
19	SAS ADDRESS							
20								
21	Reserved	SLUMBER CAPABLE	PARTIAL CAPABLE	INSIDE ZPSDS PERSISTENT	REQUESTED INSIDE ZPSDS	BREAK_REPLY CAPABLE		
22	Reserved							
27								
28	(MSB)	CRC						
31	(LSB)							

The DEVICE TYPE field indicates the type of device containing the phy, and is defined in table 86.

Table 86 — DEVICE TYPE field

Code	Description
001b	End device
010b	Expander device
011b	Expander device compliant with SAS-1.1
All others	Reserved

The ADDRESS FRAME TYPE field shall be set to the value defined in table 85.

The REASON field indicates the reason for the link reset sequence and is defined in table 87.

Table 87 — REASON field

Code	Description
0h	Unknown reason
1h	Power on
2h	Hard reset (e.g., the port containing this phy received a HARD_RESET primitive sequence during the hard reset sequence) (see 4.4.2), or SMP PHY CONTROL function HARD RESET phy operation (see 9.4.3.28)
3h	SMP PHY CONTROL function LINK RESET phy operation, or TRANSMIT SATA PORT SELECTION SIGNAL phy operation (see 9.4.3.28)
4h	Loss of dword synchronization (see 5.10)
5h	After the multiplexing sequence completes, MUX (LOGICAL LINK 0) received in logical link 1 or MUX (LOGICAL LINK 1) received in logical link 0 (see 5.11).
6h	I_T nexus loss timer expired in the STP target port of an STP/SATA bridge when the phy was attached to a SATA device (see 4.5)
7h	Break Timeout Timer expired (see 6.13.8)
8h	Phy test function stopped (see 9.4.3.29)
9h	Expander device reduced functionality (see 4.6.8)
Ah to Fh	Reserved

An SSP INITIATOR PORT bit set to one indicates that an SSP initiator port is present. An SSP INITIATOR PORT bit set to zero indicates that an SSP initiator port is not present. Expander devices shall set the SSP INITIATOR PORT bit to zero.

An STP INITIATOR PORT bit set to one indicates that an STP initiator port is present. An STP INITIATOR PORT bit set to zero indicates that an STP initiator port is not present. Expander devices shall set the STP INITIATOR PORT bit to zero.

An SMP INITIATOR PORT bit set to one indicates that an SMP initiator port is present. An SMP INITIATOR PORT bit set to zero indicates that an SMP initiator port is not present. Expander devices may set the SMP INITIATOR PORT bit to one.

An SSP TARGET PORT bit set to one indicates that an SSP target port is present. An SSP TARGET PORT bit set to zero indicates that an SSP target port is not present. Expander devices shall set the SSP TARGET PORT bit to zero.

An STP TARGET PORT bit set to one indicates that an STP target port is present. An STP TARGET PORT bit set to zero indicates that an STP target port is not present. Expander devices shall set the STP TARGET PORT bit to zero.

An SMP TARGET PORT bit set to one indicates that an SMP target port is present. An SMP TARGET PORT bit set to zero indicates that an SMP target port is not present. Expander devices shall set the SMP TARGET PORT bit to one.

The DEVICE NAME field indicates the device name (see 4.2.6) of the SAS device or expander device transmitting the IDENTIFY address frame. A DEVICE NAME field set to 00000000 00000000h indicates the device name is not provided in this field.

NOTE 36 In expander devices, the DEVICE NAME field, if not set to 00000000 00000000h, contains the same value as the SAS ADDRESS field.

For SAS ports, the SAS ADDRESS field indicates the port identifier (see 4.2.9) of the SAS port transmitting the IDENTIFY address frame. For expander ports, the SAS ADDRESS field indicates the device name (see 4.2.6) of the expander device transmitting the IDENTIFY address frame.

The PHY IDENTIFIER field indicates the phy identifier (see 4.2.10) of the phy transmitting the IDENTIFY address frame.

The REQUESTED INSIDE ZPSDS bit indicates the value of the REQUESTED INSIDE ZPSDS bit in the zone phy information (see 4.9.3.1) at the time the IDENTIFY address frame is transmitted. If the phy transmitting the IDENTIFY address frame is contained in an end device, a non-zoning expander device, or a zoning expander device with zoning disabled, then the REQUESTED INSIDE ZPSDS bit shall be set to zero.

The INSIDE ZPSDS PERSISTENT bit indicates the value of the INSIDE ZPSDS PERSISTENT bit in the zone phy information (see 4.9.3.1) at the time the IDENTIFY address frame is transmitted. If the phy transmitting the IDENTIFY address frame is contained in an end device, a non-zoning expander device, or a zoning expander device with zoning disabled, then the INSIDE ZPSDS PERSISTENT bit shall be set to zero.

The BREAK_REPLY CAPABLE field indicates that the phy is capable of responding to received BREAK primitive sequences with a BREAK_REPLY primitive sequence (see 6.13.5).

See 4.1.4 for additional requirements concerning the DEVICE TYPE field, the BREAK_REPLY CAPABLE bit, the SSP INITIATOR PORT bit, the STP INITIATOR PORT bit, the SMP INITIATOR PORT bit, the SSP TARGET PORT bit, the STP TARGET PORT bit, the SMP TARGET PORT bit, and the SAS ADDRESS field.

A PARTIAL CAPABLE bit set to one indicates that the phy is capable of supporting the partial phy power condition (see 4.10.1.3). A PARTIAL CAPABLE bit set to zero indicates that the phy is not capable of supporting the partial phy power condition.

A SLUMBER CAPABLE bit set to one indicates that the phy is capable of supporting the slumber phy power condition (see 4.10.1.4). A SLUMBER CAPABLE bit set to zero indicates that the phy is not capable of supporting the slumber phy power condition.

If multiplexing is enabled (see 5.11) or optical mode is enabled, then the PARTIAL CAPABLE bit and the SLUMBER CAPABLE bit shall be set to zero.

The CRC field is defined in 6.8.1.

6.8.3 OPEN address frame

Table 88 defines the OPEN address frame format used for connection requests.

Table 88 — OPEN address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	INITIATOR PORT	PROTOCOL			ADDRESS FRAME TYPE (1h)			
1	FEATURES				CONNECTION RATE			
2	(MSB)	INITIATOR CONNECTION TAG						(LSB)
3								
4		DESTINATION SAS ADDRESS						
11								
12								
19		SOURCE SAS ADDRESS						
20		SOURCE ZONE GROUP						
21		PATHWAY BLOCKED COUNT						
22	(MSB)	ARBITRATION WAIT TIME						(LSB)
23								
24		MORE COMPATIBLE FEATURES (00000000h)						
27								
28	(MSB)							
31		CRC						(LSB)

An INITIATOR PORT bit set to one specifies that the source port is acting as a SAS initiator port. An INITIATOR PORT bit set to zero specifies that the source port is acting as a SAS target port. If a SAS port sets the INITIATOR PORT bit to one, it shall operate only in its initiator role during the connection. If a SAS port sets the INITIATOR PORT bit to zero, it shall operate only in its target role during the connection.

If a SAS port accepts an OPEN address frame with the INITIATOR PORT bit set to one, it shall operate only in its target role during the connection. If a SAS port accepts an OPEN address frame with the INITIATOR PORT bit set to zero, it shall operate only in its initiator role during the connection.

The PROTOCOL field specifies the protocol for the connection being requested and is defined in table 89.

Table 89 — PROTOCOL field

Code	Description
000b	SMP
001b	SSP
010b	STP
All others	Reserved

The ADDRESS FRAME TYPE field shall be set to the value defined in table 88.

The FEATURES field specifies any additional features that are incompatible with SAS-1.1 and is defined in table 90.

Table 90 — FEATURES field

Code	Description
0h	No additional features
All others	Reserved

The CONNECTION RATE field specifies the connection rate (see 4.1.12) being requested between the source and destination, and is defined in table 91.

Table 91 — CONNECTION RATE field

Code	Description
8h	1.5 Gbps
9h	3 Gbps
Ah	6 Gbps
Bh to Fh	Reserved for future connection rates
All others	Reserved

A SAS initiator port shall set the initial CONNECTION RATE field to:

- the highest supported connection rate supported by a potential pathway as determined during the discover process (e.g., based on the logical link rates of each logical link reported in the SMP DISCOVER responses); or
- the logical link rate of the logical phy used to transmit the OPEN address frame.

If a SAS initiator port selected a connection rate based on discover process information, but the connection request results in OPEN_REJECT (CONNECTION RATE NOT SUPPORTED), then the discover process information is no longer current and the discover process should be run again.

A SAS target port shall set the initial CONNECTION RATE field to:

- the last known good connection rate established with the SAS initiator port; or
- for the first frame that it intends to transmit in the connection, the connection rate that was used by the SAS initiator port to deliver the command or task management function for that frame.

Each time that a connection request with a connection rate greater than 1.5 Gbps results in OPEN_REJECT (CONNECTION RATE NOT SUPPORTED), the SAS port shall reattempt the connection request with a lower connection rate (e.g., drop from 6 Gbps to 3 Gbps or 1.5 Gbps) and send the same frames in the resulting connection that the SAS port intended to send at the initial connection rate.

The INITIATOR CONNECTION TAG field is used for SSP and STP connection requests to provide a SAS initiator port an alternative to using the SAS target port's SAS address for context lookup when the SAS target port originates a connection request. An SSP initiator port or STP initiator port shall set the INITIATOR CONNECTION TAG field to FFFFh if the SSP initiator port or STP initiator port does not require that this field be provided by the SAS target port. If an SSP initiator port or STP initiator port does require the field to be provided, then the SSP initiator port or STP initiator port should set the INITIATOR CONNECTION TAG field to a unique value per SAS target port. When requesting a connection to a SAS initiator port, a SAS target port shall set the INITIATOR CONNECTION TAG field to the most recent value received or the value received in one of the connection requests for one of the outstanding commands or task management functions from the SAS initiator port. A SAS initiator port shall:

- a) use the same INITIATOR CONNECTION TAG field value for all connection requests to the same SAS target port, and
- b) only change the INITIATOR CONNECTION TAG field value when it has no commands or task management functions outstanding to that SAS target port.

SAS target ports are not required to check consistency of the INITIATOR CONNECTION TAG field in different connection requests from the same SAS initiator port. SMP initiator ports shall set the INITIATOR CONNECTION TAG field to FFFFh for SMP connection requests.

The DESTINATION SAS ADDRESS field specifies the port identifier (see 4.2.9) of the SAS port to which a connection is being requested.

The SOURCE SAS ADDRESS field specifies the port identifier (see 4.2.9) of the SAS port that originated the OPEN address frame.

The SOURCE ZONE GROUP field identifies the zone group of the phy making the connection request. The SOURCE ZONE GROUP field shall be:

- a) set to 00h when transmitted by an end device;
- b) set to 00h when transmitted by an expander device on a phy with the INSIDE ZPSDS bit set to zero;
- c) set to the source zone group for the outgoing connection request as described in table 33 (see 4.9.3.5) when transmitted by an expander device on a phy with the INSIDE ZPSDS bit set to one;
- d) ignored when received by an end device;
- e) ignored when received by an expander device on a phy with the INSIDE ZPSDS bit set to zero; or
- f) used to determine the source zone group for the incoming connection request as described in table 33 (see 4.9.3.5) when received by an expander device on a phy with the INSIDE ZPSDS bit set to one.

The PATHWAY BLOCKED COUNT field specifies the number of times the port has retried this connection request due to receiving OPEN_REJECT (PATHWAY BLOCKED), OPEN_REJECT (RESERVED STOP 0), or OPEN_REJECT (RESERVED STOP 1). The port shall not increment the PATHWAY BLOCKED COUNT value past FFh. If the port changes connection requests, then the port shall set the PATHWAY BLOCKED COUNT field to 00h.

The ARBITRATION WAIT TIME field specifies how long the port transmitting the OPEN address frame has been waiting for a connection request to be accepted or rejected. This time is maintained by the port layer in an Arbitration Wait Time timer (see 7.2.2). For values from 0000h to 7FFFh, the Arbitration Wait Time timer increments in one microsecond steps. For values from 8000h to FFFFh, the Arbitration Wait Time timer increments in one millisecond steps. The maximum value represents 32 767 ms + 32 768 µs. Table 92 describes several values of the ARBITRATION WAIT TIME field. See 6.13.3 for details on arbitration fairness.

Table 92 — ARBITRATION WAIT TIME field

Code	Description
0000h	0 μ s
0001h	1 μ s
...	...
7FFFh	32 767 μ s
8000h	0 ms + 32 768 μ s
8001h	1 ms + 32 768 μ s
...	...
FFFFh	32 767 ms + 32 768 μ s

The MORE COMPATIBLE FEATURES field shall be set to the value defined in table 88. A phy receiving an OPEN address frame shall ignore the MORE COMPATIBLE FEATURES field.

The CRC field is defined in 6.8.1.

6.9 Link reset sequence

6.9.1 Link reset sequence overview

For SATA, a link reset sequence is a phy reset sequence (see 5.7).

For SAS, a link reset sequence is either:

- a) the following sequence:
 - 1) a phy reset sequence indicating that the physical link is using SAS rather than SATA; and
 - 2) an identification sequence,
 or
- b) the following sequence:
 - 1) a phy reset sequence indicating that the physical link is using SAS rather than SATA;
 - 2) a hard reset sequence;
 - 3) another phy reset sequence indicating that the physical link is using SAS rather than SATA; and
 - 4) an identification sequence.

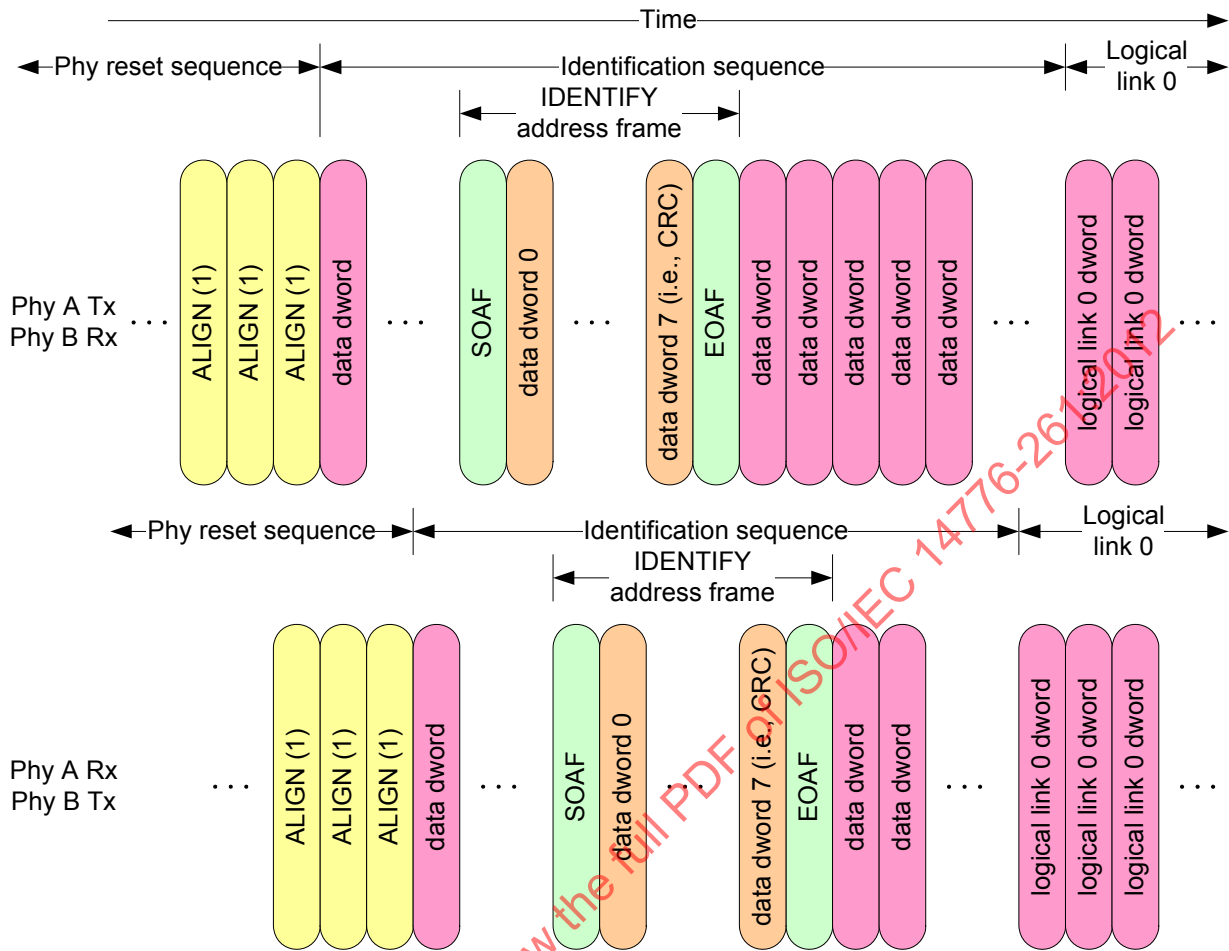
An identification sequence occurs when a logical phy:

- a) transmits one or three IDENTIFY address frames (see 6.8.2); and
- b) does not receive a HARD_RESET primitive sequence.

A hard reset sequence occurs when, after the phy reset sequence, a logical phy:

- a) transmits a HARD_RESET primitive sequence (see 6.2.6.8); or
- b) receives a HARD_RESET primitive sequence.

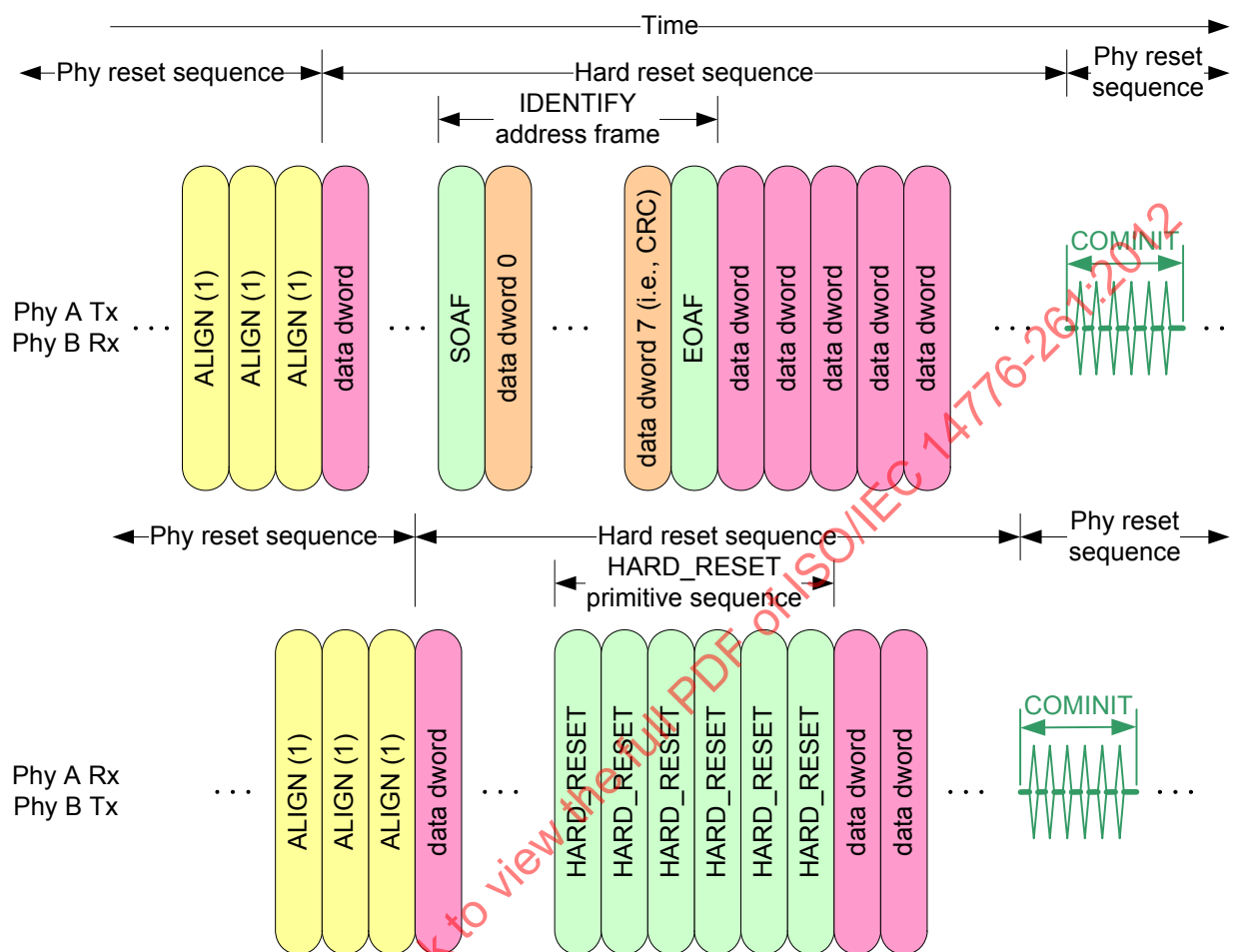
Figure 102 shows two phys with multiplexing disabled performing the identification sequence. Only one IDENTIFY address frame is shown in this example.



Note: Phys transmit deletable primitives for physical link rate tolerance management after the phy reset sequence.

Figure 102 — Identification sequence

Figure 103 shows phy A attempting to perform the identification sequence and phy B performing the hard reset sequence. Because phy A receives a HARD_RESET primitive sequence, a hard reset sequence occurs. Multiplexing is disabled and only one IDENTIFY address frame is shown in this example.



Note: Phys transmit deletable primitives for physical link rate tolerance management after the phy reset sequence.

Figure 103 — Hard reset sequence

Each logical phy receives an IDENTIFY address frame or a HARD_RESET primitive sequence from the logical phy to which it is attached.

If a logical phy receives a valid IDENTIFY address frame (see 6.10.4.3.1) within 1 ms of phy reset sequence completion, then the SAS address in the outgoing IDENTIFY address frame(s) and the SAS address in the incoming IDENTIFY address frame determine the port to which the logical phy belongs (see 4.1.4). The logical phy ignores subsequent IDENTIFY address frames and HARD_RESETs until another phy reset sequence occurs.

If a logical phy receives a HARD_RESET primitive sequence within 1 ms of phy reset sequence completion, then the logical phy shall consider this to be a reset event, and the port containing the logical phy shall process a hard reset (see 4.4.2).

If a logical phy does not receive a HARD_RESET primitive sequence or a valid IDENTIFY address frame within 1 ms of phy reset sequence completion, then the physical phy containing the logical phy shall restart the phy reset sequence.

6.9.2 Expander device handling of link reset sequences

After completing the link reset sequence on a phy and completing internal initialization, the ECM within an expander device shall be capable of routing connection requests through that phy. The expander device may return OPEN_REJECT (NO DESTINATION) until it is ready to process connection requests.

The ECM of an externally configurable expander device is dependent on the completion of the discover process (see 4.7) for routing connection requests using the table routing method.

6.10 SL_IR (link layer identification and hard reset) state machines

6.10.1 SL_IR state machines overview

The SL_IR (link layer identification and hard reset) state machines control the flow of dwords on the physical link that are associated with the identification and hard reset sequences. The state machines are as follows:

- a) SL_IR_TIR (transmit IDENTIFY or HARD_RESET primitive sequence) state machine (see 6.10.3);
- b) SL_IR_RIF (receive IDENTIFY address frame) state machine (see 6.10.4); and
- c) SL_IR_IRC (identification and hard reset control) state machine (see 6.10.5).

The SL_IR state machines send the following messages to the SL state machines (see 6.15) in SAS devices or the XL (see 6.16) state machine in expander devices:

- a) Enable Disable SAS Link (Enable); and
- b) Enable Disable SAS Link (Disable).

This state machine shall maintain the timers listed in table 93.

Table 93 — SL_IR_IRC state machine timers

Timer	Initial value
Receive Identify Timeout timer	1 ms

Figure 104 shows the SL_IR state machines.

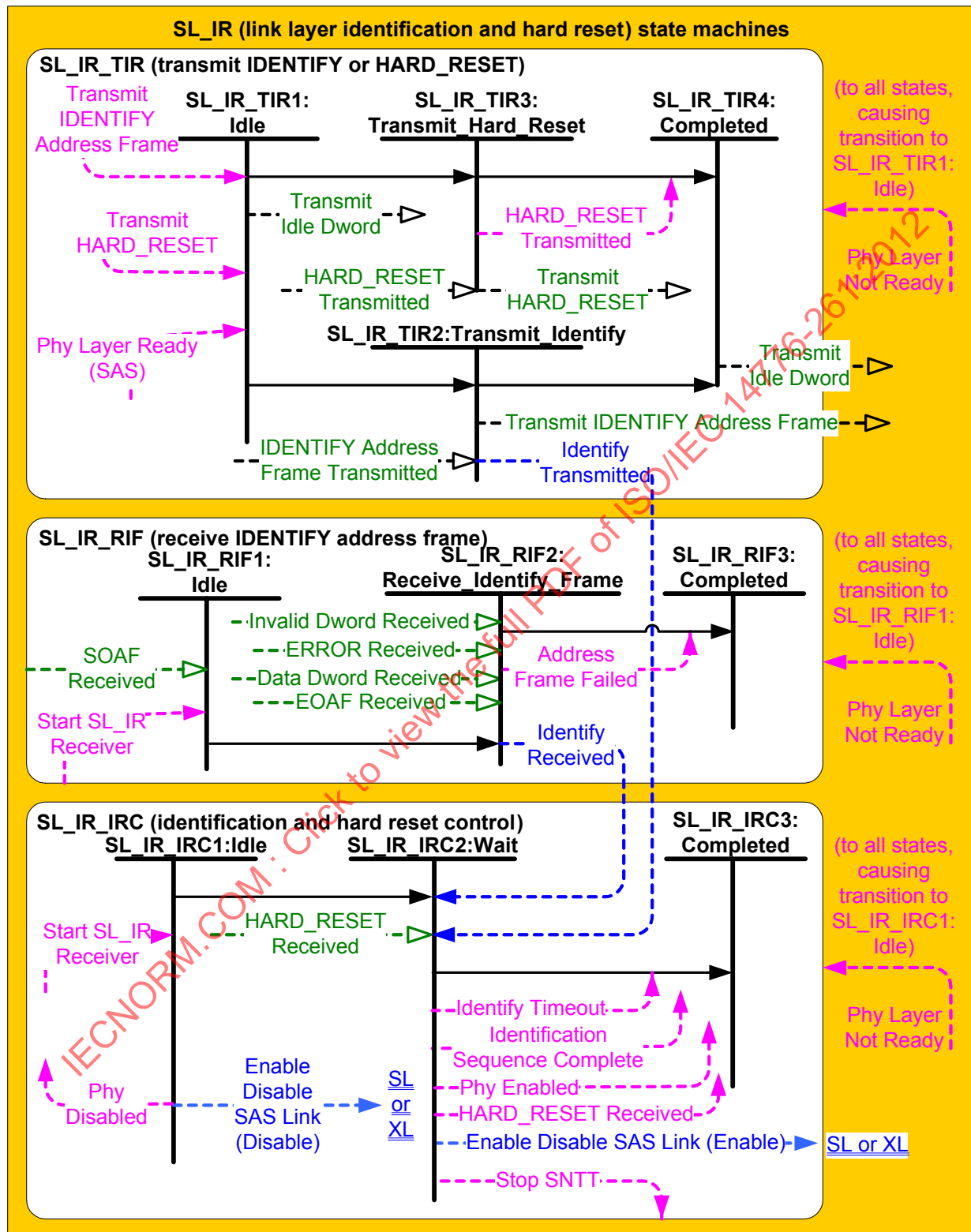


Figure 104 — SL_IR (link layer identification and hard reset) state machines

6.10.2 SL_IR transmitter and receiver

The SL_IR transmitter receives the following messages from the SL_IR state machines indicating primitive sequences, frames, and dwords to transmit:

- a) Transmit IDENTIFY Address Frame;
- b) Transmit HARD_RESET; and
- c) Transmit Idle Dword.

Upon receiving a Transmit IDENTIFY Address Frame message, the SL_IR transmitter shall transmit:

- 1) SOAF;
- 2) data dwords;
- 3) EOAF; and
- 4) at least 3 idle dwords.

NOTE 37 Phys compliant with SAS-1.1 were not required to transmit idle dwords after EOAF.

The SL_IR transmitter sends the following messages to the SL_IR state machines:

- a) HARD_RESET Transmitted; and
- b) IDENTIFY Address Frame Transmitted.

The SL_IR receiver sends the following messages to the SL_IR state machines indicating primitive sequences and dwords received from the SP_DWS receiver (see 5.10.2):

- a) SOAF Received;
- b) Data Dword Received;
- c) EOAF Received;
- d) ERROR Received;
- e) Invalid Dword Received; and
- f) HARD_RESET Received.

The SL_IR receiver shall not require reception of any idle dwords after an IDENTIFY address frame.

The SL_IR receiver shall ignore all other dwords.

The SL_IR transmitter relationship to other transmitters is defined in 4.3.2. The SL_IR receiver relationship to other receivers is defined in 4.3.3.

6.10.3 SL_IR_TIR (transmit IDENTIFY or HARD_RESET) state machine

6.10.3.1 SL_IR_TIR state machine overview

The SL_IR_TIR state machine's function is to transmit one or three IDENTIFY address frames or a HARD_RESET primitive sequence after the phy layer enables the link layer. This state machine consists of the following states:

- a) SL_IR_TIR1:Idle (see 6.10.3.2) (initial state);
- b) SL_IR_TIR2:Transmit_Identify (see 6.10.3.3);
- c) SL_IR_TIR3:Transmit_Hard_Reset (see 6.10.3.4); and
- d) SL_IR_TIR4:Completed (see 6.10.3.5).

This state machine receives the following requests from the management application layer:

- a) Transmit IDENTIFY Address Frame; and
- b) Transmit HARD_RESET.

This state machine shall start in the SL_IR_TIR1:Idle state. This state machine shall transition to the SL_IR_TIR1:Idle state from any other state after receiving a Phy Layer Not Ready confirmation.

6.10.3.2 SL_IR_TIR1:Idle state

6.10.3.2.1 State description

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IR transmitter.

6.10.3.2.2 Transition SL_IR_TIR1:Idle to SL_IR_TIR2:Transmit_Identify

This transition shall occur after both:

- a) a Phy Layer Ready (SAS) confirmation is received; and
- b) a Transmit IDENTIFY Address Frame request is received.

6.10.3.2.3 Transition SL_IR_TIR1:Idle to SL_IR_TIR3:Transmit_Hard_Reset

This transition shall occur after both:

- a) a Phy Layer Ready (SAS) confirmation is received; and
- b) a Transmit HARD_RESET request is received.

6.10.3.3 SL_IR_TIR2:Transmit_Identify state

6.10.3.3.1 State description

Upon entry into this state, this state shall send either one or three Transmit IDENTIFY Address Frame messages to the SL_IR transmitter.

NOTE 38 Phys compliant with SAS-1.1 only transmitted one Transmit IDENTIFY Address Frame message.

After this state receives an IDENTIFY Address Frame Transmitted message in response to its first Transmit IDENTIFY Address Frame message, this state shall send an Identify Transmitted message to the SL_IR_IRC state machine.

6.10.3.3.2 Transition SL_IR_TIR2:Transmit_Identify to SL_IR_TIR4:Completed

If this state sends one Transmit IDENTIFY Address Frame message, then this transition shall occur after sending an Identify Transmitted message to the SL_IR_IRC state machine.

If this state sends three Transmit IDENTIFY Address Frame messages, then this transition shall occur after receiving three Identify Transmitted messages.

6.10.3.4 SL_IR_TIR3:Transmit_Hard_Reset state

6.10.3.4.1 State description

Upon entry into this state, this state shall send a Transmit HARD_RESET message to the SL_IR transmitter.

After this state receives a HARD_RESET Transmitted message, this state shall send a HARD_RESET Transmitted confirmation to the management application layer.

6.10.3.4.2 Transition SL_IR_TIR3:Transmit_Hard_Reset to SL_IR_TIR4:Completed

This transition shall occur after sending a HARD_RESET Transmitted confirmation to the management application layer.

6.10.3.5 SL_IR_TIR4:Completed state

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IR transmitter.

6.10.4 SL_IR_RIF (receive IDENTIFY address frame) state machine

6.10.4.1 SL_IR_RIF state machine overview

The SL_IR_RIF state machine receives an IDENTIFY address frame and checks the IDENTIFY address frame to determine if the frame should be accepted or discarded by the link layer.

This state machine consists of the following states:

- a) SL_IR_RIF1:Idle (see 6.10.4.2) (initial state);
- b) SL_IR_RIF2:Receive_Identify_Frame (see 6.10.4.3); and
- c) SL_IR_RIF3:Completed (see 6.10.4.4).

This state machine shall start in the SL_IR_RIF1:Idle state. This state machine shall transition to the SL_IR_RIF1:Idle state from any other state after receiving a Phy Layer Not Ready confirmation.

6.10.4.2 SL_IR_RIF1:Idle state

6.10.4.2.1 State description

This state waits for an SOAF to be received from the physical link, indicating an address frame is arriving.

6.10.4.2.2 Transition SL_IR_RIF1:Idle to SL_IR_RIF2:Receive_Identify_Frame

This transition shall occur after both:

- a) a Start SL_IR Receiver confirmation is received; and
- b) an SOAF Received message is received.

6.10.4.3 SL_IR_RIF2:Receive_Identify_Frame state

6.10.4.3.1 State description

This state receives the dwords of an address frame and the EOAF.

If this state receives an SOAF Received message, then this state shall discard the address frame in progress, send an Address Frame Failed confirmation to the management application layer to indicate that an invalid address frame was received, and start receiving the new address frame.

If this state receives more than eight Data Dword Received messages (i.e., 32 bytes) after an SOAF Received message and before an EOAF Received message, then this state shall discard the address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid address frame was received.

If this state receives an Invalid Dword Received message or an ERROR Received message after an SOAF Received message and before an EOAF Received message, then this state shall either:

- a) ignore the invalid dword or ERROR; or
- b) discard the address frame in progress and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid address frame was received.

After receiving an EOAF Received message, this state shall check if the received frame is a valid IDENTIFY address frame.

This state shall accept an IDENTIFY address frame and send an Identify Received message to the SL_IR_IRC state machine if:

- a) the ADDRESS FRAME TYPE field is set to 0h (i.e., IDENTIFY);
- b) the number of bytes between the SOAF and EOAF is 32; and
- c) the CRC field contains a good CRC.

Otherwise, this state shall discard the address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid address frame was received.

6.10.4.3.2 Transition SL_IR_RIF2:Receive_Identify_Frame to SL_IR_RIF3:Completed

This transition shall occur after sending an Identify Received message.

6.10.4.4 SL_IR_RIF3:Completed state

This state waits for a Phy Layer Not Ready confirmation.

6.10.5 SL_IR_IRC (identification and hard reset control) state machine

6.10.5.1 SL_IR_IRC state machine overview

The SL_IR_IRC state machine ensures that IDENTIFY address frames have been both received and transmitted before enabling the rest of the link layer, and notifies the link layer if a HARD_RESET primitive sequence is received before an IDENTIFY address frame has been received.

This state machine consists of the following states:

- a) SL_IR_IRC1:Idle (see 6.10.5.2) (initial state);
- b) SL_IR_IRC2:Wait (see 6.10.5.3); and
- c) SL_IR_IRC3:Completed (see 6.10.5.4).

This state machine shall start in the SL_IR_IRC1:Idle state. This state machine shall transition to the SL_IR_IRC1:Idle state from any other state after receiving a Phy Layer Not Ready confirmation.

6.10.5.2 SL_IR_IRC1:Idle state

6.10.5.2.1 State description

This state waits for the link layer to be enabled. Upon entry into this state, this state shall:

- a) send an Enable Disable SAS Link (Disable) message to the SL state machines (see 6.15) or XL state machine (see 6.16) halting any link layer activity; and
- b) send a Phy Disabled confirmation to the port layer and the management application layer indicating that the phy is not ready for use.

6.10.5.2.2 Transition SL_IR_IRC1:Idle to SL_IR_IRC2:Wait

This transition shall occur after a Start SL_IR Receiver confirmation is received.

6.10.5.3 SL_IR_IRC2:Wait state

6.10.5.3.1 State description

This state ensures that an IDENTIFY address frame has been received by the SL_IR_RIF state machine and that an IDENTIFY address frame has been transmitted by the SL_IR_TIR state machine before enabling the rest of the link layer. The IDENTIFY address frames may be transmitted and received on the physical link in any order.

After this state receives an Identify Received message, it shall send a Stop SNTT request to the phy layer.

NOTE 39 If multiplexing is enabled, each SL_IR_IRC state machine sends a Stop SNTT request to the phy layer. The phy layer honors the first request and ignores the second request.

After this state receives an Identify Transmitted message, it shall initialize and start the Receive Identify Timeout timer. If an Identify Received message is received before the Receive Identify Timeout timer expires, then this state shall:

- a) send an Identification Sequence Complete confirmation to the management application layer, with arguments carrying the contents of the incoming IDENTIFY address frame;

- b) send an Enable Disable SAS Link (Enable) message to the SL state machines (see 6.15) in a SAS logical phy or the XL state machine (see 6.16) in an expander logical phy indicating that the rest of the link layer may start operation; and
- c) send a Phy Enabled confirmation to the port layer and the management application layer indicating that the phy is ready for use.

If the Receive Identify Timeout timer expires before an Identify Received message is received, then this state shall send an Identify Timeout confirmation to the management application layer to indicate that an identify timeout occurred.

If this state receives a HARD_RESET Received message before an Identify Received message is received, then this state shall send a HARD_RESET Received confirmation to the port layer and the management application layer and a Stop SNTT request to the phy layer.

If this state receives a HARD_RESET Received message after an Identify Received message is received, then the HARD_RESET Received message shall be ignored.

6.10.5.3.2 Transition SL_IR_IRC2:Wait to SL_IR_IRC3:Completed

This transition shall occur after sending a HARD_RESET Received confirmation, Identify Timeout confirmation, or an Identification Sequence Complete and a Phy Enabled confirmation.

6.10.5.4 SL_IR_IRC3:Completed state

This state waits for a Phy Layer Not Ready confirmation.

6.11 Entering a low phy power condition

Figure 105 shows the sequence to transition from the active phy power condition to a low phy power condition.

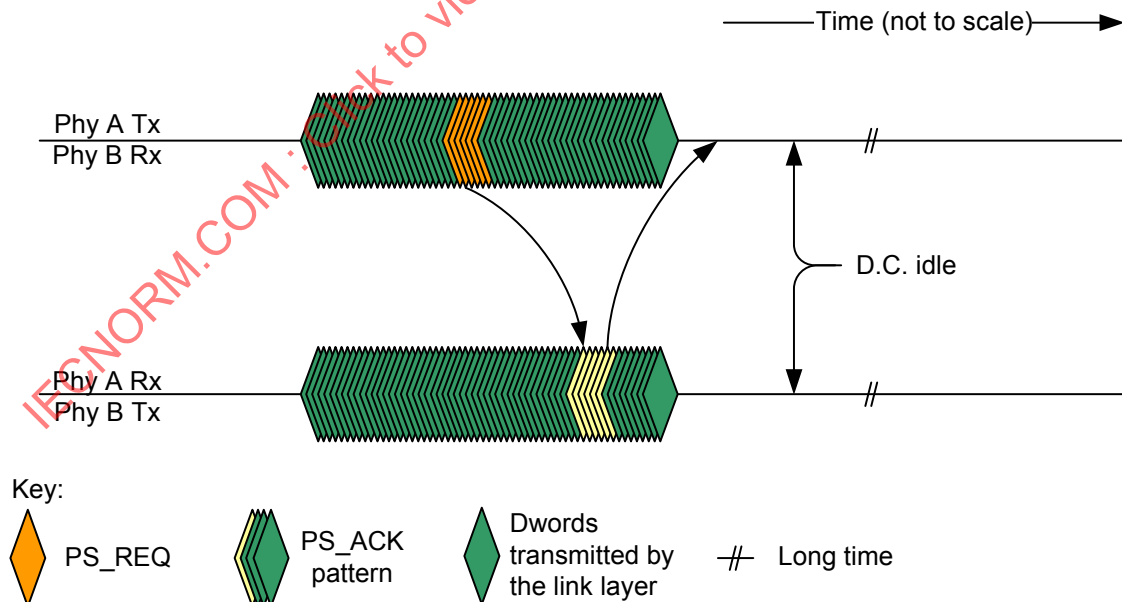


Figure 105 — Transitioning from the active phy power condition to a low phy power condition

The transition to a low phy power condition is acknowledged using a PS_ACK pattern as defined in table 94.

Table 94 — PS_ACK pattern

Pattern	Description
PS_ACK pattern	Sequence of: 1) PS_ACK primitive sequence (see 6.2.6.11); and 2) three idle dwords.

After sending a PS_ACK pattern the transmitter may continue to transmit a vendor specific number of idle dwords.

After sending a PS_REQ primitive sequence, if no PS_ACK primitive sequence is received before the Power Condition Request Timeout timer (see table 103 and table 106) expires, then the transition to the low phy power condition is aborted and the phy remains in the active phy power condition.

6.12 SAS domain changes (Broadcast (Change) usage)

An expander device shall originate Broadcast (Change) from at least one phy in each of its expander ports other than the expander port that is the cause for originating Broadcast (Change).

Expander devices shall originate Broadcast (Change) for the following expander phy-related reasons:

- after an expander phy's SP state machine transitions from the SP15:SAS_PHY_Ready state, SP22:SATA_PHY_Ready state, SP31:SAS_PS_Low_Phy_Power state, SP32:SAS_PS_ALIGN0 state, or SP33:SAS_PS_ALIGN1 state to the SP0:OOB_COMINIT state (see 5.9);

NOTE 40 This occurs when the expander phy is reset or disabled with the SMP PHY CONTROL function DISABLE, LINK RESET, HARD RESET, or TRANSMIT SATA PORT SELECTION SIGNAL phy operations (see 9.4.3.28) as well as when dword synchronization is lost for any other reason.

- after an expander phy's SP state machine reaches the SP26:SATA_SpinupHold state and sends a SATA Spinup Hold confirmation as defined in 5.9.8 and 5.12;
- after an expander phy's SP state machine sends a SATA Port Selector change confirmation to the link layer (see 5.9.3);
- after an expander phy completes the link reset sequence (see 6.9);
- after a virtual phy has been enabled or completed processing a reset requested by the SMP PHY CONTROL function LINK RESET or HARD RESET phy operations (see 9.4.3.28); and
- after an STP/SATA bridge receives an initial Register - Device to host FIS (see 8.3.1).

In zoning expander devices with zoning enabled, forwarding Broadcasts is subject to restrictions defined in 4.9.5.

In zoning expander devices with zoning enabled, a Broadcast (Change) for an expander phy-related reason shall be originated from the source zone group of the expander phy causing the Broadcast (Change) or from zone group 1.

Expander devices shall originate Broadcast (Change) for the following expander device-related reasons:

- after a self-configuring expander device has changed its SELF CONFIGURING bit from one to zero in the SMP REPORT GENERAL response (see 9.4.3.4) as described in 4.7.4. In zoning expander devices with zoning enabled, the source zone group shall be 01h; and
- after a locked expander device is unlocked (i.e., a zoning expander device has changed its ZONE CONFIGURING bit from one to zero in the SMP REPORT GENERAL response) (see 4.9.6.5, 4.9.6.6, and 9.4.3.23), with the source zone group as specified in 4.9.6.5, 4.9.6.6, and 9.4.3.23.

Expander devices shall forward Broadcast (Change) for the following reason:

- a) after an expander phy receives Broadcast (Change).

For a virtual phy, if there is any time after a reset is originated during which connection requests to the attached SAS address result in connection responses of OPEN_REJECT (NO DESTINATION), then the expander device shall originate the Broadcast (Change) twice, once at the start of the reset (i.e., when the SAS address becomes unavailable) and once at its completion (i.e., when the SAS address becomes available). If there is no such time window, then the expander device shall originate the Broadcast (Change) once.

SAS initiator ports may originate Broadcast (Change) to force other SAS initiator ports and expander ports to re-run the discover process. SAS target ports should not originate Broadcast (Change).

See 9.4.3.4 for details on counting Broadcast (Change) origination in an expander device.

6.13 Connections

6.13.1 Connections overview

A connection is opened between a SAS initiator port and a SAS target port before communication begins. A connection is established between one SAS initiator phy in the SAS initiator port and one SAS target phy in the SAS target port.

SSP initiator ports open SSP connections to transmit SCSI commands, task management functions, and transfer write data. SSP target ports open SSP connections to transfer read data, request write data, and transmit service responses.

SMP initiator ports open SMP connections to transmit SMP requests and receive SMP responses.

STP initiator ports and STP target ports open STP connections to transmit SATA frames. An STP target port in an expander device opens STP connections on behalf of SATA devices.

The OPEN address frame is used to request that a connection be opened (see 6.13.2.1). AIP primitive sequences, OPEN_ACCEPT, and OPEN_REJECT are the responses to an OPEN address frame (see 6.13.2.2). A BREAK primitive sequence is used to abort connection requests (see 6.13.6) and to break a connection (see 6.13.8). A CLOSE primitive sequence is used for orderly closing of a connection (see 6.13.7).

Connections use a single pathway from the SAS initiator phy to the SAS target phy. While a connection is open, only one pathway shall be used for that connection.

For STP connections, connections may be between the STP initiator port and an STP target port of an STP/SATA bridge in an expander device. The SATA device behind the STP/SATA bridge is not aware of SAS connection management.

A wide port may have separate connections on each of its logical phys.

6.13.2 Opening a connection

6.13.2.1 Connection request

The OPEN address frame (see 6.8.3) is used to open a connection from a source port to a destination port using one source phy (i.e., one logical phy in the source port) and one destination phy (i.e., one logical phy in the destination port).

To make a connection request, the source port shall transmit an OPEN address frame through an available logical phy (i.e., the source phy). The source phy shall transmit idle dwords after the OPEN address frame until it receives a response or aborts the connection request with a BREAK primitive sequence.

After transmitting an OPEN address frame, the source phy shall initialize and start a 1 ms Open Timeout timer. Whenever an AIP is received, the source phy shall reinitialize and restart the Open Timeout timer. Source phys are not required to enforce a limit on the number of AIPs received before aborting the connection

request. When any connection response is received, the source phy shall reinitialize the Open Timeout timer. If the Open Timeout timer expires before a connection response is received, then the source phy shall transmit a BREAK primitive sequence to abort the connection request (see 6.13.6).

The OPEN address frame flows through expander devices onto intermediate logical links. If an expander device on the pathway is unable to forward the connection request, then that expander device returns OPEN_REJECT (see 6.13.4). If the OPEN address frame reaches the destination phy, then the destination phy returns either OPEN_ACCEPT or OPEN_REJECT unless the OPEN address frame passed an OPEN address frame from the destination phy with higher arbitration priority (see 6.13.3). Rate matching shall be used on any logical links in the pathway with negotiated logical link rates that are faster than the requested connection rate (see 6.14).

A wide port should not attempt to establish more connections to a destination port than the number of phys in the destination port or the number of phys in the narrowest logical link on the pathway to the destination port. A wide port should not attempt to establish more connections than the number of phys in the narrowest common logical link on the pathways to the destination ports of those connections. Additional requirements for STP connection requests are defined in 6.18.5. Additional requirements for SMP connection requests are defined in 6.19.3.

Figure 106 shows an example of the simultaneous connection recommendations for wide ports. Multiplexing is disabled in this example.

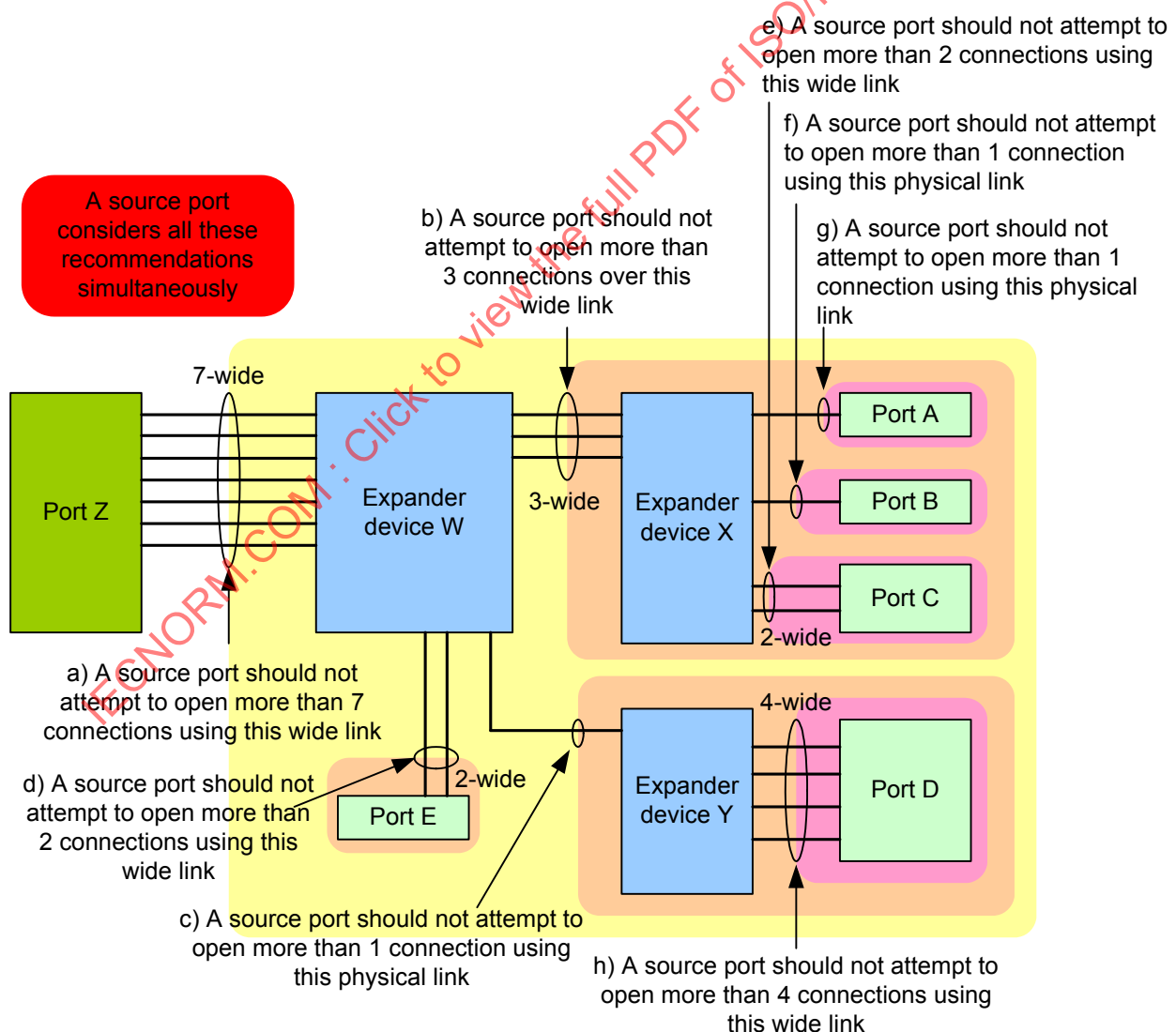


Figure 106 — Example simultaneous connection recommendations for wide ports

In figure 106, some of the recommendations are combined as follows:

- a) recommendations a), b), and e) together specify that port Z should not attempt to open more than 2 connections to port C;
- b) recommendations a), b), e), f), and g) together specify that if port Z has 2 connections open to ports A, B, and X, it should not attempt to open more than 1 connection to port C. If it has 6 connections open to ports A, B, D, E, W, X, and Y, it should not attempt to open more than 1 connection to port C; and
- c) recommendations a), c), and h) together specify that port Z should not attempt to open more than 1 connection to port D. If it has a connection open to port Y, it should not attempt to open another connection to port D until the first connection is closed.

6.13.2.2 Results of a connection request

After a logical phy transmits an OPEN address frame, it shall expect one or more of the results listed in table 95

Table 95 — Connection Results of a connection request

Result	Description
Receive AIP	Arbitration in progress. While an expander device is trying to open a connection to the selected destination port (e.g., while it is internally arbitrating for access to an expander port), the expander device returns an AIP to the source phy. The source phy shall reinitialize and restart its Open Timeout timer each time it receives an AIP.
Receive OPEN_ACCEPT	Connection request accepted. OPEN_ACCEPT is transmitted by the destination phy.
Receive OPEN_REJECT	Connection request rejected. OPEN_REJECT is transmitted by the destination phy or by an expander device in the partial pathway. The different versions are described in 6.2.6.10. See 4.5 for I_T nexus loss handling. See 6.8.3 for handling of OPEN_REJECT (CONNECTION RATE NOT SUPPORTED) for connection rates greater than 1.5 Gbps.
Receive OPEN address frame	If AIP has been previously received, then this indicates an overriding connection request. If AIP has not yet been received, then this indicates two connection requests crossing on the logical link. Arbitration fairness determines which one wins (see 6.13.3).
Receive BREAK	The destination phy or an expander device in the partial pathway may reply with a BREAK primitive sequence indicating the connection is not being established.
Open Timeout timer expires	The source phy shall abort the connection request by transmitting a BREAK primitive sequence (see 6.13.6). See 4.5 for I_T nexus loss handling.

6.13.3 Arbitration fairness

SAS supports least-recently used arbitration fairness for connection requests.

Each SAS port and expander port shall include an Arbitration Wait Time timer which counts the time from the moment when the port makes a connection request until the request is accepted or rejected. The Arbitration Wait Time timer is in the port layer state machine (see 7.2.2). The Arbitration Wait Time timer shall count in microseconds from 0 μ s to 32 767 μ s and in milliseconds from 32 768 μ s to 32 767 ms + 32 768 μ s. The Arbitration Wait Time timer shall stop incrementing when its value reaches 32 767 ms + 32 768 μ s.

A SAS port (i.e., a SAS initiator port or a SAS target port) shall start the Arbitration Wait Time timer when it transmits the first OPEN address frame (see 6.8.3) for the connection request. When the SAS port retransmits the OPEN address frame (e.g., after losing arbitration and handling an inbound OPEN address frame), it shall set the ARBITRATION WAIT TIME field to the current value of the Arbitration Wait Time timer.

A SAS port should set the Arbitration Wait Time timer to zero when it transmits the first OPEN address frame for the connection request. A SAS initiator port or SAS target port may be unfair by setting the ARBITRATION WAIT TIME field in the OPEN address frame (see 6.8.3) to a higher value than its Arbitration Wait Time timer indicates. However, an unfair SAS port shall not set the ARBITRATION WAIT TIME field to a value greater than or equal to 8000h.

The expander port that receives an OPEN address frame shall set the Arbitration Wait Time timer to the value of the incoming ARBITRATION WAIT TIME field and start the Arbitration Wait Time timer as it arbitrates for internal access to the outgoing expander port. When the expander device transmits the OPEN address frame out another expander port, it shall set the outgoing ARBITRATION WAIT TIME field to the current value of the Arbitration Wait Time timer maintained by the incoming expander port.

A SAS port shall stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when the SAS port has no more frames to send.

A SAS port shall stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when the SAS port receives one of the following connection responses:

- a) OPEN_ACCEPT;
- b) OPEN_REJECT (PROTOCOL NOT SUPPORTED);
- c) OPEN_REJECT (ZONE VIOLATION);
- d) OPEN_REJECT (RESERVED ABANDON 1);
- e) OPEN_REJECT (RESERVED ABANDON 2);
- f) OPEN_REJECT (RESERVED ABANDON 3);
- g) OPEN_REJECT (STP RESOURCES BUSY); or
- h) OPEN_REJECT (WRONG DESTINATION).

When an OPEN_REJECT (RETRY), OPEN_REJECT (RESERVED CONTINUE 0), or OPEN_REJECT (RESERVED CONTINUE 1) is received:

- a) if the CONTINUE AWT bit is set to one in the Protocol-Specific Port mode page (see 9.2.7.4), then a connection response of OPEN_REJECT (RETRY), OPEN_REJECT (RESERVED CONTINUE 0), or OPEN_REJECT (RESERVED CONTINUE 1) shall not stop the Arbitration Wait Time timer and shall not set the Arbitration Wait Time timer to zero; or
- b) If the CONTINUE AWT bit is set to zero, then a SAS port shall stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero.

A SAS port should not stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when the SAS port receives an incoming OPEN address frame that has priority over the outgoing OPEN address frame according to table 96, regardless of whether the SAS port replies with an OPEN_ACCEPT or an OPEN_REJECT.

When arbitrating for access to an outgoing expander port, the expander device shall select the connection request based on the rules described in 6.13.4.

If two connection requests pass on a logical link, then the logical phy shall determine the winner by comparing OPEN address frame field contents using the arbitration priority described in table 96.

Table 96 — Arbitration priority for OPEN address frames passing on a logical link

Bits 79-64 (79 is MSB)	Bits 63-0 (0 is LSB)
ARBITRATION WAIT TIME field value	SOURCE SAS ADDRESS field value

See 6.8.3 for details on the OPEN address frame and the ARBITRATION WAIT TIME field.

6.13.4 Arbitration inside an expander device

6.13.4.1 Expander logical phy arbitration requirements

An expander logical phy shall set its Request Path request High Priority argument to one when the expander logical phy requests a path after:

- a) the expander logical phy has forwarded an OPEN address frame to the logical link;
- b) the expander logical phy receives an OPEN address frame with higher arbitration priority (see 6.13.3); and
- c) the destination SAS address and connection rate of the received OPEN address frame are not equal to the source SAS address and connection rate of the transmitted OPEN address frame (see 6.16.4 and 6.16.9).

Otherwise, the expander logical phy shall set the High Priority argument to one.

See the XL state machine (see 6.16) for detailed expander logical phy requirements.

6.13.4.2 ECM arbitration requirements

6.13.4.2.1 ECM arbitration requirements overview

The ECM shall arbitrate and assign or deny path resources for Request Path requests (see 4.6.6.3) from each expander logical phy.

Arbitration includes adherence to the SAS arbitration fairness algorithm and path recovery. Path recovery is used to avoid potential deadlock scenarios within the SAS topology by deterministically choosing which partial pathway(s) to tear down to allow at least one connection to complete.

The ECM shall maintain an Arbitration Wait Time state machine variable for each Request Path request that the ECM is processing. The ECM shall initialize the Arbitration Wait Time state machine variable to the Arbitration Wait Time argument upon receiving the Request Path request and shall continuously update the Arbitration Wait Time state machine variable until the ECM completes responding to the Request Path request.

The Source SAS Address argument, and Connection Rate argument are filled in from the received OPEN address frame and are used with the High Priority argument and the Arbitration Wait Time state machine variable by the ECM to compare Request Path requests. The High Priority argument is used to increase the priority of a Request Path request after a Backoff Retry response is sent by an expander logical phy (see 6.16.4).

When the ECM in an expander device receives a connection request:

- 1) if the destination SAS address is that of the expander device itself, then the ECM shall arbitrate for access to its SMP target port;
- 2) if the destination SAS address matches the SAS address attached to one or more of the expander logical phys, then the ECM shall arbitrate for access to those expander logical phys;
- 3) if the destination SAS address matches an enabled SAS address in the expander route table for one or more expander logical phys that is using the table routing method, then the ECM shall arbitrate for access to those expander logical phys; and
- 4) if at least one expander logical phy is using the subtractive routing method and the request did not come from one of those expander logical phys, then the ECM shall arbitrate for access to one of those expander logical phys.

The ECM shall respond to each Request Path request by returning the following confirmations to the requesting expander logical phy while processing the Request Path request:

- a) Arbitrating (Normal) (see 6.13.4.2.2);
- b) Arbitrating (Waiting On Partial) (see 6.13.4.2.2);
- c) Arbitrating (Blocked On Partial) (see 6.13.4.2.2); and
- d) Arbitrating (Waiting On Connection) (see 6.13.4.2.2).

The ECM shall complete responding to each Request Path request by returning one of the following confirmations to the requesting expander logical phy:

- a) Arb Won (see 6.13.4.2.3);
- b) Arb Lost (see 6.13.4.2.4); or
- c) Arb Reject (see 6.13.4.2.5).

If the ECM receives an Idle request from a phy that is involved in a connection before it has received a Forward Close request from that phy and sent a Forward Close indication to that phy, then the ECM shall send a Forward Break indication to the destination phy.

6.13.4.2.2 Arbitrating confirmations

The ECM shall send an Arbitrating (Normal) confirmation after it has received a Request Path request.

The ECM shall send an Arbitrating (Waiting On Partial) confirmation if it is waiting on a partial pathway (see 4.1.11). The ECM is waiting on a partial pathway if:

- a) there is a destination port capable of routing to the requested destination SAS address;
- b) at least one expander logical phy within the destination port supports the requested connection rate;
- c) each of the expander logical phys within the destination port is returning a Phy Status (Partial Pathway) or Phy Status (Blocked Partial Pathway) response; and
- d) at least one of the expander logical phys within the destination port is returning a Phy Status (Partial Pathway) response.

The ECM shall send an Arbitrating (Blocked On Partial) confirmation if it is waiting on a blocked partial pathway (see 4.1.11). The ECM is waiting on a blocked partial pathway if:

- a) there is a destination port capable of routing to the requested destination SAS address;
- b) at least one expander logical phy within the destination port supports the requested connection rate; and
- c) each of the expander logical phys within the destination port is returning a Phy Status (Blocked Partial Pathway) response.

The ECM shall send an Arbitrating (Waiting On Connection) confirmation if it is waiting on a connection to complete (see 4.1.12). The ECM is waiting on a connection to complete if:

- a) the connection request is blocked by an active connection; or
- b) there are insufficient routing resources within the expander to complete the connection request.

A connection request shall be considered blocked by an active connection when:

- a) there is a destination port capable of routing to the requested destination SAS address;
- b) at least one expander logical phy within the destination port supports the requested connection rate;
- c) each of the expander logical phys within the destination port is returning a Phy Status (Partial Pathway) response, a Phy Status (Blocked Partial Pathway) response, a Phy Status (Breaking Connection) response, or a Phy Status (Connection) response; and
- d) at least one of the expander logical phys within the destination port is returning a Phy Status (Connection) response.

6.13.4.2.3 Arb Won confirmation

The ECM shall generate the Arb Won confirmation when all of the following conditions are met:

- a) the Request Path request maps to a destination expander logical phy that:
 - A) supports the connection rate; and
 - B) is not reporting a Phy Status (Partial Pathway) response, a Phy Status (Blocked Partial Pathway) response, a Phy Status (Breaking Connection) response, or a Phy Status (Connection) response unless that expander logical phy is arbitrating for the requesting expander logical phy;
- b) there are sufficient routing resources to complete the connection request;
- c) no higher priority Request Path requests are present with the requesting expander logical phy as the destination; and

- d) the Request Path request is the highest priority Request Path request (see table 97) mapping to the destination expander logical phy (i.e., only send one Arb Won confirmation for Request Path requests to the same destination phy).

If two or more Request Path requests contend, then the ECM shall select the request with the highest arbitration priority. The arbitration priority for a Request Path request (see table 97) consists of the following:

- a) High Priority argument;
- b) Arbitration Wait Time state machine variable;
- c) Source SAS Address argument; and
- d) Connection Rate argument.

Table 97 — Arbitration priority for a Request Path request in the ECM

Bit 84 (84 is MSB)	Bits 83-68	Bits 67-4	Bits 3-0 (0 is LSB)
High Priority argument	Arbitration Wait Time state machine variable	Source Sas Address argument	Connection Rate argument

6.13.4.2.4 Arb Lost confirmation

The ECM shall generate the Arb Lost confirmation when all of the following conditions are met:

- a) the Request Path request maps to a destination expander logical phy that:
 - A) supports the connection rate; and
 - B) is not reporting a Phy Status (Partial Pathway) response, a Phy Status (Blocked Partial Pathway) response, a Phy Status (Breaking Connection) response, or a Phy Status (Connection) response unless that expander logical phy is arbitrating for the requesting expander logical phy;
- b) there are sufficient routing resources to complete the connection request; and
- c) one of the following conditions are met:
 - A) the destination expander logical phy is making a Request Path request with the requesting expander logical phy as its destination (i.e., when two expander logical phys both receive an OPEN address frame destined for each other, the ECM provides the Arb Lost confirmation to the expander logical phy that received the lowest priority OPEN address frame); or
 - B) the ECM is sending an Arb Won confirmation to another expander logical phy that is using the requesting expander logical phy as the destination.

6.13.4.2.5 Arb Reject confirmation

The ECM shall generate one of the following Arb Reject confirmations when any of the following conditions are met and all the Arb Won conditions (see 6.13.4.2.3) are not met:

- 1) an Arb Reject (Bad Destination) confirmation if the source expander logical phy and destination expander logical phy(s) are in the same expander port and are using the direct routing method;
- 2) an Arb Reject (Retry) confirmation if the expander device is unable to process the connection request because it has reduced functionality (see 4.6.8);
- 3) if the source expander logical phy and destination expander logical phy(s) are in the same expander port and are using the table routing method or the subtractive routing method, then:
 - A) an Arb Reject (No Destination) confirmation if the expander device is not configuring (see 4.7.4); or
 - B) an Arb Reject (Retry) confirmation if the expander device is configuring;
- 4) if there are no destination expander logical phys (i.e., there is no direct routing or table routing match and there is no subtractive phy), then:
 - A) an Arb Reject (No Destination) confirmation if the expander device is not configuring; or
 - B) an Arb Reject (Retry) confirmation if the expander device is configuring;
- 5) if access to the destination expander logical phy(s) is prohibited by zoning (see 4.9.3), then:
 - A) an Arb Reject (Zone Violation) confirmation if the zoning expander device is unlocked; or
 - B) an Arb Reject (Retry) confirmation if the zoning expander device is locked;

- 6) an Arb Reject (Connection Rate Not Supported) confirmation if none of the destination expander logical phys supports the connection rate; and
- 7) an Arb Reject (Pathway Blocked) confirmation if all the destination expander logical phys that support the connection rate contain blocked partial pathways (i.e., are all returning Phy Status (Blocked Partial Pathway) confirmations) and pathway recovery rules require this Request Path request be rejected to release path resources (see 6.13.4.5).

6.13.4.3 Arbitration status

Arbitration status shall be conveyed between expander devices and by expander devices to SAS endpoints using AIP primitive sequences (see 6.2.6.1). This status is used to monitor the progress of connection attempts and to facilitate pathway recovery as part of deadlock recovery.

The arbitration status of an expander logical phy is set to the last type of AIP received.

Before an expander device transmits an AIP primitive sequence, the expander device may have transmitted an OPEN address frame on the same physical link. Arbitration fairness dictates which OPEN address frame wins (see 6.13.3).

After an expander device transmits an AIP, the expander device shall not transmit an OPEN address frame unless it has higher arbitration priority than the incoming connection request.

After transmitting an AIP primitive sequence, an expander device shall transmit at least one other dword (e.g., an idle dword) before transmitting another AIP primitive sequence.

Expander devices shall transmit at least one AIP primitive sequence every 128 dwords while originating AIP (NORMAL) primitive sequences, AIP (WAITING ON PARTIAL) primitive sequences, or AIP (WAITING ON CONNECTION) primitive sequence.

NOTE 41 Expander devices compliant with SAS-1.1 were not required to transmit three consecutive AIP primitives, as AIP was defined as a single primitive sequence (see 6.2.4.2) rather than as an extended primitive sequence (see 6.2.4.5).

Expander devices shall transmit an AIP primitive sequence (e.g., an AIP (NORMAL)) within 128 dwords of receiving an OPEN address frame.

6.13.4.4 Partial Pathway Timeout timer

Each expander logical phy shall maintain a Partial Pathway Timeout timer. This timer is used to identify potential deadlock conditions and to request resolution by the ECM. An expander logical phy shall initialize the Partial Pathway Timeout timer to the time reported in the PARTIAL PATHWAY TIMEOUT VALUE field in the SMP DISCOVER response (see 9.4.3.10) and run the Partial Pathway Timeout timer whenever the ECM provides confirmation to the expander logical phy that all expander logical phys within the requested destination port are blocked waiting on partial pathways.

NOTE 42 The partial pathway timeout value allows flexibility in specifying how long an expander device waits before attempting pathway recovery. The recommended default value (see 9.4.3.10) was chosen to cover a wide range of topologies. Selecting small partial pathway timeout value values within a large topology may compromise performance because of the time a device waits after receiving OPEN_REJECT (PATHWAY BLOCKED) before retrying the connection request. Similarly, selecting large partial pathway timeout value values within a small topology may compromise performance due to waiting longer than necessary to detect pathway blockage.

When the Partial Pathway Timeout timer is not running, an expander logical phy shall initialize and start the Partial Pathway Timeout timer when all expander logical phys within the requested destination port contain a blocked partial pathway (i.e., are returning Phy Status (Blocked Partial Pathway)).

NOTE 43 The Partial Pathway Timeout timer is not initialized and started if one or more of the expander logical phys within a requested destination port are being used for a connection.

When one of the conditions above is not met, the expander logical phy shall stop the Partial Pathway Timeout timer. If the timer expires, then pathway recovery shall occur (see 6.13.4.5).

6.13.4.5 Pathway recovery

Pathway recovery provides a means to abort connection requests in order to prevent deadlock using pathway recovery priority comparisons. Pathway recovery priority comparisons compare the PATHWAY BLOCKED COUNT fields and SOURCE SAS ADDRESS fields of the OPEN address frames of the blocked connection requests as described in table 98.

Table 98 — Pathway recovery priority

Bits 71-64 (71 is MSB)	Bits 63-0 (0 is LSB)
PATHWAY BLOCKED COUNT field value	SOURCE SAS ADDRESS field value

When the Partial Pathway Timeout timer for an arbitrating expander logical phy expires (i.e., reaches a value of zero), the ECM shall determine whether to continue the connection request or to abort the connection request.

The ECM shall reply to a connection request with Arb Reject (Pathway Blocked) when:

- the Partial Pathway Timeout timer expires; and
- the pathway recovery priority of the arbitrating expander logical phy (i.e., the expander logical phy requesting the connection) is less than or equal to the pathway recovery priority of any of the expander logical phys within the destination port that are sending Phy Status (Blocked Partial Pathway) responses to the ECM.

The pathway blocked count and source SAS address values used to form the pathway recovery priority of a destination phy are those of the Request Path request if the expander logical phy sent a Request Path request to the ECM or those of the Forward Open indication if the expander logical phy received a Forward Open indication from the ECR.

6.13.5 BREAK handling

A logical phy aborts a connection request (see 6.13.6) and breaks a connection (see 6.13.8) by transmitting a BREAK primitive sequence.

Logical phys shall enable the BREAK_REPLY method of responding to received BREAK primitive sequences when:

- the BREAK_REPLY CAPABLE bit transmitted by the logical phy in the outgoing IDENTIFY address frame is set to one; and
- the BREAK_REPLY CAPABLE bit received by the logical phy in the incoming IDENTIFY address frame is set to one.

Logical phys shall disable the BREAK_REPLY method of responding to received BREAK primitive sequences if the BREAK_REPLY CAPABLE bit received by the logical phy in the incoming IDENTIFY address frame is set to zero.

Logical phys contained within SAS devices or expander devices that are compliant with this standard shall set the BREAK_REPLY CAPABLE bit to one in their outgoing IDENTIFY address frame.

If the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled, then the logical phy shall transmit a BREAK_REPLY primitive sequence in response to a received BREAK primitive sequence.

If the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled, then the logical phy shall transmit a BREAK primitive sequence in response to a received BREAK primitive sequence.

NOTE 44 Phys compliant with SAS-1.1 do not set the BREAK_REPLY CAPABLE bit to one in their outgoing IDENTIFY address frames.

6.13.6 Aborting a connection request

BREAK may be used to abort a connection request. The source phy shall transmit a BREAK primitive sequence after the Open Timeout timer expires or if the source phy chooses to abort its request for any other reason before a connection is established.

After transmitting a BREAK primitive sequence, the source phy shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer.

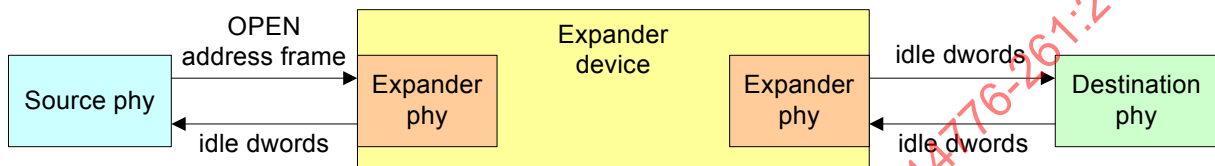
After a source phy transmits a BREAK primitive sequence to abort a connection request, the source phy shall expect one of the results listed in table 99.

Table 99 — Results of aborting a connection request

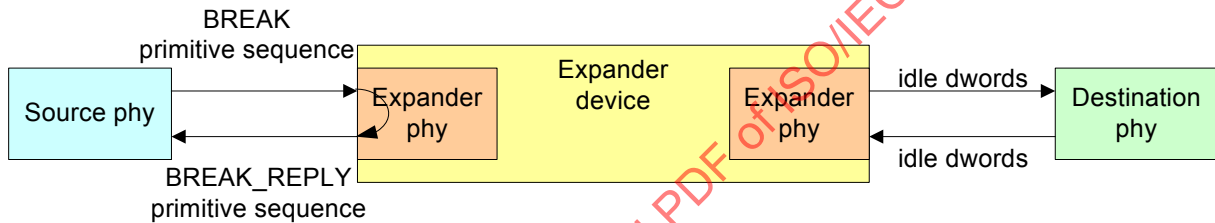
BREAK_REPLY method of responding to received BREAK primitive sequences	Result	Description
Disabled	Receive a BREAK primitive sequence	This confirms that the connection request has been aborted.
	Receive a BREAK_REPLY primitive sequence	Ignore the BREAK_REPLY primitive sequence.
Enabled	Receive a BREAK primitive sequence	The originating phy shall transmit a BREAK_REPLY primitive sequence and wait to receive a BREAK_REPLY primitive sequence or for the BREAK Timeout timer to expire.
	Receive a BREAK_REPLY primitive sequence	This confirms that the connection request has been aborted.
Enabled or disabled	Break Timeout timer expires	The originating phy shall assume the connection request has been aborted.

When a logical phy transmitting a BREAK primitive sequence is attached to an expander device, the BREAK or BREAK_REPLY response to the logical phy is generated by the expander logical phy to which the logical phy is attached, not the other SAS logical phy in the connection. If the expander device has transmitted a connection request to the destination, then the expander device shall also transmit a BREAK primitive sequence to the destination. If the expander device has not transmitted a connection request to the destination, then the expander device shall not transmit a BREAK primitive sequence to the destination. After transmitting a BREAK primitive sequence or BREAK_REPLY primitive sequence back to the source phy, the expander device shall ensure that a connection response does not occur (i.e., the expander device shall no longer forward dwords from the destination). Figure 107 shows an example of BREAK usage. Multiplexing is disabled on all phys in the example.

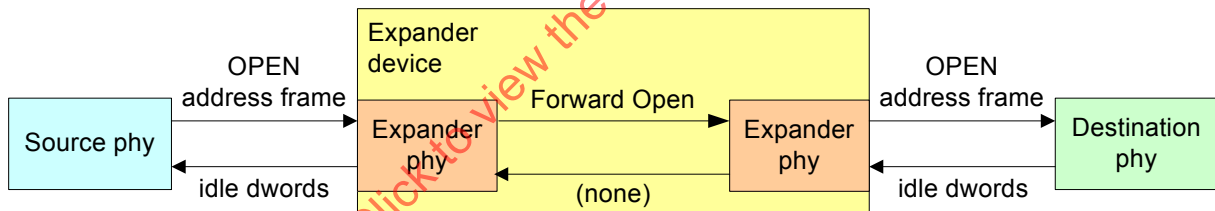
Case 1: OPEN address frame has not propagated through the expander device:



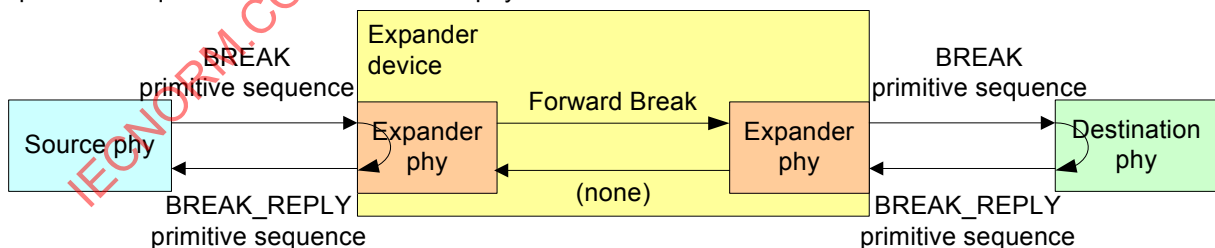
Case 1 result: expander device transmits a BREAK_REPLY primitive sequence to the source phy



Case 2: OPEN address frame has propagated through the expander device:



Case 2 result: Expander device transmits a BREAK_REPLY primitive sequence to the source phy and a BREAK primitive sequence to the destination phy, then waits for a BREAK_REPLY primitive sequence from the destination phy



Note: If the BREAK_REPLY method of responding to BREAK primitive sequences is disabled, phys transmit a BREAK primitive sequence rather than a BREAK_REPLY primitive sequence in response to a BREAK primitive sequence .

Figure 107 — Aborting a connection request with a BREAK primitive sequence

Figure 108 shows the sequence for a connection request where the Open Timeout timer expires.

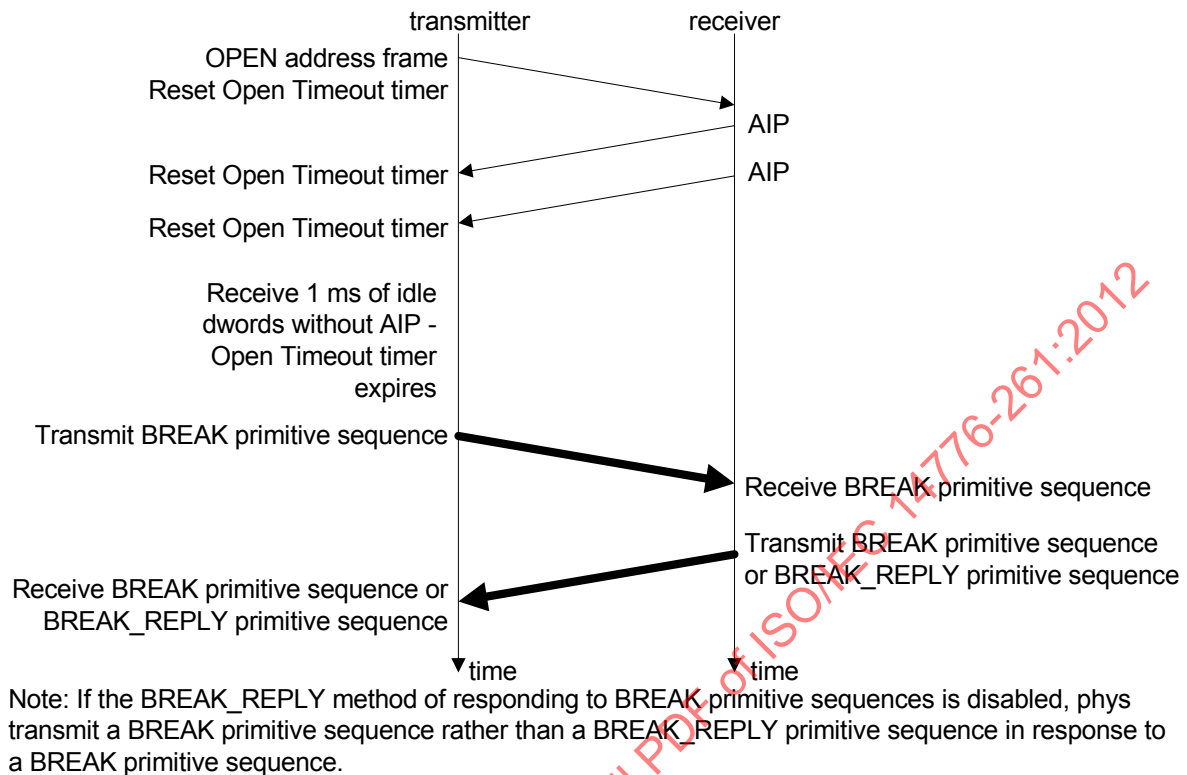


Figure 108 — Connection request timeout example

6.13.7 Closing a connection

CLOSE is used to close a connection of any protocol. See 6.17.7 for details on closing SSP connections, 6.18.6 for details on closing STP connections, and 6.19.4 for details on closing SMP connections.

After transmitting a CLOSE primitive sequence, the originating phy shall initialize a Close Timeout timer to 1 ms and start the Close Timeout timer.

After the logical phy transmits a CLOSE primitive sequence to close a connection, logical phy shall expect one of the results listed in table 100.

Table 100 — Results of closing a connection

Result	Description
Receive a CLOSE primitive sequence	This confirms that the connection has been closed.
Close Timeout timer expires	The originating phy shall attempt to break the connection (see 6.13.8).

No additional dwords for the connection shall follow the CLOSE primitive sequence. Expander devices shall close the full-duplex connection upon forwarding a CLOSE primitive sequence in each direction.

When a logical phy has both transmitted and received a CLOSE primitive sequence, the logical phy shall consider the connection closed.

Figure 109 shows example sequences for closing a connection.

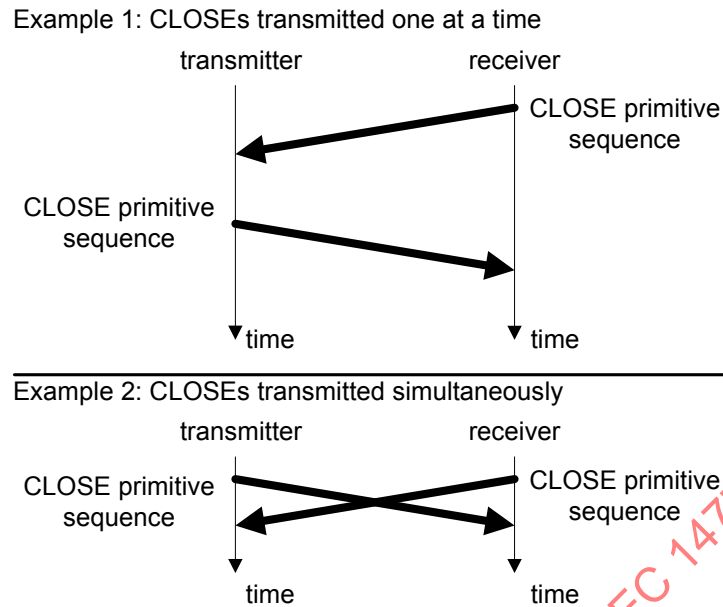


Figure 109 — Closing a connection example

6.13.8 Breaking a connection

In addition to aborting a connection request, a BREAK primitive sequence may also be used to break a connection in cases where CLOSE is not available. After transmitting a BREAK primitive sequence, the originating phy shall ignore all incoming dwords except for BREAKs, BREAK_REPLYS, and deletable primitives.

After transmitting a BREAK primitive sequence, the originating phy shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer.

After a logical phy transmits a BREAK primitive sequence to break a connection, the logical phy shall expect one of the results listed in table 101.

Table 101 — Results of breaking a connection

BREAK_REPLY method of responding to received BREAK primitive sequences	Result	Description
Disabled	Receive a BREAK primitive sequence	This confirms that the connection has been broken.
	Receive BREAK_REPLY	Ignore the BREAK_REPLY.
Enabled	Receive a BREAK primitive sequence	The originating phy shall transmit a BREAK_REPLY primitive sequence and wait to receive a BREAK_REPLY primitive sequence or for the BREAK Timeout timer to expire.
	Receive a BREAK_REPLY primitive sequence	This confirms that the connection has been broken.
Enabled or disabled	Break Timeout timer expires	The originating phy shall assume the connection has been broken. The originating phy may perform a link reset sequence.

In addition to a BREAK, a connection is considered broken if a link reset sequence starts (i.e., the SP state machine transitions from SP15:SAS_PHY_Ready or SP22:SATA_PHY_Ready to SP0:OOB_COMINIT (see 5.9)).

See 6.17.6 for additional rules on breaking an SSP connection.

6.14 Rate matching

Each successful connection request contains the connection rate (see 4.1.12) of the pathway.

Each logical phy in the pathway shall insert deletable primitives between dwords if the logical phy's logical link rate is faster than the connection rate as described in table 102.

Table 102 — Rate matching deletable primitive insertion requirements

Logical link rate	Connection rate	Requirement
1.5 Gbps	1.5 Gbps	None
3 Gbps	1.5 Gbps	One deletable primitive within every 2 dwords that are not physical link rate tolerance management deletable primitives (i.e., every overlapping window of 2 dwords) (e.g., a repeating pattern of a deletable primitive followed by a dword, or a repeating pattern of a dword followed by an deletable primitive)
	3 Gbps	None
6 Gbps	1.5 Gbps	Three deletable primitives within every 4 dwords that are not physical link rate tolerance management deletable primitives (i.e., 3 in every overlapping window of 4 dwords)
	3 Gbps	One deletable primitive within every 2 dwords that are not physical link rate tolerance management deletable primitives (i.e., every overlapping window of 2 dwords) (e.g., a repeating pattern of a deletable primitive followed by a dword, or a repeating pattern of a dword followed by an deletable primitive)
	6 Gbps	None

Deletable primitives inserted for rate matching are in addition to deletable primitives inserted for physical link rate tolerance management (see 6.3). See Annex G for a summary of their combined requirements.

Figure 110 shows an example of rate matching between a 3 Gbps originating phy and a 3 Gbps receiving phy, with an intermediate 1.5 Gbps physical link in between them. Multiplexing is disabled in this example.

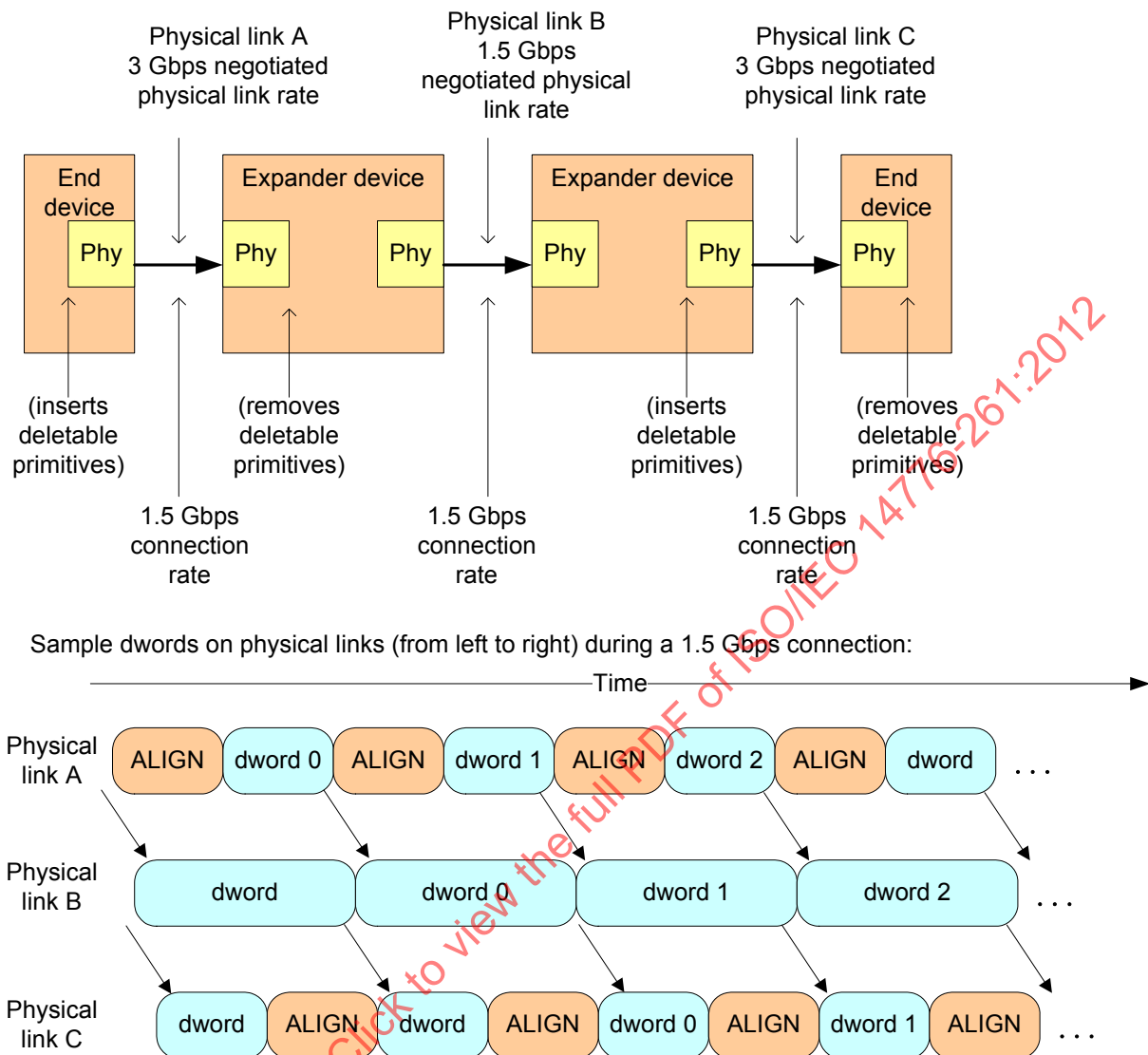


Figure 110 — Rate matching example

A logical phy originating dwords shall start rate matching at the selected connection rate starting with the first dword that is not a deletable primitive inserted for physical link rate tolerance management following:

- transmitting the EOAF for an OPEN address frame; or
- transmitting an OPEN_ACCEPT.

An expander logical phy forwarding dwords shall not insert deletable primitives for rate matching based on counting dwords transmitted, and shall insert deletable primitives whenever it underflows.

The source phy transmits idle dwords including deletable primitives at the selected connection rate while waiting for the connection response. This enables each expander device to start forwarding dwords from the source phy to the destination phy after forwarding an OPEN_ACCEPT.

A logical phy shall stop inserting deletable primitives for rate matching after:

- transmitting the first dword in a CLOSE primitive sequence;
- transmitting the first dword in a BREAK primitive sequence;
- transmitting the first dword in a BREAK_REPLY primitive sequence;
- receiving an OPEN_REJECT for a connection request; or
- losing arbitration to a received OPEN address frame.

6.15 SL (link layer for SAS logical phys) state machines

6.15.1 SL state machines overview

The SL (link layer for SAS logical phys) state machines control connections, handling both connection requests (i.e., OPEN address frames), CLOSEs, and BREAKs. The SL state machines are as follows:

- a) SL_RA (receive OPEN address frame) state machine (see 6.15.3); and
- b) SL_CC (connection control) state machine (see 6.15.4).

All the SL state machines shall begin after receiving an Enable Disable SAS Link (Enable) message from the SL_IR state machines.

If a state machine consists of multiple states, then the initial state is as indicated in the state machine description.

Figure 111, figure 112, and figure 113 show the SL state machines.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

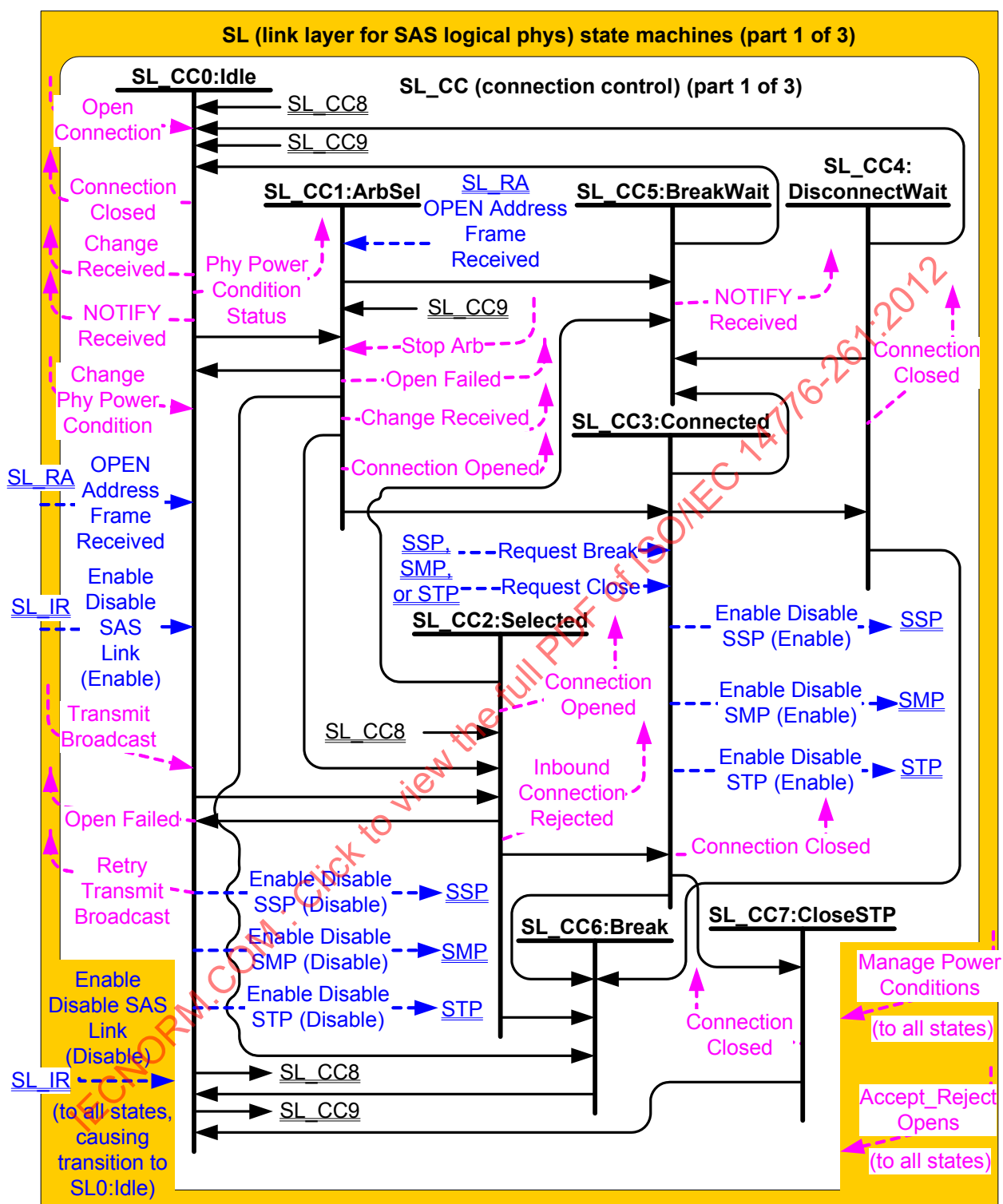


Figure 111 — SL (link layer for SAS logical phys) state machines (part 1)

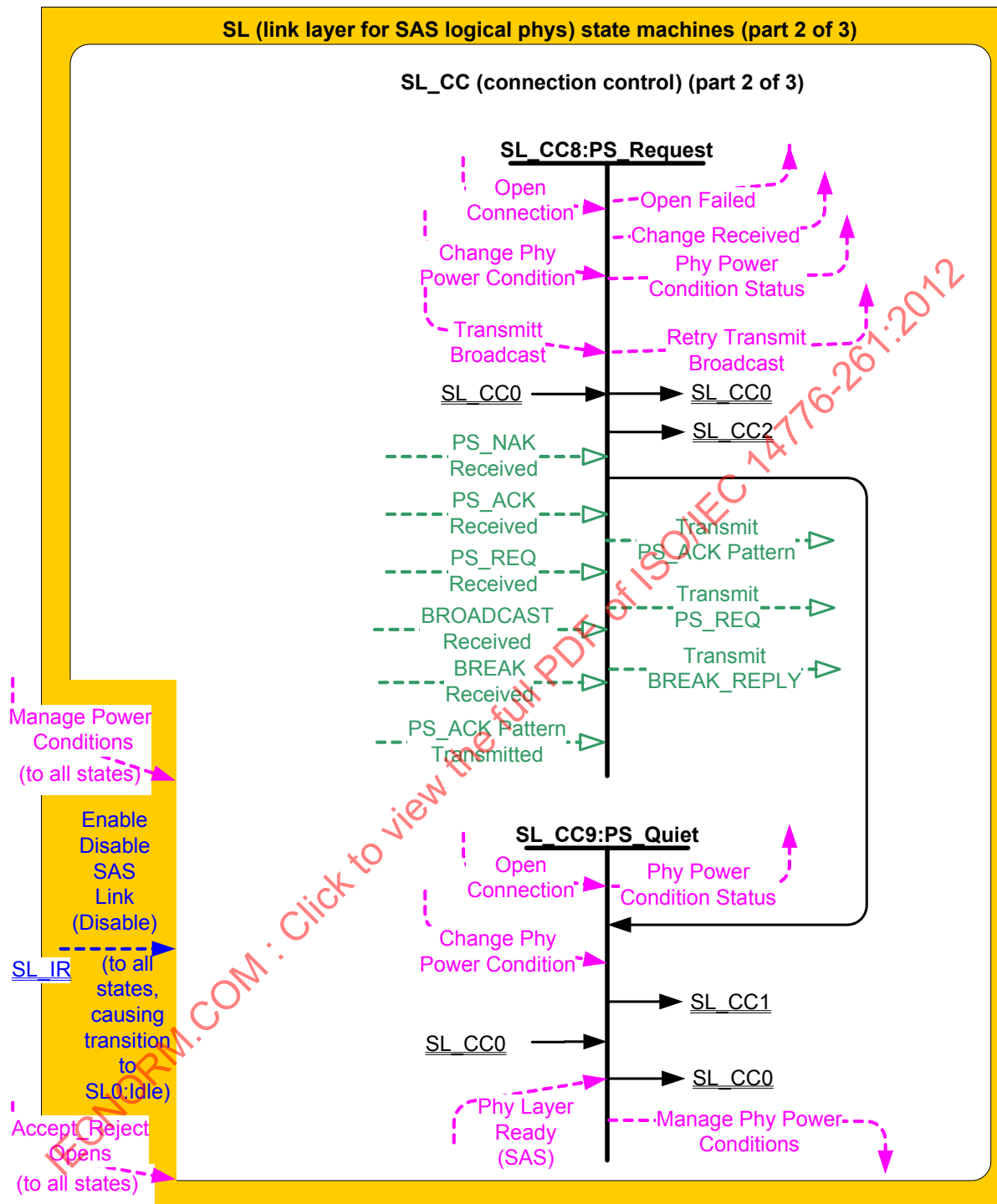


Figure 112 — SL (link layer for SAS logical phys) state machines (part 2)

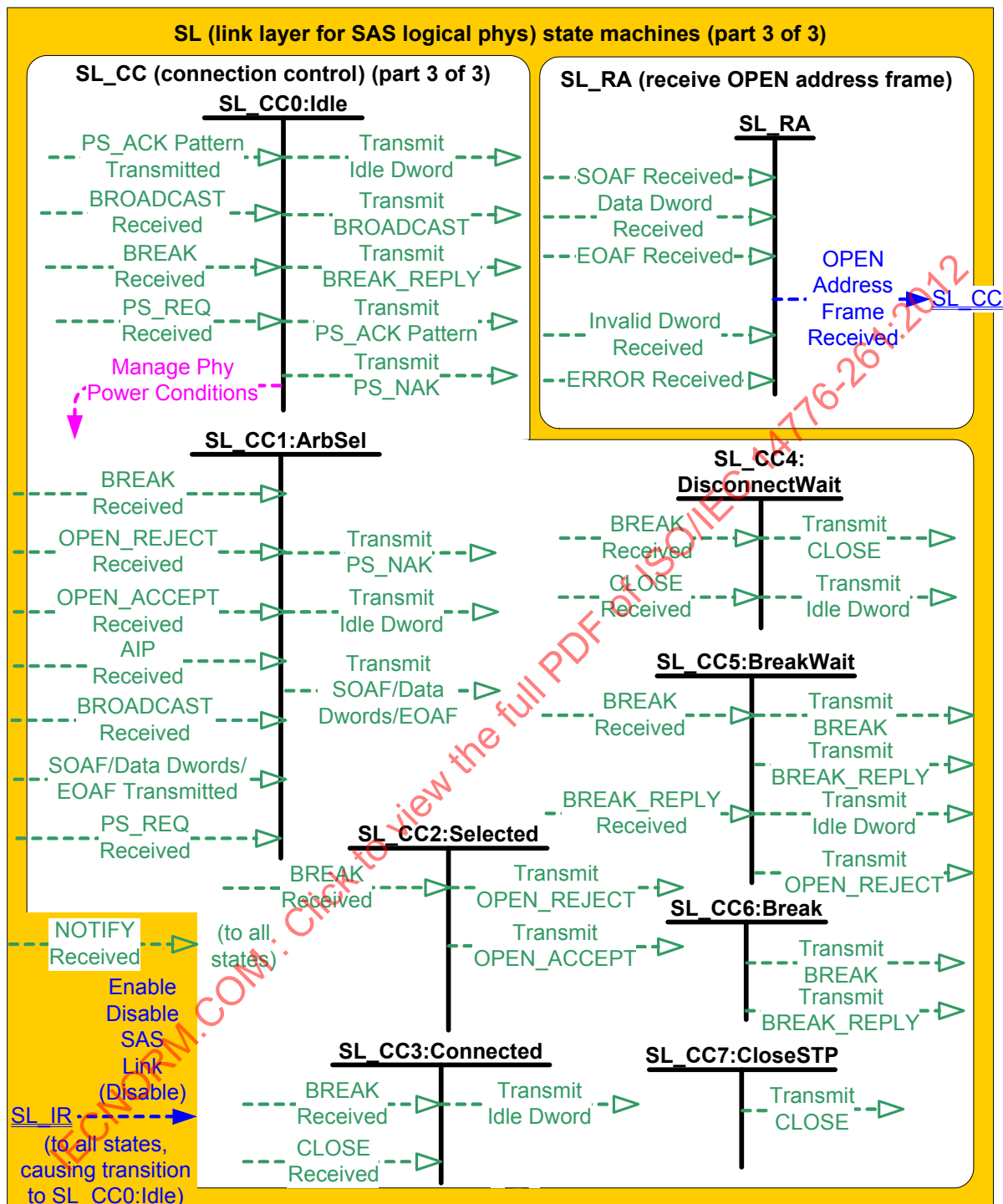


Figure 113 — SL (link layer for SAS logical phys) state machines (part 3)

6.15.2 SL transmitter and receiver

The SL transmitter receives the following messages from the SL state machines specifying primitive sequences, frames, and dwords to transmit:

- Transmit Idle Dword;
- Transmit SOAF/Data Dwords/EOAF;
- Transmit OPEN_ACCEPT;

- d) Transmit OPEN_REJECT with an argument indicating the specific type (e.g., Transmit OPEN_REJECT (Retry));
- e) Transmit BREAK;
- f) Transmit BREAK_REPLY;
- g) Transmit PS_REQ with an argument indicating the specific type (e.g., Transmit PS_REQ (Partial) or Transmit PS_REQ (Slumber));
- h) Transmit PS_ACK Pattern (see 6.11);
- i) Transmit PS_NAK;
- j) Transmit BROADCAST; and
- k) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal)).

When the SL transmitter is requested to transmit a dword from any state within any of the SL state machines, the SL transmitter shall transmit that dword. If there are multiple requests to transmit, then the following priority should be followed when selecting the dword to transmit:

- 1) BREAK_REPLY;
- 2) BREAK;
- 3) CLOSE;
- 4) OPEN_ACCEPT or OPEN_REJECT;
- 5) SOAF, data dword, or EOAF;
- 6) PS_REQ, PS_ACK pattern, or PS_NAK; and
- 7) idle dword.

When there is no outstanding message specifying a dword to transmit, the SL transmitter shall transmit idle dwords.

The SL transmitter sends the following message to the SL state machines based on dwords that have been transmitted:

- a) SOAF/Data Dwords/EOAF Transmitted; and
- b) PS_ACK Pattern Transmitted.

The SL receiver sends the following messages to the SL state machines indicating primitive sequences and dwords received from the SP_DWS receiver (see 5.10.2):

- a) SOAF Received;
- b) Data Dword Received;
- c) EOAF Received;
- d) BROADCAST Received with an argument indicating the specific type (e.g., BROADCAST Received (Change));
- e) BREAK Received;
- f) BREAK_REPLY Received;
- g) OPEN_ACCEPT Received;
- h) OPEN_REJECT Received with an argument indicating the specific type (e.g., OPEN_REJECT Received (No Destination));
- i) AIP Received;
- j) CLOSE Received with an argument indicating the specific type (e.g., CLOSE Received (Normal));
- k) NOTIFY Received (Power Loss Expected);
- l) PS_REQ Received with an argument indicated the specific type (e.g., PS_REQ Received (Partial) or PS_REQ Received (Slumber));
- m) PS_ACK Received;
- n) PS_NAK Received;
- o) ERROR Received; and
- p) Invalid Dword Received.

The SL receiver shall ignore all other dwords.

The SL transmitter relationship to other transmitters is defined in 4.3.2. The SL receiver relationship to other receivers is defined in 4.3.3.

6.15.3 SL_RA (receive OPEN address frame) state machine

The SL_RA state machine's function is to receive address frames and determine if each received address frame is a valid OPEN address frame. This state machine consists of one state.

This state machine receives SOAFs, dwords of an OPEN address frames, and EOAFs.

This state machine shall ignore all messages except SOAF Received, Data Dword Received, and EOAF Received.

If this state machine receives a subsequent SOAF Received message after receiving an SOAF Received message but before receiving an EOAF Received message, then this state machine shall discard the address frame in progress.

If this state machine receives more than eight Data Dword Received messages (i.e., 32 bytes) after an SOAF Received message and before an EOAF Received message, then this state machine shall discard the address frame.

If this state machine receives an Invalid Dword Received message or an ERROR Received message after an SOAF Received message and before an EOAF Received message, then this state machine shall either:

- a) ignore the invalid dword or ERROR; or
- b) discard the address frame.

After receiving an EOAF Received message, this state machine shall check if the address frame is a valid OPEN address frame.

This state machine shall accept an address frame if:

- a) the ADDRESS FRAME TYPE field is set to 1h (i.e., OPEN);
- b) the number of data dwords between the SOAF and EOAF is 8; and
- c) the CRC field contains a good CRC.

Otherwise, this state machine shall discard the address frame. If the frame is not discarded then this state machine shall send a OPEN Address Frame Received message to the SL_CC0:Idle state and the SL_CC1:ArbSel state with an argument that contains all the data dwords received in the OPEN address frame.

6.15.4 SL_CC (connection control) state machine

6.15.4.1 SL_CC state machine overview

The SL_CC state machine consists of the following states:

- a) SL_CC0:Idle (see 6.15.4.2) (initial state);
- b) SL_CC1:ArbSel (see 6.15.4.3);
- c) SL_CC2:Selected (see 6.15.4.4);
- d) SL_CC3:Connected (see 6.15.4.5);
- e) SL_CC4:DisconnectWait (see 6.15.4.6);
- f) SL_CC5:BreakWait (see 6.15.4.7);
- g) SL_CC6:Break (see 6.15.4.8);
- h) SL_CC7:CloseSTP (see 6.15.4.9);
- i) SL_CC8:PS_Request (see 6.15.4.10); and
- j) SL_CC9:PS_Quiet (see 6.15.4.11).

This state machine receives the following requests from the management application layer:

- a) Transmit Broadcast.

This state machine shall start in the SL_CC0:Idle state. The state machine shall transition to the SL_CC0:Idle state from any other state after receiving an Enable Disable SAS Link (Disable) message from the SL_IR state machines (see 6.10).

This machine receives the following messages from the SSP link layer state machine (see 6.17.8), the STP link layer state machine, and SMP link layer state machine (see 6.19.5):

- a) Request Break; and
- b) Request Close.

This state machine sends the following messages to the SSP link layer state machine, the STP link layer state machine, and SMP link layer state machine:

- a) Enable Disable SSP (Enable);
- b) Enable Disable SSP (Disable);
- c) Enable Disable STP (Enable);
- d) Enable Disable STP (Disable);
- e) Enable Disable SMP (Enable); and
- f) Enable Disable SMP (Disable).

This state machine receives the following messages from the SL_IR state machines (see 6.10):

- a) Enable Disable SAS Link (Enable); and
- b) Enable Disable SAS Link (Disable).

Any message received by a state that is not referred to in the description of that state or in this subclause shall be ignored.

If this state machine receives an Accept_Reject OPENS (Accept SSP) request, then this state machine shall set the Reject SSP Opens state machine variable (see table 104) to NO. If this state machine receives an Accept_Reject OPENS (Reject SSP) request, then this state machine shall set the Reject SSP Opens state machine variable to YES.

If this state machine receives an Accept_Reject OPENS (Accept SMP) request, then this state machine shall set the Reject SMP Opens state machine variable (see table 104) to NO. If this state machine receives an Accept_Reject OPENS (Reject SMP) request, then this state machine shall set the Reject SMP Opens state machine variable to YES.

If this state machine receives an Accept_Reject OPENS (Accept STP) request, then this state machine shall set the Reject STP Opens state machine variable (see table 104) to NO. If this state machine receives an Accept_Reject OPENS (Reject STP) request, then this state machine shall set the Reject STP Opens state machine variable to YES.

If this state machine receives a Manage Power Conditions (Accept Partial) request from the management application layer, then this state machine shall set the Accept Partial state machine variable (see table 104) to YES. If this state machine receives a Manage Power Conditions (Accept Slumber) request from the management application layer, then this state machine shall set the Accept Slumber state machine variable (see table 104) to YES.

If this state machine receives a Manage Power Conditions (Reject Partial) request from the management application layer, then this state machine shall set the Accept Partial state machine variable to NO. If this state machine receives a Manage Power Conditions (Reject Slumber) request from the management application layer, then this state machine shall set the Accept Slumber state machine variable to NO.

The default value of the Accept Partial state machine variable shall be set to NO and default value of the Accept Slumber state machine variable shall be set to NO.

Any detection of an internal error shall cause the SL_CC state machine to transition to the SL_CC5:BreakWait state.

This state machine shall maintain the timers listed in table 103.

Table 103 — SL_CC state machine timers

Timer	Initial value
Open Timeout timer	1 ms
Close Timeout timer	1 ms
Break Timeout timer	1 ms
Power Condition Request Timeout timer	1 ms

This state machine shall maintain the state machine variables listed in table 104.

Table 104 — SL_CC state machine variables

State machine variable	Description
Reject SSP Opens	Used to determine if the SCSI application layer is permitting SSP connection requests to be accepted on this logical phy.
Reject SMP Opens	Used to determine if the management application layer is permitting SMP connection requests to be accepted on this logical phy.
Reject STP Opens	Used to determine if the ATA application layer is permitting STP connection requests to be accepted on this logical phy.
Accept Partial	Used to determine if the management application layer is permitting this phy to enter a partial phy power condition.
Accept Slumber	Used to determine if the management application layer is permitting this phy to enter a slumber phy power condition.

6.15.4.2 SL_CC0:Idle state

6.15.4.2.1 State description

This state is the initial state and is the state that is used when there is no connection pending or established.

Upon entry into this state, this state shall send:

- an Enable Disable SSP (Disable) message to the SSP link layer state machines;
- an Enable Disable SMP (Disable) message to the SMP link layer state machines;
- an Enable Disable STP (Disable) message to the STP link layer state machines; and
- a Connection Closed (Transition to Idle) confirmation to the port layer.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter (see 6.4).

If a BROADCAST Received (Change) message, BROADCAST Received (Reserved Change 0) message, or BROADCAST Received (Reserved Change 1) message is received, then this state shall send a Change Received confirmation to the management application layer.

If a Transmit Broadcast request is received with any argument and this state has not sent a Transmit PS_ACK Pattern message, then this state shall send a Transmit BROADCAST message with the same argument to the SL transmitter. If a Transmit Broadcast request is received and this state has sent a Transmit PS_ACK Pattern message to the SL transmitter, then this state shall send a Retry Transmit Broadcast confirmation to the management application layer.

If the Transmit Broadcast request is received coincident with receiving either an Open Connection request or an OPEN Address Frame Received message, then this state shall send a Transmit BROADCAST message with the same argument to the SL transmitter before transitioning to another state.

If a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the SL transmitter.

After this state receives an Enable Disable SAS Link (Enable) message, this state shall:

- a) set the Reject SSP Opens state machine variable to a vendor specific default value (i.e., YES or NO);
- b) set the Reject SMP Opens state machine variable to a vendor specific default value (i.e., YES or NO); and
- c) set the Reject STP Opens state machine variable to a vendor specific default value (i.e., YES or NO).

If this state receives a NOTIFY Received (Power Loss Expected) message and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port), then this state shall send a Notify Received (Power Loss Expected) confirmation to the port layer.

If a PS_REQ Received (Partial) message is received and the Accept Partial state machine variable is set to YES, then this state shall:

- a) send a Transmit PS_ACK Pattern message to the SL transmitter; and
- b) send a Manage Phy Power Conditions (Stop DWS) request to the phy layer.

If a PS_REQ Received (Slumber) message is received and the Accept Slumber state machine variable is set to YES, then this state shall:

- a) send a Transmit PS_ACK Pattern message to the SL transmitter; and
- b) send a Manage Phy Power Conditions (Stop DWS) request to the phy layer.

If a PS_REQ Received (Partial) message is received and the Accept Partial state machine variable is set to NO, then this state shall send a Transmit PS_NAK message to the SL transmitter. If a PS_REQ Received (Slumber) message is received and the Accept Slumber state machine variable is set to NO, then this state shall send a Transmit PS_NAK message to the SL transmitter.

If this state receives an Open Connection request and this state has sent a Transmit PS_ACK Pattern message to the SL transmitter, then this state shall send an Open Failed (Low Phy Power Condition) confirmation to the port layer.

If this state receives a Change Phy Power Condition request and this state has sent a Transmit PS_ACK Pattern message to the SL transmitter, then this state shall send a Phy Power Condition Status (Retry Change Phy Power Condition Request) confirmation to the management application layer.

This state shall ignore any Change Phy Power Condition (Exit Power Condition) requests.

6.15.4.2.2 Transition SL_CC0:Idle to SL_CC1:ArbSel

If this state has not sent a Transmit PS_ACK Pattern message to the SL transmitter, then this transition shall occur after receiving:

- a) an Enable Disable SAS Link (Enable) message; and
- b) an Open Connection request.

The Open Connection request includes these arguments:

- a) Initiator Port Bit;
- b) Protocol;
- c) Connection Rate;
- d) Initiator Connection Tag.
- e) Destination SAS Address;
- f) Source SAS Address;
- g) Pathway Blocked Count; and
- h) Arbitration Wait Time.

6.15.4.2.3 Transition SL_CC0:Idle to SL_CC2:Selected

This transition shall occur after receiving both an Enable Disable SAS Link (Enable) message and an OPEN Address Frame Received message.

6.15.4.2.4 Transition SL_CC0:Idle to SL_CC8:PS_Request

If this state has not sent a Transmit PS_ACK Pattern message to the SL transmitter, then this transition shall occur after receiving:

- a) an Enable Disable SAS Link (Enable) message: and
- b) a Change Phy Power Condition (Enter Slumber) or Change Phy Power Condition (Enter Partial) request from the management application layer.

This transition shall include the Phy Power Condition (Enter Partial) argument or Phy Power Condition (Enter Slumber) argument that corresponds to the Change Phy Power Condition request.

6.15.4.2.5 Transition SL_CC0:Idle to SL_CC9:PS_Quiet

This transition shall occur after receiving a PS_ACK Pattern Transmitted message from the SL transmitter.

This transition shall include the Phy Power Condition (Enter Partial) argument or Phy Power Condition (Enter Slumber) argument that corresponds to the PS_REQ Received message.

6.15.4.3 SL_CC1:ArbSel state

6.15.4.3.1 State description

This state is used to make a connection request.

Upon entry into this state, this state shall:

- 1) request an OPEN address frame be transmitted by sending a Transmit SOAF/Data Dwords/EOAF message to the SL transmitter with the dwords containing the OPEN address frame with its fields set to the arguments received with the Open Connection request;
- 2) initialize and start the Open Timeout timer; and
- 3) request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter.

See 6.13 for details on rate matching when opening a connection.

This state shall ignore OPEN_REJECT Received and OPEN_ACCEPT Received messages from the time a Transmit SOAF/Data Dwords/EOAF message is sent to the SL transmitter until an SOAF/Data Dwords/EOAF Transmitted message is received from the SL transmitter.

If a BROADCAST Received (Change) message, BROADCAST Received (Reserved Change 0) message, or BROADCAST Received (Reserved Change 1) message is received, then this state shall send a Change Received confirmation to the management application layer.

If a PS_REQ Received message is received, then this state shall send a Transmit PS_NAK to the SL transmitter.

If an AIP Received message is received after requesting the OPEN address frame be transmitted, then this state shall reinitialize and restart the Open Timeout timer. The state machine shall not enforce a limit on the number of AIPs received.

If a Stop Arb request is received, then this state shall send an Open Failed (Arb Stopped) confirmation to the port layer.

If there is no response to the OPEN address frame before the Open Timeout timer expires, then this state shall send an Open Failed (Open Timeout Occurred) confirmation to the port layer.

If a BREAK Received message is received, then this state shall send an Open Failed (Break Received) confirmation to the port layer.

If this state receives an OPEN_REJECT Received message listed in table 105 after transmitting the OPEN address frame, then this state shall send the corresponding Open Failed confirmation listed in table 105 to the port layer.

Table 105 — OPEN_REJECT Received message to Open Failed confirmation mapping

OPEN_REJECT Received message	Open Failed confirmation
OPEN_REJECT Received (Bad Destination)	Open Failed (Bad Destination)
OPEN_REJECT Received (Connection Rate Not Supported)	Open Failed (Connection Rate Not Supported)
OPEN_REJECT Received (Protocol Not Supported)	Open Failed (Protocol Not Supported)
OPEN_REJECT Received (Reserved Abandon 1)	Open Failed (Reserved Abandon 1)
OPEN_REJECT Received (Reserved Abandon 2)	Open Failed (Reserved Abandon 2)
OPEN_REJECT Received (Reserved Abandon 3)	Open Failed (Reserved Abandon 3)
OPEN_REJECT Received (STP Resources Busy)	Open Failed (STP Resources Busy)
OPEN_REJECT Received (Wrong Destination)	Open Failed (Wrong Destination)
OPEN_REJECT Received (Zone Violation)	Open Failed (Zone Violation)
OPEN_REJECT Received (No Destination)	Open Failed (No Destination)
OPEN_REJECT Received (Pathway Blocked)	Open Failed (Pathway Blocked)
OPEN_REJECT Received (Reserved Continue 0)	Open Failed (Reserved Continue 0)
OPEN_REJECT Received (Reserved Continue 1)	Open Failed (Reserved Continue 1)
OPEN_REJECT Received (Reserved Initialize 0)	Open Failed (Reserved Initialize 0)
OPEN_REJECT Received (Reserved Initialize 1)	Open Failed (Reserved Initialize 1)
OPEN_REJECT Received (Reserved Stop 0)	Open Failed (Reserved Stop 0)
OPEN_REJECT Received (Reserved Stop 1)	Open Failed (Reserved Stop 1)
OPEN_REJECT Received (Retry)	Open Failed (Retry)

6.15.4.3.2 Transition SL_CC1:ArbSel to SL_CC0:Idle

This transition shall occur after sending an Open Failed confirmation.

6.15.4.3.3 Transition SL_CC1:ArbSel to SL_CC2:Selected

This transition shall occur after receiving a SOAF/Data Dwords/EOAF Transmitted message if:

- one or more AIP Received messages have been received before an OPEN Address Frame Received message is received (i.e., the incoming OPEN address frame overrides the outgoing OPEN address frame); or
- no AIP Received messages have been received before an OPEN Address Frame Received message is received, and the arbitration fairness rules (see 6.13.3) indicate the received OPEN address frame overrides the outgoing OPEN address frame.

The arbitration fairness comparison shall compare:

- the value of the arbitration wait time argument in the Open Connection request for the outgoing OPEN address frame; and
- the value of the ARBITRATION WAIT TIME field received in the incoming OPEN address frame.

6.15.4.3.4 Transition SL_CC1:ArbSel to SL_CC3:Connected

This transition shall occur if this state receives an SOAF/Data Dwords/EOAF Transmitted message followed by an OPEN_ACCEPT Received message.

If the PROTOCOL field in the transmitted OPEN address frame was set to STP, then this state shall send a Connection Opened (STP, Source Opened) confirmation to the port layer before the transition. This transition shall include an Open STP Connection argument. At this point an STP connection has been opened between the source phy and the destination phy.

If the PROTOCOL field in the transmitted OPEN address frame was set to SSP, then this state shall send a Connection Opened (SSP, Source Opened) confirmation to the port layer before the transition. This transition shall include an Open SSP Connection argument. At this point an SSP connection has been opened between the source phy and the destination phy.

If the PROTOCOL field in the transmitted OPEN address frame was set to SMP, then this state shall send a Connection Opened (SMP, Source Opened) confirmation to the port layer before the transition. This transition shall include an Open SMP Connection argument. At this point an SMP connection has been opened between the source phy and the destination phy.

6.15.4.3.5 Transition SL_CC1:ArbSel to SL_CC5:BreakWait

This transition shall occur after receiving a SOAF/Data Dwords/EOAF Transmitted message if a BREAK Received message has not been received and after:

- a) sending an Open Failed (Arb Stopped) confirmation to the port layer;
- b) sending an Open Failed (Open Timeout Occurred) confirmation to the port layer; or
- c) a NOTIFY Received (Power Loss Expected) message is received.

If a NOTIFY Received (Power Loss Expected) message was received and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port), then this transition shall include a Power Loss Expected argument.

6.15.4.3.6 Transition SL_CC1:ArbSel to SL_CC6:Break

This transition shall occur after:

- a) receiving a SOAF/Data Dwords/EOAF Transmitted message;
- b) receiving a BREAK Received message; and
- c) sending an Open Failed (Break Received) confirmation to the port layer.

If a NOTIFY Received (Power Loss Expected) message was received and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port), then this transition shall include a Power Loss Expected argument.

6.15.4.4 SL_CC2:Selected state

6.15.4.4.1 State description

This state completes the establishment of an SSP, SMP, or STP connection when an incoming connection request has won arbitration by sending a Transmit OPEN_ACCEPT message, or rejects opening a connection by sending a Transmit OPEN_REJECT message to the SL transmitter.

This state shall respond to an incoming OPEN address frame using the following rules:

- 1) If the OPEN address frame DESTINATION SAS ADDRESS field does not match the SAS address of this port, then this state shall send a Transmit OPEN_REJECT (Wrong Destination) message to the SL transmitter (see 6.15.4.4.2);
- 2) If the OPEN address frame INITIATOR PORT bit, PROTOCOL field, FEATURES field, and/or INITIATOR CONNECTION TAG field are set to values that are not supported (e.g., a connection request from an SMP target port), then this state shall send a Transmit OPEN_REJECT (Protocol Not Supported) message to the SL transmitter (see 6.15.4.4.2);

- 3) If the OPEN address frame CONNECTION RATE field is set to a connection rate that is not supported, then this state shall send a Transmit OPEN_REJECT (Connection Rate Not Supported) message to the SL transmitter (see 6.15.4.4.2);
- 4) If the OPEN address frame PROTOCOL field is set to STP, the STP target port supports affiliations, and the source SAS address is not that of an STP initiator port with an affiliation established (see 6.18.4), then this state shall send a Transmit OPEN_REJECT (STP Resources Busy) message to the SL transmitter (see 6.15.4.4.2);
- 5) If the OPEN address frame PROTOCOL field is set to SSP and the Reject SSP Opens state machine variable is set to YES, then this state shall send a Transmit OPEN_REJECT (Retry) message to the SL transmitter (see 6.15.4.4.2);
- 6) If the OPEN address frame PROTOCOL field is set to SMP and the Reject SMP Opens state machine variable is set to YES, then this state shall send a Transmit OPEN_REJECT (Retry) message to the SL transmitter (see 6.15.4.4.2);
- 7) If the OPEN address frame PROTOCOL field is set to STP and the Reject STP Opens state machine variable is set to YES, then this state shall send a Transmit OPEN_REJECT (Retry) message to the SL transmitter (see 6.15.4.4.2);
- 8) If the OPEN address frame PROTOCOL field is set to SSP and the Reject SSP Opens state machine variable is set to NO, then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (SSP, Destination Opened) confirmation to the port layer (see 6.15.4.4.3);
- 9) If the OPEN address frame PROTOCOL field is set to SMP and the Reject SMP Opens state machine variable is set to NO, then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (SMP, Destination Opened) confirmation to the port layer (see 6.15.4.4.3); or
- 10) If the OPEN address frame PROTOCOL field is set to STP and the Reject STP Opens state machine variable is set to NO, then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (STP, Destination Opened) confirmation to the port layer (see 6.15.4.4.3).

If this state sends a Transmit OPEN_REJECT message to the SL transmitter, then it shall also send an Inbound Connection Rejected confirmation to the port layer.

NOTE 45 Possible livelock scenarios occur if the BREAK_REPLY method of responding to BREAK primitive sequences is disabled and a SAS logical phy transmits a BREAK primitive sequence to abort a connection request (e.g., if its Open Timeout time expires). SAS logical phys should respond to OPEN Address frames faster than 1 ms to reduce susceptibility to this problem.

6.15.4.4.2 Transition SL_CC2:Selected to SL_CC0:Idle

This transition shall occur after sending a Transmit OPEN_REJECT message to the SL transmitter.

6.15.4.4.3 Transition SL_CC2:Selected to SL_CC3:Connected

This transition shall occur after sending a Connection Opened confirmation to the port layer.

This transition shall include:

- a) an Open SSP Connection, Open STP Connection, or Open SMP Connection argument based on the requested protocol; and
- b) the received OPEN address frame.

6.15.4.4.4 Transition SL_CC2:Selected to SL_CC5:BreakWait

If the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port), then this transition shall occur after receiving a NOTIFY Received (Power Loss Expected) message and shall include a Power Loss Expected argument.

6.15.4.4.5 Transition SL_CC2:Selected to SL_CC6:Break

This transition shall occur after a BREAK Received message is received.

6.15.4.5 SL_CC3:Connected state

6.15.4.5.1 State description

This state enables the SSP, STP, or SMP link layer state machine to transmit dwords during a connection. See 6.14 for details on rate matching during the connection.

If this state is entered from SL_CC1:ArbSel state or the SL_CC2:Selected state with an argument of Open SMP Connection, then this state shall send an Enable Disable SMP (Enable) message to the SMP link layer state machines (see 6.19.5).

If this state is entered from SL_CC1:ArbSel state or the SL_CC2:Selected state with an argument of Open SSP Connection, then this state shall send an Enable Disable SSP (Enable) message to the SSP link layer state machines (see 6.17.8).

If this state is entered from SL_CC1:ArbSel state or the SL_CC2:Selected state with an argument of Open STP Connection, then this state shall send an Enable Disable STP (Enable) message to the STP link layer state machines (see 6.18.8).

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter until the SSP, SMP, or STP link layer state machine starts transmitting.

A CLOSE Received message may be received at any time while in this state, but shall be ignored during SSP and SMP connections. If a CLOSE Received (Clear Affiliation) message is received during an STP connection, then this state shall clear any affiliation (see 6.18.4).

If a Request Break message is received and a BREAK Received message has not been received, then this state shall send a Connection Closed (Break Requested) confirmation to the port layer.

If a BREAK Received message is received, then this state shall send a Connection Closed (Break Received) confirmation to the port layer.

6.15.4.5.2 Transition SL_CC3:Connected to SL_CC4:DisconnectWait

This transition shall occur if a Request Close message is received.

6.15.4.5.3 Transition SL_CC3:Connected to SL_CC5:BreakWait

This transition shall occur after:

- a) sending a Connection Closed (Break Requested) confirmation to the port layer; or
- b) receiving a NOTIFY Received (Power Loss Expected) message, if the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port).

If a NOTIFY Received (Power Loss Expected) message was received and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port), then this state shall include a Power Loss Expected argument.

6.15.4.5.4 Transition SL_CC3:Connected to SL_CC6:Break

This transition shall occur after sending a Connection Closed (Break Received) confirmation to the port layer.

6.15.4.5.5 Transition SL_CC3:Connected to SL_CC7:CloseSTP

This transition shall occur if a CLOSE Received message is received during an STP connection.

6.15.4.6 SL_CC4:DisconnectWait state

6.15.4.6.1 State description

This state closes the connection and releases all resources associated with the connection.

Upon entry into this state, this state shall:

- 1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter (see 6.18.6); and
- 2) initialize and start the Close Timeout timer.

A CLOSE Received message may be received at any time while in this state. If a CLOSE Received (Clear Affiliation) message is received during an STP connection, then this state shall clear any affiliation (see 6.18.4).

NOTE 46 Possible livelock scenarios occur if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled and a SAS logical phy transmits a BREAK primitive sequence to break a connection (e.g., if its Close Timeout timer expires). SAS logical phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

If a CLOSE Received message is received, then this state shall send a Connection Closed (Normal) confirmation to the port layer.

If a BREAK Received message is received, then this state shall send a Connection Closed (Break Received) confirmation to the port layer.

If a BREAK Received message has not been received and no CLOSE Received message is received in response to a Transmit CLOSE message before the Close Timeout timer expires then this state shall send a Connection Closed (Close Timeout) confirmation to the port layer.

6.15.4.6.2 Transition SL_CC4:DisconnectWait to SL_CC0:Idle

This transition shall occur after sending a Connection Closed (Normal) confirmation to the port layer.

6.15.4.6.3 Transition SL_CC4:DisconnectWait to SL_CC5:BreakWait

This transition shall occur after:

- a) receiving a NOTIFY Received (Power Loss Expected) message; or
- b) sending a Connection Closed (Close Timeout) confirmation to the port layer.

If a NOTIFY Received (Power Loss Expected) message was received and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port), then this state shall include a Power Loss Expected argument.

6.15.4.6.4 Transition SL_CC4:DisconnectWait to SL_CC6:Break

This transition shall occur after sending a Connection Closed (Break Received) confirmation to the port layer.

6.15.4.7 SL_CC5:BreakWait state

6.15.4.7.1 State description

This state closes the connection if one is established and releases all resources associated with the connection.

Upon entry into this state, this state shall:

- 1) send a Transmit BREAK message to the SL transmitter; and
- 2) initialize and start the Break Timeout timer.

If this state:

- a) is entered with a Power Loss Expected argument; or
- b) receives a NOTIFY Received (Power Loss Expected) message and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port),

then this state shall send a Notify Received (Power Loss Expected) confirmation to the port layer.

If a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the SL transmitter.

NOTE 47 Some SAS logical phys compliant with SAS-1.1 send a Transmit OPEN_REJECT (Retry) message to the SL transmitter in response to each OPEN Address Frame Received message received while in this state.

6.15.4.7.2 Transition SL_CC5:BreakWait to SL_CC0:Idle

This transition shall occur after:

- a) receiving a BREAK_REPLY Received message if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5);
- b) receiving a BREAK Received message if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled (see 6.13.5); or
- c) the Break Timeout timer expires.

6.15.4.8 SL_CC6:Break state

6.15.4.8.1 State description

This state closes any connection and releases all resources associated with this connection.

Upon entry into this state:

- a) if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the SL transmitter (see 6.15.4.8.2); and
- b) if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled (see 6.13.5), then this state shall send a Transmit BREAK message to the SL transmitter.

If this state:

- a) is entered with a Power Loss Expected argument; or
- b) receives a NOTIFY Received (Power Loss Expected) message and the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port),

then this state shall send a Notify Received (Power Loss Expected) confirmation to the port layer.

6.15.4.8.2 Transition SL_CC6:Break to SL_CC0:Idle

This transition shall occur after sending a Transmit BREAK message or a Transmit BREAK_REPLY message to the SL transmitter.

6.15.4.9 SL_CC7:CloseSTP state

6.15.4.9.1 State description

This state closes an STP connection and releases all resources associated with the connection.

Upon entry into this state, this state shall:

- 1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter (see 6.18.6); and

- 2) send a Connection Closed (Normal) confirmation to the port layer (see 6.15.4.9.2).

NOTE 48 Possible livelock scenarios occur if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled and a SAS logical phy transmits a BREAK primitive sequence to break a connection (e.g., if its Close Timeout timer expires). SAS logical phys should respond to a CLOSE primitive sequence faster than 1 ms to reduce susceptibility to this problem.

6.15.4.9.2 Transition SL_CC7:CloseSTP to SL_CC0:Idle

This transition shall occur after sending a Connection Closed (Normal) confirmation to the port layer.

6.15.4.10 SL_CC8:PS_Request state

6.15.4.10.1 State description

This state requests the attached phy change to a low phy power condition.

Upon entry into this state, this state shall:

- a) if entered with the Phy Power Condition (Enter Partial) argument, then send a Transmit PS_REQ (Partial) message to the SL transmitter;
- b) if entered with the Phy Power Condition (Enter Slumber) argument, then send a Transmit PS_REQ (Slumber) message to the SL transmitter; and
- c) initialize and start the Power Condition Request Timeout timer.

If this state:

- a) is entered with the Phy Power Condition (Enter Partial) argument; and
- b) receives a PS_REQ Received (Slumber) message,

then this state shall set the Phy Power Condition argument to Enter Partial.

If this state:

- a) is entered with the Phy Power Condition (Enter Slumber) argument; and
- b) receives a PS_REQ Received (Partial) message,

then this state shall:

- a) set the Phy Power Condition argument to Enter Partial; and
- b) if the Accept Partial state machine variable is set to:
 - A) YES, then send a Transmit PS_ACK Pattern message to the SL transmitter; or
 - B) NO, then send a Transmit PS_NAK message to the SL transmitter.

If this state:

- a) receives a PS_REQ Received message;
- b) the PS_REQ Received message argument is requesting the same power condition as the Phy Power Condition argument; and
- c) the attached SAS address is greater than the port identifier (i.e., SAS address),

then this state shall:

- a) send a Transmit PS_ACK Pattern message to the SL transmitter; and
- b) send a Manage Phy Power Conditions (Stop DWS) request to the phy layer.

If a Change Phy Power Condition request is received, then this state shall send a Phy Power Condition Status (Retry Change Phy Power Condition Request) confirmation to the management application layer.

If an Open Connection request is received, then this state shall send an Open Failed (Low Phy Power Condition) confirmation to the port layer.

If a BROADCAST Received (Change) message, BROADCAST Received (Reserved Change 0) message, or BROADCAST Received (Reserved Change 1) message is received, then this state shall send a Change Received confirmation to the management application layer.

If a Transmit Broadcast request is received, then this state shall send a Retry Transmit Broadcast confirmation to the management application layer.

If a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the SL transmitter.

6.15.4.10.2 Transition SL_CC8:PS_Request to SL_CC9:PS_Quiet

This transition shall occur after:

- a) receiving a PS_ACK Pattern Transmitted message from the SL transmitter; or
- b) receiving a PS_ACK Received message.

This transition shall include the Phy Power Condition argument.

6.15.4.10.3 Transition SL_CC8:PS_Request to SL_CC0:Idle

This transition shall occur after:

- a) sending a Transmit PS_NAK message to the SL transmitter;
- b) receiving a PS_NAK Received message;
- c) sending a Transmit BREAK_REPLY message to the SL transmitter;
- d) a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled; or
- e) the Power Condition Request Timeout timer expires.

6.15.4.10.4 Transition SL_CC8:PS_Request to SL_CC2:Selected

This transition shall occur after receiving an OPEN Address Frame Received message.

6.15.4.11 SL_CC9:PS_Quiet state

6.15.4.11.1 State description

This state requests the phy be placed into a low phy power condition.

If this state is entered with a Phy Power Condition (Enter Partial) argument, then this state shall send:

- a) a Manage Phy Power Conditions (Enter Partial) request to the phy layer; and
- b) a Phy Power Condition Status (In Partial) confirmation to the management application layer.

If this state is entered with Phy Power Condition (Enter Slumber) argument, then this state shall send:

- a) a Manage Phy Power Conditions (Enter Slumber) request to the phy layer; and
- b) a Phy Power Condition Status (In Slumber) confirmation to the management application layer.

If this state receives a:

- a) Change Phy Power Condition (Enter Slumber) request and the current phy power condition is partial;
or
- b) Change Phy Power Condition (Enter Partial) request and the current phy power condition is slumber,

then this state shall send a Phy Power Condition Status (Request Exit Power Condition) confirmation to the management application layer.

If a Transmit Broadcast request is received, then this state shall send a Retry Transmit Broadcast confirmation to the management application layer.

If this state receives an Open Connection request or a Change Phy Power Condition (Exit Power Condition) request, then this state shall send a Manage Phy Power Conditions (Exit) request to the phy layer.

6.15.4.11.2 Transition SL_CC9:PS_Quiet to SL_CC0:Idle

This transition shall occur:

- a) after a Phy Layer Ready (SAS) confirmation is received; and
- b) an Open Connection request has not been received.

6.15.4.11.3 Transition SL_CC9:PS_Quiet to SL_CC1:ArbSel

This transition shall occur after receiving both a Phy Layer Ready (SAS) confirmation and an Open Connection request. The Open Connection request includes these arguments:

- a) Initiator Port Bit;
- b) Protocol;
- c) Connection Rate;
- d) Initiator Connection Tag;
- e) Destination SAS Address;
- f) Source SAS Address;
- g) Pathway Blocked Count; and
- h) Arbitration Wait Time.

6.16 XL (link layer for expander logical phys) state machine

6.16.1 XL state machine overview

The XL state machine controls the flow of dwords on the logical link and establishes and maintains connections with another XL state machine as facilitated by the expander function (e.g., the ECM and ECR).

This state machine consists of the following states:

- a) XL0:Idle (see 6.16.3) (initial state);
- b) XL1:Request_Path (see 6.16.4);
- c) XL2:Request_Open (see 6.16.5);
- d) XL3:Open_Confirm_Wait (see 6.16.6);
- e) XL4:Open_Reject (see 6.16.7);
- f) XL5:Forward_Open (see 6.16.8);
- g) XL6:Open_Response_Wait (see 6.16.9);
- h) XL7:Connected (see 6.16.10);
- i) XL8:Close_Wait (see 6.16.11);
- j) XL9:Break (see 6.16.12);
- k) XL10:Break_Wait (see 6.16.13);
- l) XL11:PS_Request (see 6.16.14); and
- m) XL12:PS_Quiet (see 6.16.15).

This state machine shall start in the XL0:Idle state. The XL state machine shall transition to the XL0:Idle state from any other state after receiving an Enable Disable SAS Link (Disable) message from the SL_IR state machines (see 6.10).

This state machine receives the following messages from the SL_IR state machines:

- a) Enable Disable SAS Link (Enable); and
- b) Enable Disable SAS Link (Disable).

Any message received by a state that is not referred to in the description of that state shall be ignored.

If this state machine receives a Manage Power Conditions (Accept Partial) request from the ECM, then this state machine shall set the Accept Partial state machine variable (see table 107) to YES. If this state machine receives a Manage Power Conditions (Accept Slumber) request from the ECM, then this state machine shall set the Accept Slumber state machine variable (see table 107) to YES.

If this state machine receives a Manage Power Conditions (Reject Partial) request from the ECM, then this state machine shall set the Accept Partial state machine variable to NO. If this state machine receives a Manage Power Conditions (Reject Slumber) request from the ECM, then this state machine shall set the Accept Slumber state machine variable to NO.

The default value of the Accept Partial state machine variable and Accept Slumber state machine variable shall be NO.

This state machine shall maintain the timers listed in table 106.

Table 106 — XL state machine timers

Timer	Initial value
Partial Pathway Timeout timer	Partial pathway timeout value (see 6.13.4.4)
Break Timeout timer	1 ms
Power Condition Request Timeout timer	1 ms

The XL state machine shall maintain the state machine variables listed in table 107.

Table 107 — XL state machine variable

State machine variable	Description
Accept Partial	Used to determine if the ECM is permitting this phy to enter a partial phy power condition.
Accept Slumber	Used to determine if the ECM is permitting this phy to enter a slumber phy power condition.

Figure 114 shows several states in the XL state machine.

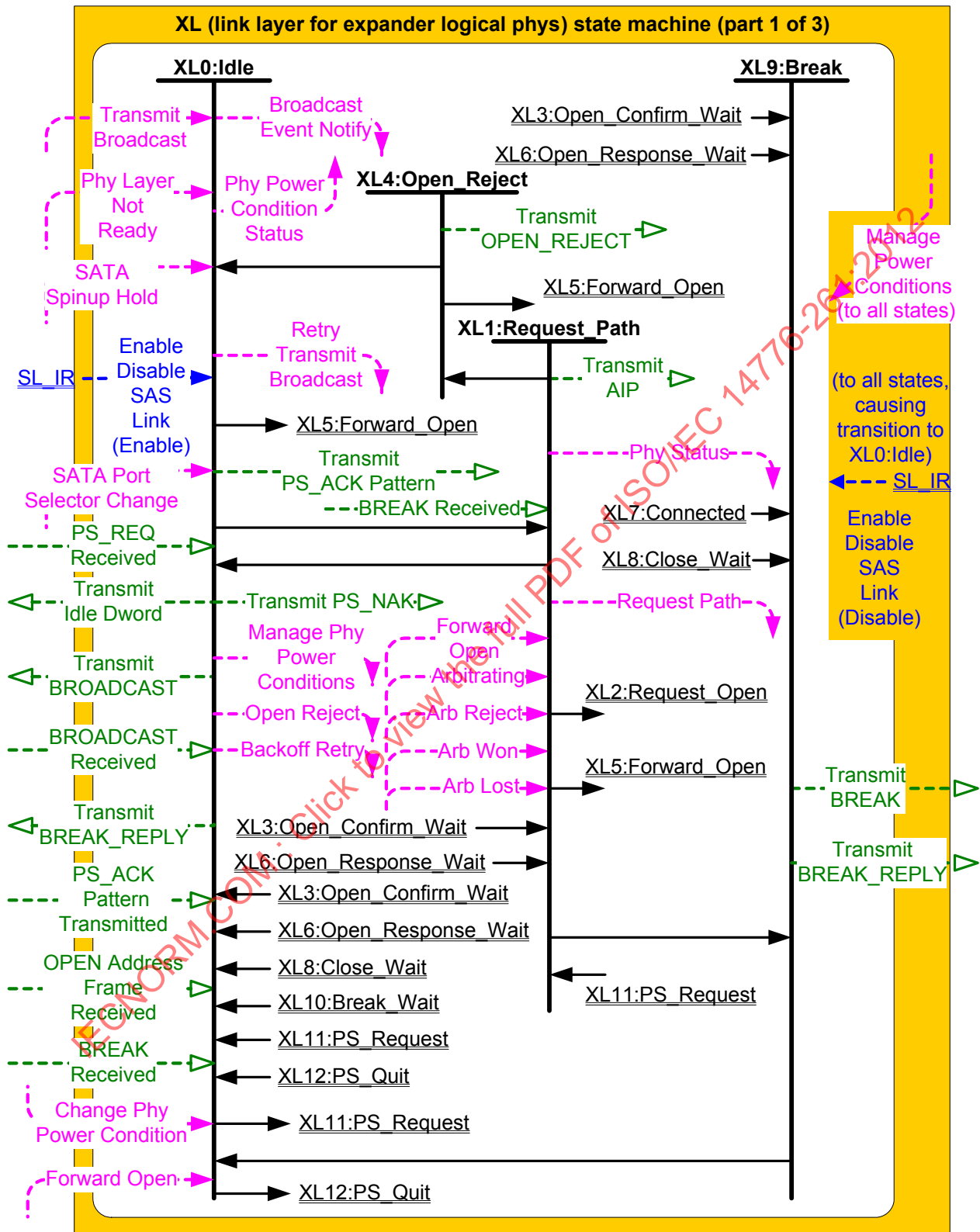


Figure 114 — XL (link layer for expander logical phys) state machine (part 1)

XL (link layer for expander logical phys) state machine (part 2 of 3)

The diagram illustrates the state machine for the XL (link layer for expander logical phys) state machine (part 2 of 3). The states and transitions are as follows:

- States:** XL0:Idle, XL1:Request_Path, XL2:Request_Open, XL3:Open_Confirm_Wait, XL5:Forward_Open, XL6:Open_Response_Wait, XL7:Connected, XL9:Break, XL10:Break_Wait.
- Transitions:**
 - XL0:Idle to XL1:Request_Path:** Triggered by **BREAK Received**.
 - XL1:Request_Path to XL2:Request_Open:** Triggered by **PS_REQ Received**.
 - XL2:Request_Open to XL3:Open_Confirm_Wait:** Triggered by **OPEN Address Frame Transmitted**.
 - XL3:Open_Confirm_Wait to XL5:Forward_Open:** Triggered by **OPEN Address Frame Received**.
 - XL5:Forward_Open to XL6:Open_Response_Wait:** Triggered by **BROADCAST Received**.
 - XL6:Open_Response_Wait to XL7:Connected:** Triggered by **AIP Received**.
 - XL7:Connected to XL9:Break:** Triggered by **BREAK Received**.
 - XL9:Break to XL10:Break_Wait:** Triggered by **BROADCAST Received**.
 - XL10:Break_Wait to XL0:Idle:** Triggered by **BROADCAST Received**.
- Actions:**
 - XL2:Request_Open:** Transmit Idle Dword.
 - XL3:Open_Confirm_Wait:** Transmit AIP, Transmit OPEN_ACCEPT, Transmit OPEN_REJECT, Transmit Idle Dword.
 - XL5:Forward_Open:** Transmit PS_NAK.
 - XL6:Open_Response_Wait:** Transmit OPEN Address Frame, Transmit PS_NAK, Transmit Idle Dword, Transmit Dword.

Additional notes from the diagram:

- Forward Break:** Indicated by a dashed line from XL3:Open_Confirm_Wait to XL5:Forward_Open.
- Backoff Reverse Path:** Indicated by a dashed line from XL5:Forward_Open to XL3:Open_Confirm_Wait.
- Forward Dword:** Indicated by a dashed line from XL5:Forward_Open to XL3:Open_Confirm_Wait.
- PS_REQ Received:** Indicated by a dashed line from XL1:Request_Path to XL2:Request_Open.
- OPEN Address Frame Received:** Indicated by a dashed line from XL3:Open_Confirm_Wait to XL5:Forward_Open.
- BROADCAST Received:** Indicated by a dashed line from XL6:Open_Response_Wait to XL7:Connected.
- AIP Received:** Indicated by a dashed line from XL7:Connected to XL9:Break.
- BREAK Received:** Indicated by a dashed line from XL9:Break to XL10:Break_Wait.
- BROADCAST Received:** Indicated by a dashed line from XL10:Break_Wait to XL0:Idle.

Manage Power Conditions (to all states)

(to all states, causing transition to XL0:Idle)

SL_IR

Enable Disable SAS Link (Disable)

Figure 115 — XL (link layer for expander logical phys) state machine (part 2)

Figure 116 shows additional states in the XL state machine.

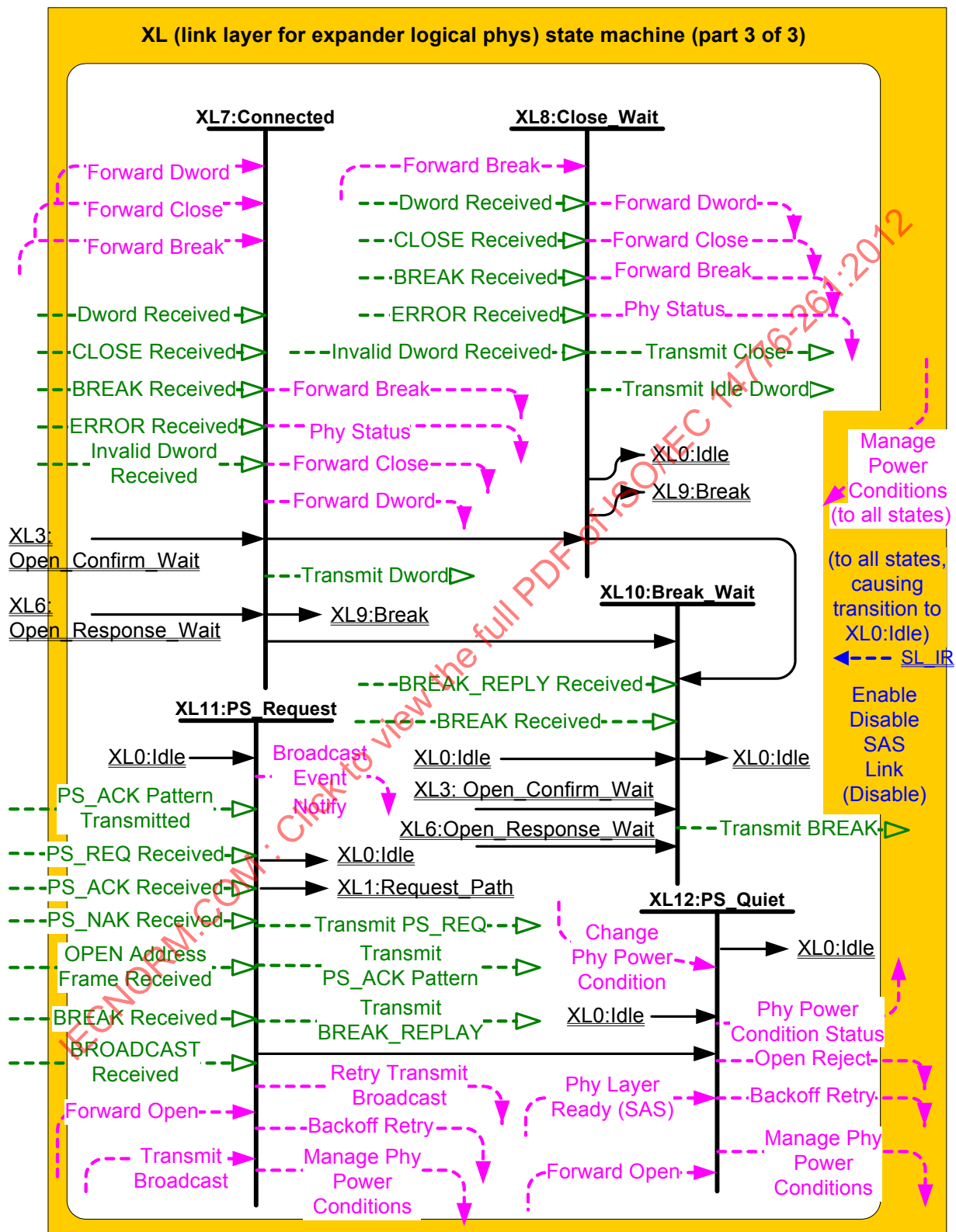


Figure 116 — XL (link layer for expander logical phys) state machine (part 3)

6.16.2 XL transmitter and receiver

The XL transmitter receives the following messages from the XL state machine specifying primitive sequences, frames, and dwords to transmit:

- a) Transmit Idle Dword;
- b) Transmit AIP with an argument indicating the specific type (e.g., Transmit AIP (Normal));
- c) Transmit BREAK;
- d) Transmit BREAK_REPLY;
- e) Transmit BROADCAST with an argument indicating the specific type (e.g., Transmit BROADCAST (Change));
- f) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal));
- g) Transmit OPEN_ACCEPT;
- h) Transmit OPEN_REJECT, with an argument indicating the specific type (e.g., Transmit OPEN_REJECT (No Destination));
- i) Transmit PS_REQ with an argument indicating the specific type (e.g., Transmit PS_REQ (Partial) or Transmit PS_REQ (Slumber));
- j) Transmit PS_ACK Pattern (see 6.11);
- k) Transmit PS_NAK;
- l) Transmit OPEN Address Frame; and
- m) Transmit Dword.

The XL transmitter sends the following message to the XL state machine based on dwords that have been transmitted:

- a) OPEN Address Frame Transmitted and
- b) PS_ACK Pattern Transmitted.

The XL transmitter shall ensure physical link rate tolerance management requirements are met (see 6.3) while originating dwords.

The XL transmitter shall ensure physical link rate tolerance management requirements are met while forwarding dwords (i.e., during a connection) by inserting or deleting as many deletable primitives as required to match the transmit and receive connection rates (see 6.3.2).

The XL transmitter shall ensure physical link rate tolerance management requirements are met (see 6.3) during and after switching from forwarding dwords to originating dwords, including, for example:

- a) when transmitting a BREAK primitive sequence;
- b) when transmitting a BREAK_REPLY primitive sequence;
- c) when transmitting a CLOSE primitive sequence;
- d) when transmitting an idle dword after closing a connection (i.e., after receiving a BREAK primitive sequence, BREAK_REPLY primitive sequence, or CLOSE primitive sequence);
- e) while transmitting a SATA frame to a SAS logical link during an STP connection, when transmitting the first SATA_HOLD in response to detection of SATA_HOLD; and
- f) while receiving dwords of a SATA frame from a SAS logical link during an STP connection, when transmitting SATA_HOLD.

NOTE 49 The XL transmitter may always insert a deletable primitive before transmitting a BREAK, BREAK_REPLY, CLOSE, or SATA_HOLD to meet physical link rate tolerance management requirements.

The XL transmitter shall insert a deletable primitive before switching from originating dwords to forwarding dwords, including, for example:

- a) when transmitting OPEN_ACCEPT;
- b) when transmitting the last idle dword before a connection is established (i.e., after receiving OPEN_ACCEPT);
- c) while transmitting a SATA frame to a SAS logical link during an STP connection, when transmitting the last dword from the STP flow control buffer in response to release of SATA_HOLD;
- d) while transmitting a SATA frame to a SAS logical link during an STP connection, when transmitting the last SATA_HOLD in response to release of SATA_HOLD (e.g., if the STP flow control buffer is empty); and

- e) while receiving dwords of a SATA frame from a SAS logical link during an STP connection, when transmitting the last SATA_HOLD.

NOTE 50 This ensures that physical link rate tolerance management requirements are met, even if the forwarded dword stream does not include a deletable primitive until the last possible dword.

The XL transmitter shall ensure rate matching requirements are met during a connection (see 6.14).

When there is no outstanding message specifying a dword to transmit, the XL transmitter shall transmit idle dwords.

The XL receiver sends the following messages to the XL state machine indicating primitive sequences, frames, and dwords received from the SP_DWS receiver (see 5.10.2):

- a) AIP Received with an argument indicating the specific type (e.g., AIP Received (Normal));
- b) BREAK Received;
- c) BREAK_REPLY Received;
- d) BROADCAST Received;
- e) CLOSE Received;
- f) OPEN_ACCEPT Received;
- g) OPEN_REJECT Received;
- h) OPEN Address Frame Received;
- i) PS_REQ Received with an argument indicated the specific type (e.g., PS_REQ Received (Partial)) or PS_REQ Received (Slumber));
- j) PS_ACK Received;
- k) PS_NAK Received;
- l) Dword Received with an argument indicating the data dword or primitive received. Deletable primitives are not included; and
- m) Invalid Dword Received.

The XL receiver shall ignore all other dwords.

While receiving an address frame, if the XL receiver receives an invalid dword or ERROR, then the XL receiver shall either:

- a) ignore the invalid dword or ERROR; or
- b) discard the address frame.

The XL transmitter relationship to other transmitters is defined in 4.3.2. The XL receiver relationship to other receivers is defined in 4.3.3.

6.16.3 XL0:Idle state

6.16.3.1 State description

This state is the initial state and is the state that is used when there is no connection pending or established.

This state shall repeatedly send Idle requests to the ECM.

If a Phy Layer Not Ready confirmation is received, then this state shall send a Broadcast Event Notify (Phy Not Ready) request to the BPP.

If a SATA Spinup Hold confirmation is received, then this state shall send a Broadcast Event Notify (SATA Spinup Hold) request to the BPP.

If an Enable Disable SAS Link (Enable) message is received, then this state shall send a Broadcast Event Notify (Identification Sequence Complete) request to the BPP.

If a SATA Port Selector Change confirmation is received, then this state shall send a Broadcast Event Notify (SATA Port Selector Change) request to the BPP.

If a BROADCAST Received message is received, then this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive sequence received (e.g., Change Received).

If a Transmit Broadcast indication is received and this state has not sent a Transmit PS_ACK Pattern message, then this state shall send a Transmit BROADCAST message to the XL transmitter with an argument specifying the specific type from the Transmit Broadcast indication. Otherwise, this state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter. If a Transmit Broadcast indication is received and this state has sent a Transmit PS_ACK Pattern message to the XL transmitter, then this state shall send a Retry Transmit Broadcast request to the BPP.

If a Transmit Broadcast indication is received coincident with receiving either a Forward Open indication or an OPEN Address Frame Received message, then this state shall send a Transmit BROADCAST message with the same argument to the XL transmitter before transitioning to another state.

If a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the XL transmitter.

If a PS_REQ Received (Partial) message is received and the Accept Partial state machine variable is set to YES, then this state shall:

- a) send a Transmit PS_ACK Pattern message to the XL transmitter; and
- b) send a Manage Phy Power Conditions (Stop DWS) request to the phy layer.

If a PS_REQ Received (Slumber) message is received and the Accept Slumber state machine variable is set to YES, then this state shall:

- a) send a Transmit PS_ACK Pattern message to the XL transmitter; and
- b) send a Manage Phy Power Conditions (Stop DWS) request to the phy layer.

If a PS_REQ Received (Partial) message is received and the Accept Partial state machine variable is set to NO, then this state shall send a Transmit PS_NAK message to the XL transmitter. If a PS_REQ Received (Slumber) message is received and the Accept Slumber state machine variable is set to NO, then this state shall send a Transmit PS_NAK message to the XL transmitter.

If a Forward Open indication is received from the ECM and this state has sent a Transmit PS_ACK Pattern message to the XL transmitter, then this state shall send a Backoff Retry response to the ECM.

If a Change Phy Power Condition is received and this state has sent a Transmit PS_ACK Pattern message to the XL transmitter, then this state shall send a Phy Power Condition Status (Retry Change Phy Power Condition Request) confirmation to the ECM.

This state shall ignore any Change Phy Power Condition (Exit Power Condition) requests.

6.16.3.2 Transition XL0:Idle to XL1:Request_Path

This transition shall occur if:

- a) an Enable Disable SAS Link (Enable) message has been received;
- b) a Forward Open indication is not being received; and
- c) an OPEN Address Frame Received message is received.

This state shall include an OPEN Address Frame Received argument with the transition.

6.16.3.3 Transition XL0:Idle to XL5:Forward_Open

6.16.3.3.1 General

If this state has not sent a Transmit PS_ACK Pattern message to the XL transmitter, then this transition shall occur after receiving:

- a) an Enable Disable SAS Link (Enable) message; and
- b) a Forward Open indication.

This transition shall include a set of arguments containing the arguments received in the Forward Open indication.

If an OPEN Address Frame Received message is received, then this state shall include an OPEN Address Frame Received argument with the transition.

6.16.3.3.2 Transition XL0:Idle to XL11:PS_Request

If this state has not sent a Transmit PS_ACK Pattern message to the XL transmitter, then this transition shall occur after receiving:

- a) an Enable Disable SAS Link (Enable) message; and
- b) a Change Phy Power Condition request from the ECM.

This transition shall include the Phy Power Condition argument (Enter Partial) or Phy Power Condition (Enter Slumber) argument that corresponds to the Change Phy Power Condition request.

6.16.3.3.3 Transition XL0:Idle to XL12:PS_Quiet

This transition shall occur after receiving a PS_ACK Pattern Transmitted message from the XL transmitter.

This transition shall include the Phy Power Condition (Enter Partial) argument or Phy Power Condition (Enter Slumber) argument that corresponds to the PS_REQ Received message.

6.16.4 XL1:Request_Path state

6.16.4.1 State description

This state is used to arbitrate for connection resources and to specify the destination of the connection.

If an Arbitrating (Normal) confirmation is received, then this state shall repeatedly send Transmit AIP (Normal) messages and Transmit Idle Dword messages to the XL transmitter in accordance with AIP transmission rules (see 6.13.4.3).

If an Arbitrating (Waiting On Partial) confirmation or an Arbitrating (Blocked On Partial) confirmation is received, then this state shall repeatedly send Transmit AIP (Waiting On Partial) messages and Transmit Idle Dword messages to the XL transmitter in accordance with AIP transmission rules (see 6.13.4.3).

If an Arbitrating (Waiting On Partial) confirmation is received, then this state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM.

If an Arbitrating (Blocked On Partial) confirmation is received, then this state shall repeatedly send a Phy Status (Blocked Partial Pathway) response to the ECM.

If an Arbitrating (Waiting On Connection) confirmation is received, then this state shall repeatedly send Transmit AIP (Waiting On Connection) messages and Transmit Idle Dword messages to the XL transmitter in accordance with AIP transmission rules (see 6.13.4.3).

If an Arbitrating (Waiting On Connection) confirmation is received, then this state shall repeatedly send a Phy Status (Connection) response to the ECM.

If this state is entered from the XL6:Open_Response_Wait state, then the High Priority argument shall be set to one. If this state is entered from any other state, then the High Priority argument shall be set to zero.

Upon entry into this state, this state shall send a Request Path request to the ECM with the following arguments:

- a) Initiator Port Bit;
- b) Protocol;
- c) Connection Rate;
- d) Initiator Connection Tag;
- e) Destination SAS Address;
- f) Source SAS Address;
- g) Pathway Blocked Count;
- h) Arbitration Wait Time; and
- i) High Priority.

This state maintains the Partial Pathway Timeout timer.

If the Partial Pathway Timeout timer is not already running, then the Partial Pathway Timeout timer shall be initialized and started when an Arbitrating (Blocked On Partial) confirmation is received.

If the Partial Pathway Timeout timer is already running, then the Partial Pathway Timeout timer shall continue to run if an Arbitrating (Blocked On Partial) confirmation is received.

The Partial Pathway Timeout timer shall be stopped when one of the following confirmations is received:

- a) Arbitrating (Waiting On Partial); or
- b) Arbitrating (Waiting On Connection).

If the Partial Pathway Timeout timer expires, then this state shall send a Partial Pathway Timeout Timer Expired request to the ECM.

6.16.4.2 Transition XL1:Request_Path to XL0:Idle

This transition shall occur if:

- a) a BREAK Received message has not been received; and
- b) an Arb Lost confirmation is received.

6.16.4.3 Transition XL1:Request_Path to XL2:Request_Open

This transition shall occur if:

- a) a BREAK Received message has not been received; and
- b) an Arb Won confirmation is received.

6.16.4.4 Transition XL1:Request_Path to XL4:Open_Reject

This transition shall occur if:

- a) a BREAK Received message has not been received; and
- b) an Arb Reject confirmation is received.

This transition shall include an Arb Reject argument corresponding to the Arb Reject confirmation.

6.16.4.5 Transition XL1:Request_Path to XL5:Forward_Open

This transition shall occur if a Forward Open indication is received and none of the following confirmations have been received:

- a) Arbitrating (Normal);
- b) Arbitrating (Waiting On Partial);
- c) Arbitrating (Blocked On Partial);
- d) Arbitrating (Waiting On Connection);
- e) Arb Won;
- f) Arb Lost;
- g) Arb Reject (No Destination);
- h) Arb Reject (Bad Destination);
- i) Arb Reject (Connection Rate Not Supported);
- j) Arb Reject (Zone Violation);
- k) Arb Reject (Pathway Blocked); or
- l) Arb Reject (Retry).

This transition shall include:

- a) an OPEN Address Frame Received argument containing the arguments received in the Forward Open indication; and
- b) a BREAK Received argument if a BREAK Received message was received.

6.16.4.6 Transition XL1:Request_Path to XL9:Break

This transition shall occur after receiving a BREAK Received message if a Forward Open indication has not been received.

6.16.5 XL2:Request_Open state

6.16.5.1 State description

This state is used to forward an OPEN address frame through the ECR to a destination phy.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

Upon entry into this state, this state shall send a Forward Open request to the ECR, received by the destination phy as a Forward Open indication (see 6.16.5.2). The arguments to the Forward Open request are:

- a) initiator port bit;
- b) protocol;
- c) features;
- d) connection rate;
- e) initiator connection tag;
- f) Destination SAS Address;
- g) source SAS address;
- h) source zone group;
- i) pathway blocked count;
- j) arbitration wait time; and
- k) more compatible features.

6.16.5.2 Transition XL2:Request_Open to XL3:Open_Confirm_Wait

This transition shall occur after sending a Forward Open request to the ECR.

If a BREAK Received message is received, then this state shall include a BREAK Received argument with the transition.

6.16.6 XL3:Open_Confirm_Wait state

6.16.6.1 State description

This state waits for confirmation for an OPEN address frame sent on a destination phy.

This state shall send the following messages to the XL transmitter:

- a) a Transmit AIP (Normal) message when an Arb Status (Normal) confirmation is received;
- b) a Transmit AIP (Waiting On Partial) message when an Arb Status (Waiting On Partial) confirmation is received;
- c) a Transmit AIP (Waiting On Connection) message when an Arb Status (Waiting On Connection) confirmation is received;
- d) a Transmit AIP (Waiting On Device) message when an Arb Status (Waiting On Device) confirmation is received;
- e) a Transmit OPEN_ACCEPT message when an Open Accept confirmation is received (see 6.16.6.5);
- f) a Transmit OPEN_REJECT message when an Open Reject confirmation is received with the argument from the Open Reject confirmation, after releasing path resources (see 6.16.6.2); or
- g) request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages when none of the previous conditions are present.

If a Backoff Retry confirmation is received, then this state shall release path resources.

If a BREAK Received message is received or a BREAK Received argument is included in the transition into this state, then this state shall send a Forward Break request to the ECR (see 6.16.6.6).

This state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM until an Arb Status (Waiting On Partial) confirmation is received. After an Arb Status (Waiting on Partial) confirmation is received, this state shall repeatedly send a Phy Status (Blocked Partial Pathway) response to the ECM.

If a Dword Received message is received containing a valid dword except a BREAK, then this state shall send a Forward Dword request to the ECR containing that dword.

If:

- a) an Invalid Dword Received message is received; and
- b) the expander logical phy is forwarding to an expander logical phy attached to a SAS logical link,

then the expander logical phy shall:

- a) send an ERROR with the Forward Dword request instead of the invalid dword; or
- b) delete the invalid dword.

If:

- a) an ERROR is received with the Dword Received message or an Invalid Dword Received message is received; and
- b) the expander logical phy is forwarding to an expander phy attached to a SATA physical link,

then the expander logical phy shall:

- a) send a SATA_ERROR with the Forward Dword request instead of the invalid dword or ERROR primitive; or
- b) delete the ERROR or invalid dword.

6.16.6.2 Transition XL3:Open_Confirm_Wait to XL0:Idle

This transition shall occur after sending a Transmit OPEN_REJECT message to the XL transmitter if:

- a) a BREAK Received message has not been received; and
- b) a BREAK Received argument was not included in the transition into this state.

6.16.6.3 Transition XL3:Open_Confirm_Wait to XL1:Request_Path

This transition shall occur after receiving a Backoff Retry confirmation, after releasing path resources if:

- a) a BREAK Received message has not been received; and
- b) a BREAK Received argument was not included in the transition into this state.

6.16.6.4 Transition XL3:Open_Confirm_Wait to XL5:Forward_Open

This transition shall occur after receiving a Backoff Reverse Path confirmation if:

- a) a BREAK Received message has not been received; and
- b) a BREAK Received argument was not included in the transition into this state.

The transition shall include the Backoff Reverse Path arguments (i.e., the OPEN address frame).

6.16.6.5 Transition XL3:Open_Confirm_Wait to XL7:Connected

This transition shall occur after sending a Transmit OPEN_ACCEPT message to the XL transmitter if:

- a) a BREAK Received message has not been received; and
- b) a BREAK Received argument was not included in the transition into this state.

6.16.6.6 Transition XL3:Open_Confirm_Wait to XL9:Break

This transition shall occur after sending a Forward Break request to the ECR.

6.16.6.7 Transition XL3:Open_Confirm_Wait to XL10:Break_Wait

This transition shall occur after receiving a Forward Break indication if:

- a) a BREAK Received message has not been received; and
- b) a BREAK Received argument was not included in the transition into this state.

6.16.7 XL4:Open_Reject state

6.16.7.1 State description

This state is used to reject a connection request.

This state shall send one of the following messages to the XL transmitter (see 6.16.7.2):

- a) a Transmit OPEN_REJECT (No Destination) message when an Arb Reject (No Destination) argument is received with the transition into this state;
- b) a Transmit OPEN_REJECT (Bad Destination) message when an Arb Reject (Bad Destination) argument is received with the transition into this state;
- c) a Transmit OPEN_REJECT (Connection Rate Not Supported) message when an Arb Reject (Connection Rate Not Supported) argument is received with the transition into this state;
- d) a Transmit OPEN_REJECT (Zone Violation) message when an Arb Reject (Zone Violation) argument is received with the transition into this state;
- e) a Transmit OPEN_REJECT (Pathway Blocked) message when an Arb Reject (Pathway Blocked) argument is received with the transition into this state; or
- f) a Transmit OPEN_REJECT (Retry) message when an Arb Reject (Retry) argument is received with the transition into this state.

6.16.7.2 Transition XL4:Open_Reject to XL0:Idle

This transition shall occur after sending a Transmit OPEN_REJECT message to the XL transmitter.

6.16.7.3 Transition XL4:Open_Reject to XL5:Forward_Open

This transition shall occur if a Forward Open indication is received. This transition shall include an OPEN Address Frame Received argument containing the arguments received in the Forward Open indication.

6.16.8 XL5:Forward_Open state

6.16.8.1 State description

This state is used to transmit an OPEN address frame passed with the transition into this state.

If a BROADCAST Received message is received, then this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive sequence received (e.g., Change Received).

Upon entry into this state, this state shall send a Transmit OPEN Address Frame message to the XL transmitter with the fields set to the values specified with the transition into this state.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

If a PS_REQ Received message is received this state shall send a Transmit PS_NAK to the XL transmitter.

6.16.8.2 Transition XL5:Forward_Open to XL6:Open_Response_Wait

This transition shall occur after receiving an OPEN Address Frame Transmitted message.

If an OPEN Address Frame Received message or argument is received, then this state shall include an OPEN Address Frame Received argument with the transition.

If a BREAK Received message or argument is received, then this state shall include a BREAK Received argument with the transition.

6.16.9 XL6:Open_Response_Wait state

6.16.9.1 State description

This state waits for a response to a transmitted OPEN address frame and determines the appropriate action to take based on the response.

This state shall either:

- a) request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter, honoring ALIGN insertion rules for rate matching and physical link rate tolerance management; or
- b) send Transmit Dword messages to the XL transmitter to transmit all dwords received with Forward Dword indications.

If a BROADCAST Received message is received before an AIP Received message is received, then this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive sequence received (e.g., Broadcast Event Notify (Change Received)).

This state shall send the following responses to the ECR, which are received by the source phy as confirmations:

- a) an Open Accept response when an OPEN_ACCEPT Received message is received (see 6.16.9.5);
- b) an Open Reject response when an OPEN_REJECT Received message is received, after releasing any path resources (see 6.16.9.2);
- c) a Backoff Retry response, after releasing path resources (see 6.16.9.3), when:
 - A) an AIP Received message has not been received;
 - B) an OPEN Address Frame Received message is received or an OPEN Address Frame Received argument is included in the transition into this state containing a higher priority OPEN address frame according to the arbitration fairness comparison (see 6.13.3); and
 - C) the destination SAS address and connection rate of the received OPEN address frame are not equal to the source SAS address and connection rate of the transmitted OPEN address frame;
- d) a Backoff Retry response, after releasing path resources (see 6.16.9.3), when:
 - A) an AIP Received message has been received;
 - B) an OPEN Address Frame Received message is received or an OPEN Address Frame Received argument is included in the transition into this state; and
 - C) the destination SAS address and connection rate of the received OPEN address frame are not equal to the source SAS address and connection rate of the transmitted OPEN address frame;
- e) a Backoff Reverse Path response (see 6.16.9.4) when:
 - A) an AIP Received message has not been received,
 - B) an OPEN Address Frame Received message is received or an OPEN Address Frame Received argument is included in the transition into this state containing a higher priority OPEN address frame according to the arbitration fairness comparison (see 6.13.3); and
 - C) the destination SAS address and connection rate of the received OPEN address frame are equal to the source SAS address and connection rate of the transmitted OPEN address frame;

and
- f) a Backoff Reverse Path response (see 6.16.9.4) when:
 - A) an AIP Received message has been received;
 - B) an OPEN Address Frame Received message is received or an OPEN Address Frame Received argument is included in the transition into this state; and
 - C) the destination SAS address and connection rate of the received OPEN address frame are equal to the source SAS address and connection rate of the transmitted OPEN address frame.

A Backoff Reverse Path response shall include the contents of the OPEN Address Frame Received message or argument.

This state shall send the following responses to the ECR, which are received by the source phy as confirmations:

- a) an Arb Status (Waiting On Device) response upon entry into this state;
- b) an Arb Status (Normal) response when an AIP Received (Normal) message is received;
- c) an Arb Status (Waiting On Partial) response when an AIP Received (Waiting On Partial) message is received;
- d) an Arb Status (Waiting On Connection) response when an AIP Received (Waiting On Connection) message is received; and
- e) an Arb Status (Waiting On Device) response when an AIP Received (Waiting On Device) message is received.

If a BREAK Received message is received or a BREAK Received argument is included in the transition into this state, then this state shall send a Forward Break request to the ECR (see 6.16.9.6).

This state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM until an AIP Received (Waiting On Partial) message is received. After an AIP Received (Waiting On Partial) message is received, this state shall repeatedly send a Phy Status (Blocked Partial Pathway) response to the ECM.

If a PS_REQ Received message is received this state shall send a Transmit PS_NAK to the XL transmitter.

6.16.9.2 Transition XL6:Open_Response_Wait to XL0:Idle

This transition shall occur after sending an Open Reject response to the ECR.

6.16.9.3 Transition XL6:Open_Response_Wait to XL1:Request_Path

This transition shall occur after sending a Backoff Retry response to the ECR.

6.16.9.4 Transition XL6:Open_Response_Wait to XL2:Request_Open

This transition shall occur after sending a Backoff Reverse Path response to the ECR.

6.16.9.5 Transition XL6:Open_Response_Wait to XL7:Connected

This transition shall occur after sending an Open Accept response to the ECR.

6.16.9.6 Transition XL6:Open_Response_Wait to XL9:Break

This transition shall occur after sending a Forward Break response to the ECR.

6.16.9.7 Transition XL6:Open_Response_Wait to XL10:Break_Wait

This transition shall occur after receiving a Forward Break indication if:

- a) a BREAK Received message has not been received; and
- b) a BREAK Received argument was not included in the transition into this state.

6.16.10 XL7:Connected state

6.16.10.1 State description

This state provides a full-duplex path between two phys within an expander device.

This state shall send Transmit Dword messages to the XL transmitter to transmit all dwords received with Forward Dword indications. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence (see 6.2.4).

If this state has not sent a Forward Close request to the ECR, then this state shall send Forward Dword requests to the ECR containing each valid dword except BREAK and CLOSE received with Dword Received

messages. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence (see 6.2.4).

If:

- a) an Invalid Dword Received message is received; and
- b) the expander phy is forwarding to an expander logical phy attached to a SAS logical link,

the expander logical phy shall:

- a) send an ERROR with the Forward Dword request instead of the invalid dword; or
- b) delete the invalid dword.

If:

- a) an ERROR is received with the Dword Received message or an Invalid Dword Received message is received; and
- b) the expander phy is forwarding to an expander logical phy attached to a SATA phy,

the expander logical phy shall:

- a) send a SATA_ERROR with the Forward Dword request instead of the invalid dword or ERROR primitive; or
- b) delete the ERROR or invalid dword.

If a CLOSE Received message is received, then this state shall send a Forward Close request to the ECR with the argument from the CLOSE Received message.

If a BREAK Received message is received, then this state shall send a Forward Break request to the ECR (see 6.16.10.3).

This state shall repeatedly send a Phy Status (Connection) response to the ECM.

6.16.10.2 Transition XL7:Connected to XL8:Close_Wait

This transition shall occur after receiving a Forward Close indication if a BREAK Received message has not been received.

6.16.10.3 Transition XL7:Connected to XL9:Break

This transition shall occur after sending a Forward Break request to the ECR.

6.16.10.4 Transition XL7:Connected to XL10:Break_Wait

This transition shall occur after receiving a Forward Break indication if a BREAK Received message has not been received. If this state has sent a Forward Close request to the ECR, then this transition shall include a Close Forwarded argument.

6.16.11 XL8:Close_Wait state

6.16.11.1 State description

This state closes a connection and releases path resources.

Upon entry into this state, this state shall send a Transmit CLOSE message to the XL transmitter with the argument from the Forward Close indication, then shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

NOTE 51 Possible livelock scenarios occur if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled and a phy transmits a BREAK primitive sequence to break a connection (e.g., if its Close Timeout timer expires). Phys should respond to a CLOSE primitive sequence faster than 1 ms to reduce susceptibility to this problem.

If a Dword Received message is received containing a valid dword except a BREAK or CLOSE, then this state shall send a Forward Dword request to the ECR containing that dword. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence (see 6.2.4).

If:

- a) an Invalid Dword Received message is received; and
- b) the expander logical phy is forwarding to an expander logical phy attached to a SAS logical link,

the expander logical phy shall:

- a) send an ERROR with the Forward Dword request instead of the invalid dword; or
- b) delete the invalid dword.

If:

- a) an ERROR is received with the Dword Received message or an Invalid Dword Received message is received; and
- b) the expander logical phy is forwarding to an expander phy attached to a SATA physical link,

the expander logical phy shall:

- a) send a SATA_ERROR with the Forward Dword request instead of the invalid dword or ERROR primitive; or
- b) delete the ERROR or invalid dword.

If a CLOSE Received message is received, then this state shall release path resources and send a Forward Close request to the ECR with the argument from the CLOSE Received message (see 6.16.11.2).

If this state was entered with an argument of Close Forwarded, then this state shall release path resources.

If a BREAK Received message is received, then this state shall send a Forward Break request to the ECR (see 6.16.11.3).

This state shall repeatedly send a Phy Status (Connection) response to the ECM.

6.16.11.2 Transition XL8:Close_Wait to XL0:Idle

This transition shall occur after:

- a) sending a Forward Close request to the ECR; or
- b) sending a Transmit CLOSE message to the XL transmitter if this state was entered with an argument of Close Forwarded.

6.16.11.3 Transition XL8:Close_Wait to XL9:Break

This transition shall occur after sending a Forward Break request to the ECR.

6.16.11.4 Transition XL8:Close_Wait to XL10:Break_Wait

This transition shall occur after receiving a Forward Break indication if a BREAK Received message has not been received.

6.16.12 XL9:Break state

6.16.12.1 State description

This state closes the connection, if there is one, and releases all path resources associated with the connection.

This state shall:

- a) send a Transmit BREAK_REPLY message to the XL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5); and

- b) send a Transmit BREAK message to the XL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled (see 6.13.5).

6.16.12.2 Transition XL9:Break to XL0:Idle

This transition shall occur after sending a Transmit BREAK message or a Transmit BREAK_REPLY message to the XL transmitter.

6.16.13 XL10:Break_Wait state

6.16.13.1 State description

This state closes the connection, if there is one, and releases path resources associated with the connection.

Upon entry into this state, this state shall:

- 1) send a Transmit BREAK message to the XL transmitter;
- 2) initialize and start the Break Timeout timer; and
- 3) repeatedly send a Phy Status (Breaking Connection) request to the ECM.

If a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the XL transmitter.

6.16.13.2 Transition XL10:Break_Wait to XL0:Idle

This transition shall occur after:

- a) a BREAK_REPLY Received message is received if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5);
- b) a BREAK Received message is received if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled (see 6.13.5); or
- c) the Break Timeout timer expires.

6.16.14 XL11:PS_Request state

6.16.14.1 State description

This state requests the attached phy change to a low phy power condition.

Upon entry into this state, this state shall:

- a) if entered with the Phy Power Condition (Enter Partial) argument, then send a Transmit PS_REQ (Partial) message to the XL transmitter;
- b) if entered with the Phy Power Condition (Enter Slumber) argument, then send a Transmit PS_REQ (Slumber) message to the XL transmitter; and
- c) initialize and start the Power Condition Request Timeout timer.

If this state:

- a) is entered with the Phy Power Condition (Enter Partial) argument; and
- b) receives a PS_REQ Received (Slumber) message.

then this state shall set the Phy Power Condition argument to Partial.

If this state:

- a) is entered with the Phy Power Condition (Enter Slumber) argument; and
- b) receives a PS_REQ Received (Partial) message.

then this state shall:

- a) set the Phy Power Condition argument to Partial; and
- b) if the Accept Partial state machine variable is set to:

- A) YES, then send a Transmit PS_ACK Pattern message to the XL transmitter; or
- B) NO, then send a Transmit PS_NAK message to the XL transmitter.

If this state:

- a) receives a PS_REQ Received message;
- b) the PS_REQ Received message argument is requesting the same power condition as the Phy Power Condition argument; and
- c) the attached SAS address is greater than the port identifier (i.e., SAS Address),

then this state shall:

- a) send a Transmit PS_ACK Pattern message to the XL transmitter; and
- b) send a Manage Phy Power Conditions (Stop DWS) request to the phy layer.

If this state receives a Forward Open indication, this state shall send a Backoff Retry response to the ECR.

If a BROADCAST Received message is received, this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive sequence received (e.g., Change Received).

If a Transmit Broadcast request is received, then this state shall send a Retry Transmit Broadcast request to the BPP.

If a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 6.13.5), then this state shall send a Transmit BREAK_REPLY message to the XL transmitter.

6.16.14.2 Transition XL11:PS_Request to XL12:PS_Quiet

This transition shall occur after:

- a) receiving a PS_ACK Pattern Transmitted message from the XL transmitter; or
- b) receiving a PS_ACK Received message.

This transition shall include the Phy Power Condition argument.

6.16.14.3 Transition XL11:PS_Request to XL0:Idle

This transition shall occur after:

- a) sending a Transmit PS_NAK message to the XL transmitter;
- b) a PS_NAK Received message is received;
- c) sending a Transmit BREAK_REPLY message to the XL transmitter;
- d) a BREAK Received message is received and the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled (see 6.13.5); or
- e) the Power Condition Request Timeout timer expires.

6.16.14.4 Transition XL11:PS_Request to XL1:Request_Path

This transition shall occur after receiving an OPEN Address Frame Received message.

6.16.15 XL12:PS_Quiet state

6.16.15.1 State description

This state requests the phy be placed into a low power condition.

If this state is entered with a Phy Power Condition (Enter Partial) argument, then this state shall send:

- a) a Manage Phy Power Conditions (Enter Partial) request to the phy layer; and
- b) a Phy Power Condition Status (In Partial) confirmation to the ECM.

If this state is entered with a Phy Power Condition (Enter Slumber) argument, then this state shall send:

- a) a Manage Phy Power Conditions (Enter Slumber) request to the phy layer; and
- b) a Phy Power Condition Status (In Slumber) confirmation to the ECM.

If this state receives a:

- a) Change Phy Power Condition (Enter Slumber) request and the current phy power condition is partial;
or
- b) Change Phy Power Condition (Enter Partial) request and the current phy power condition is slumber,

then this state shall send a Phy Power Condition Status (Request Exit Power Condition) confirmation to the ECM.

If a Transmit Broadcast request is received, then this state shall send a Retry Transmit Broadcast request to the BPP.

If this state receives a Change Phy Power Condition (Exit Power Condition) request, then this state shall send a Manage Phy Power Conditions (Exit) request to the phy layer.

If this state receives an Forward Open indication from the ECM, then this state shall send a Manage Phy Power Conditions (Exit) request to the phy layer and:

- a) if the phy is in a slumber phy power condition, send an Open Reject (Retry) response to the ECR; or
- b) if the phy is in a partial phy power condition, send a Backoff Retry response to the ECR.

6.16.15.2 Transition XL12:PS_Quiet to XL0:Idle

This transition shall occur after a Phy Layer Ready (SAS) confirmation is received.

6.17 SSP link layer

6.17.1 Opening an SSP connection

An SSP phy that accepts a connection request (i.e., an OPEN address frame) shall transmit at least one RRDY in that connection within 1 ms of transmitting an OPEN_ACCEPT. If the SSP phy is not able to grant credit, then it shall respond with OPEN_REJECT (RETRY) and not accept the connection request.

To prevent livelocks (e.g., where ports are waiting on each other to accept a connection request):

- a) a SAS phy shall not reject an incoming connection request to an SSP initiator port with OPEN_REJECT (RETRY) because the SAS port containing that SAS phy needs an outgoing connection request to be accepted (e.g., if the SAS phy is used by an SSP initiator port and an SSP target port, they share a buffer, that buffer is being used by the SSP target port, and the SSP target port needs to transmit a frame to another SSP initiator port before it is able to free that buffer);
- b) a SAS phy may reject an incoming connection request to an SSP initiator port with OPEN_REJECT (RETRY) for any reason that is not dependent on the SAS port containing that SAS phy having an outgoing connection request accepted (e.g., a temporary buffer full condition); and
- c) a SAS phy may reject an incoming connection request to an SSP target port with OPEN_REJECT (RETRY) for any reason, including because the SAS port containing that SAS phy needs an outgoing connection request to be accepted (e.g., to transmit a frame and empty a buffer).

6.17.2 Full duplex

SSP is a full duplex protocol. An SSP phy may receive an SSP frame or primitive in a connection while it is transmitting an SSP frame or primitive in the same connection. A wide SSP port may send and/or receive SSP frames or primitives concurrently on different connections (i.e., on different phys).

When a connection is open and an SSP phy has no more SSP frames to transmit on that connection, the SSP phy transmits a DONE to start closing the connection (see 7.2.2.3.4). The other direction may still be active, so the DONE may be followed by one or more of the following primitives:

- a) CREDIT_BLOCKED;
- b) RRDY;
- c) ACK; or
- d) NAK.

6.17.3 SSP frame transmission and reception

During an SSP connection, SSP frames are preceded by SOF and followed by EOF as shown in figure 117.

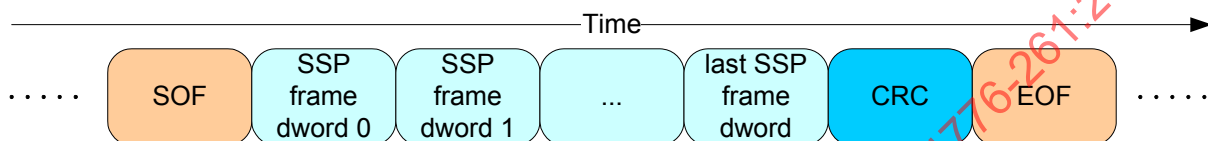


Figure 117 — SSP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 6.5). The SSP link layer state machine checks that the frame is a valid length and that the CRC is valid (see 6.17.8.7). Other primitives (e.g., CREDIT_BLOCKED, RRDY, ACK, and NAK) may be interspersed between the SOF, data dwords, and EOF.

Receiving SSP phys shall acknowledge SSP frames within 1 ms, if not discarded as described in 6.17.8.7, with either:

- a) ACK (i.e., positive acknowledgement) if the SSP frame was received into a frame buffer without errors; or
- b) NAK (CRC ERROR) (i.e., negative acknowledgement) if the SSP frame was received with a CRC error (i.e., a bad CRC), an invalid dword, or an ERROR.

NOTE 52 It is not required that frame recipients generate NAK (CRC ERROR) from invalid dwords and ERRORS (see 6.17.8.2).

If an SSP frame results in a link layer error (e.g., is NAKed or creates an ACK/NAK timeout), then:

- a) the transport layer (see 8.2.4) retries sending the SSP frame; or
- b) the SCSI application layer aborts the SCSI command associated with that SSP frame.

6.17.4 SSP flow control

An SSP phy uses RRDY to grant credit for permission for the other SSP phy in the connection to transmit frames. Each RRDY increments credit by one frame. Frame transmission decrements credit by one frame. Credit of zero frames is established at the beginning of each connection.

SSP phys shall not increment credit past 255 frames.

To prevent deadlocks where an SSP initiator port and SSP target port are both waiting on each other to provide credit, an SSP initiator port shall not refuse to provide credit by withholding RRDY because it needs to transmit a frame itself. An SSP initiator port may refuse to provide credit for other reasons (e.g., temporary buffer full conditions).

An SSP target port may refuse to provide credit for any reason, including because it needs to transmit a frame itself.

If credit is zero, SSP phys that are going to be unable to provide credit for 1 ms may send CREDIT_BLOCKED. The other phy may use this to avoid waiting 1 ms to transmit DONE (CREDIT TIMEOUT) (see 6.17.8).

If credit is nonzero, SSP phys that are going to be unable to provide additional credit for 1 ms, even if they receive frames per the existing credit, may transmit CREDIT_BLOCKED.

After sending CREDIT_BLOCKED, an SSP phy shall not transmit any additional RRDYs in the connection.

6.17.5 Interlocked frames

Table 108 shows which SSP frames shall be interlocked and which are non-interlocked.

Table 108 — SSP frame interlock requirements

SSP frame type	Interlock requirement
COMMAND	Interlocked
TASK	Interlocked
XFER_RDY	Interlocked
DATA	Non-interlocked
RESPONSE	Interlocked
See 8.2 for SSP frame type definitions.	

Before transmitting an interlocked frame, an SSP phy shall wait for all SSP frames to be acknowledged with ACK or NAK, even if credit is available. After transmitting an interlocked frame, an SSP phy shall not transmit another SSP frame until that interlocked frame has been acknowledged with ACK or NAK, even if credit is available.

Before transmitting a non-interlocked frame, an SSP phy shall wait for:

- a) all non-interlocked frames with different initiator port transfer tags; and
- b) all interlocked frames,

to be acknowledged with ACK or NAK, even if credit is available.

After transmitting a non-interlocked frame, an SSP phy may transmit another non-interlocked frame with the same initiator port transfer tag if credit is available. The phy shall not transmit:

- a) a non-interlocked frame with a different initiator port transfer tag; or
- b) an interlocked frame.

until all SSP frames have been acknowledged with ACK or NAK, even if credit is available.

Interlocking does not prevent transmitting and receiving interlocked frames simultaneously (e.g., it is possible for an SSP initiator phy to be transmitting a COMMAND frame while receiving XFER_RDY, DATA, or RESPONSE frames for a different command).

An SSP phy may transmit primitives responding to traffic that it is receiving (e.g., an ACK or NAK to acknowledge an SSP frame, an RRDY to grant more receive credit, or a CREDIT_BLOCKED to specify that no more RRDYs are going to be transmitted in the connection) while waiting for an interlocked frame that it transmitted to be acknowledged. These primitives may also be interspersed within an SSP frame.

Figure 118 shows an example of interlocked frame transmission.

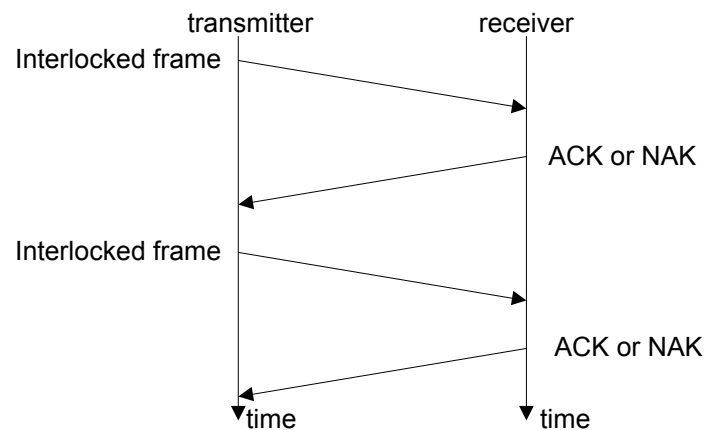


Figure 118 — Interlocked frames

Figure 119 shows an example of non-interlocked frame transmission with the same initiator port transfer tags.

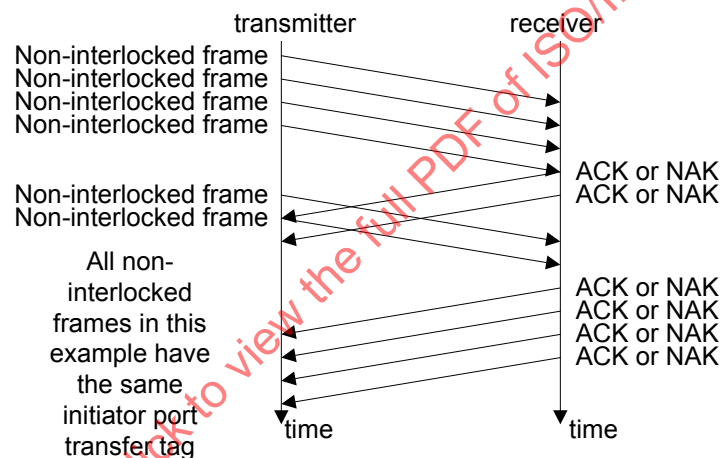


Figure 119 — Non-interlocked frames with the same initiator port transfer tags

Figure 120 shows an example of non-interlocked frame transmission with different initiator port transfer tags.

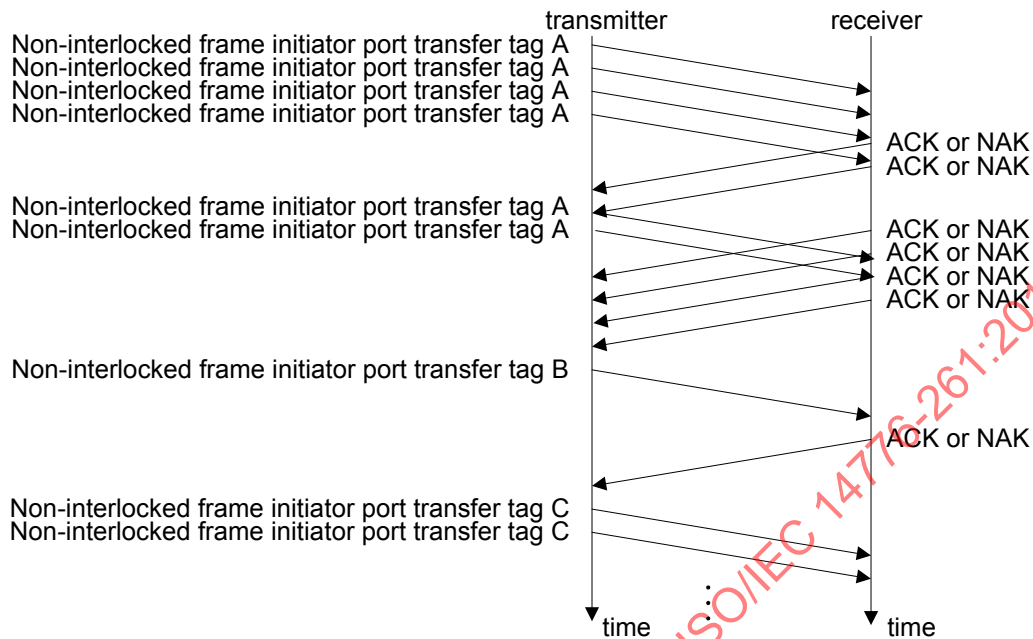


Figure 120 — Non-interlocked frames with different initiator port transfer tags

6.17.6 Breaking an SSP connection

In addition to the actions described in 6.13.8, the following shall be the responses by an SSP phy to a broken connection:

- received frames having no CRC error may be considered valid regardless of whether an ACK has been transmitted in response to the frame prior to the broken connection;
- transmitted frames for which an ACK has been received prior to a broken connection shall be considered transmitted without error; and
- transmitted frames for which an ACK or NAK has not been received prior to a broken connection shall be considered to have been transmitted with an error.

6.17.7 Closing an SSP connection

DONE shall be exchanged prior to closing an SSP connection (see 7.2.2.3.4). The type of DONE indicates additional information about why the SSP connection is being closed as follows:

- DONE (NORMAL) specifies that the transmitter has no more SSP frames to transmit (i.e., normal completion);
- DONE (CREDIT TIMEOUT) specifies that the transmitter still has SSP frames to transmit but did not receive an RRDY granting frame credit within 1 ms, or the transmitter has received a CREDIT_BLOCKED and has consumed all RRDYs received; and
- DONE (ACK/NAK TIMEOUT) specifies that the transmitter transmitted an SSP frame but did not receive the corresponding ACK or NAK within 1 ms. As a result, the ACK/NAK count is not balanced and the transmitter is going to transmit a BREAK primitive sequence in 1 ms unless the recipient replies with DONE and the connection is closed.

If the transmitter has no more SSP frames to transmit and receives a CREDIT_BLOCKED, then it may transmit either DONE (NORMAL) or DONE (CREDIT TIMEOUT).

After transmitting DONE, the transmitting phy initializes and starts a 1 ms DONE Timeout timer (see 6.17.8.5).

After transmitting DONE, the transmitting phy shall not transmit any more SSP frames during this connection. However, the phy may transmit ACK, NAK, RRDY, and CREDIT_BLOCKED as needed after transmitting

DONE if the other phy is still transmitting SSP frames in the reverse direction. Once an SSP phy has both transmitted and received DONE, it shall close the connection by transmitting a CLOSE (NORMAL) primitive sequence (see 6.13.7).

Figure 121 shows the sequence for a closing an SSP connection.

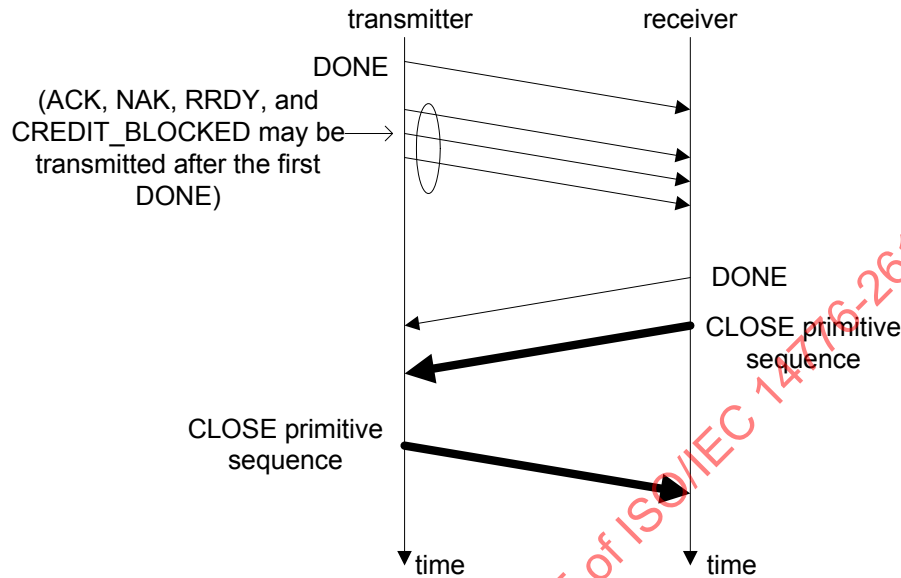


Figure 121 — Closing an SSP connection example

6.17.8 SSP (link layer for SSP phys) state machines

6.17.8.1 SSP state machines overview

The SSP link layer contains several state machines that run in parallel to control the flow of dwords on the physical link during an SSP connection. The SSP state machines are as follows:

- SSP_TIM (transmit interlocked frame monitor) state machine (see 6.17.8.3);
- SSP_TCM (transmit frame credit monitor) state machine (see 6.17.8.4);
- SSP_D (DONE control) state machine (see 6.17.8.5);
- SSP_TF (transmit frame control) state machine (see 6.17.8.6);
- SSP_RF (receive frame control) state machine (see 6.17.8.7);
- SSP_RCM (receive frame credit monitor) state machine (see 6.17.8.8);
- SSP_RIM (receive interlocked frame monitor) state machine (see 6.17.8.9);
- SSP_TC (transmit credit control) state machine (see 6.17.8.10); and
- SSP_TAN (transmit ACK/NAK control) state machine (see 6.17.8.11).

All the SSP state machines shall start after receiving an Enable Disable SSP (Enable) message from the SL state machines (see 6.15).

All the SSP state machines shall terminate after:

- receiving an Enable Disable SSP (Disable) message from the SL state machines;
- receiving a Request Close message from the SSP_D state machine indicating that the connection has been closed;
- receiving a Request Break message from the SSP_D state machine indicating that a BREAK primitive sequence has been transmitted; or
- receiving a NOTIFY Received (Power Loss Expected) message from the SP_DWS receiver if the SAS port that contains this state machine supports NOTIFY (Power Loss Expected) (e.g., the SAS port is an SSP target port).

If a state machine consists of multiple states, then the initial state is as indicated in the state machine description in this subclause.

The SSP state machines shall maintain the timers listed in table 109.

Table 109 — SSP state machines timers

Timer	Initial value	State machine	Reference
ACK/NAK Timeout timer	1 ms	SSP_TIM	6.17.8.3
DONE Timeout timer	1 ms	SSP_D	6.17.8.5
Credit Timeout timer	1 ms	SSP_TF	6.17.8.6

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Figure 122 shows the SSP state machines and states related to frame transmission.

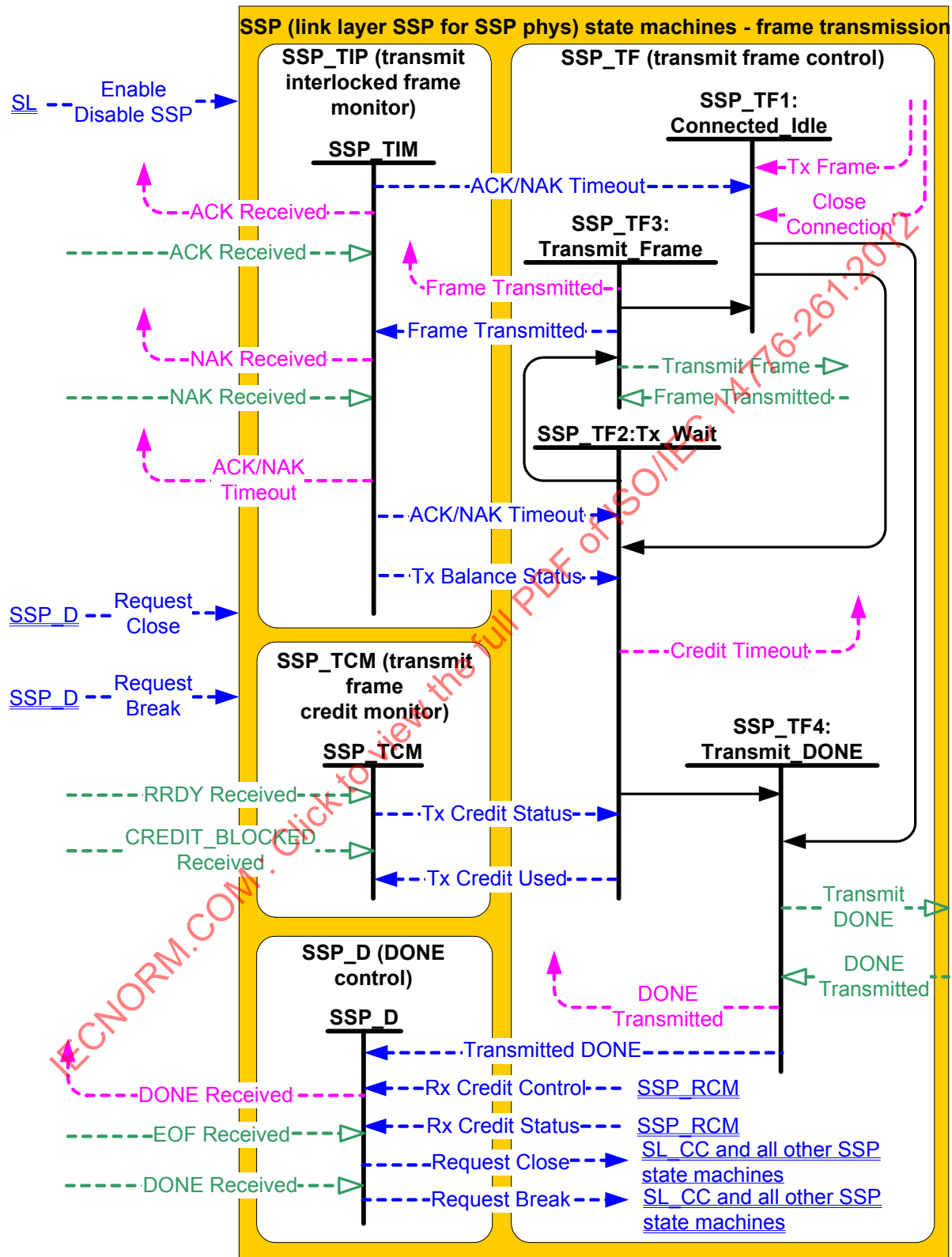


Figure 122 — SSP (link layer for SSP phys) state machines (part 1 - frame transmission)

Figure 123 shows the SSP state machines and states related to frame reception.

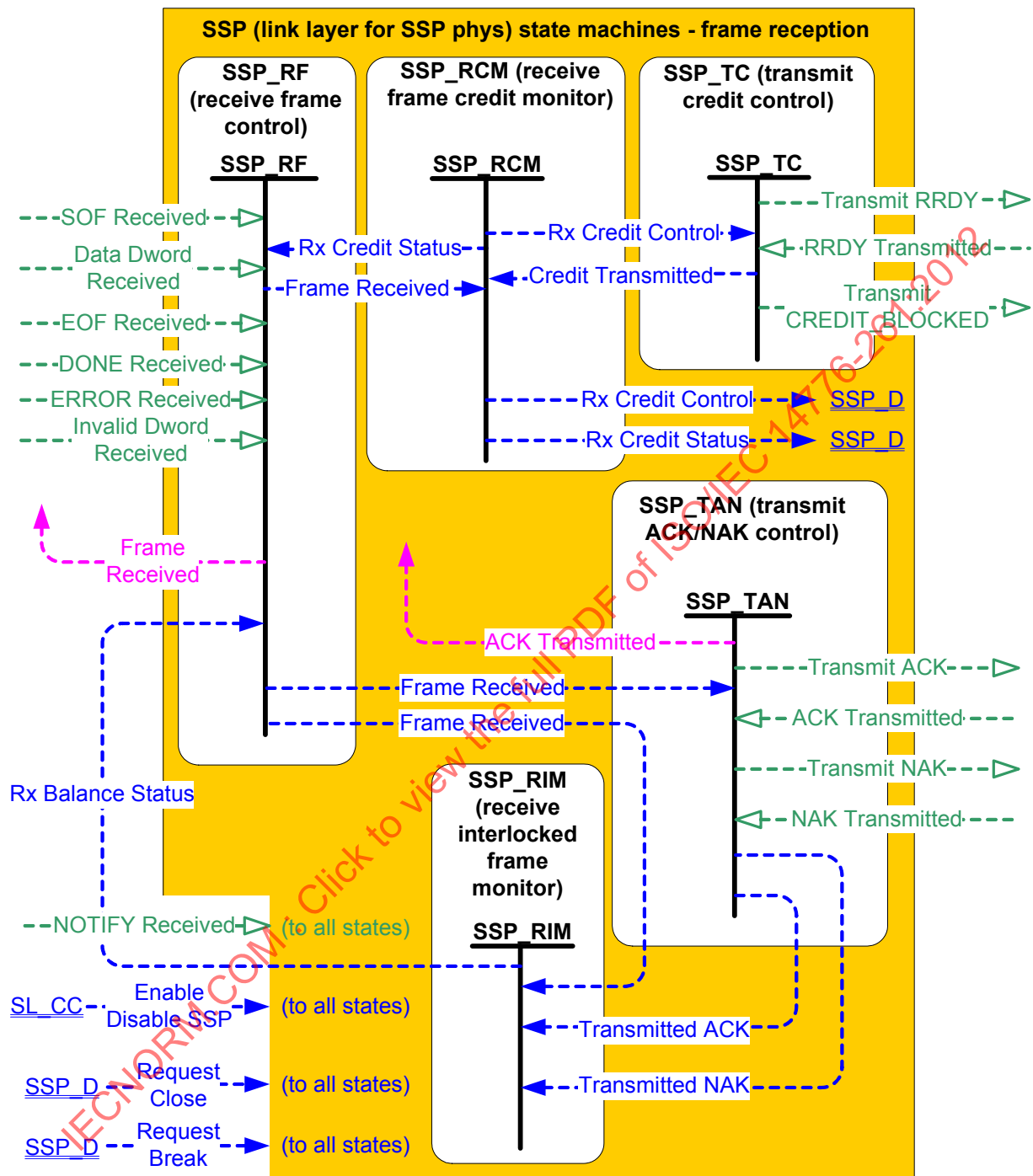


Figure 123 — SSP (link layer for SSP phys) state machines (part 2 - frame reception)

6.17.8.2 SSP transmitter and receiver

The SSP transmitter receives the following messages from the SSP state machines specifying primitive sequences and frames to transmit:

- Transmit RRDY with an argument indicating the specific type (e.g., Transmit RRDY (Normal));
- Transmit CREDIT_BLOCKED;

- c) Transmit ACK;
- d) Transmit NAK with an argument indicating the specific type (e.g., Transmit NAK (CRC Error));
- e) Transmit Frame with an argument containing the frame contents; and
- f) Transmit DONE with an argument indicating the specific type (e.g., Transmit DONE (Normal)).

In response to the Transmit Frame message, the SSP transmitter transmits:

- 1) SOF;
- 2) the frame contents;
- 3) CRC; and
- 4) EOF.

The SSP transmitter sends the following messages to the SSP state machines based on dwords that have been transmitted:

- a) DONE Transmitted;
- b) RRDY Transmitted;
- c) CREDIT_BLOCKED Transmitted;
- d) ACK Transmitted;
- e) NAK Transmitted; and
- f) Frame Transmitted.

When there is no outstanding message specifying a dword to transmit, the SSP transmitter shall transmit idle dwords.

The SSP receiver sends the following messages to the SSP state machines indicating primitive sequences and dwords received from the SP_DWS receiver (see 5.10.2):

- a) ACK Received;
- b) NAK Received;
- c) RRDY Received;
- d) CREDIT_BLOCKED Received;
- e) DONE Received with an argument indicating the specific type (e.g., DONE Received (Normal));
- f) SOF Received;
- g) Data Dword Received;
- h) EOF Received;
- i) NOTIFY Received (Power Loss Expected);
- j) ERROR Received; and
- k) Invalid Dword Received.

The SSP receiver shall ignore all other dwords.

The SSP transmitter relationship to other transmitters is defined in 4.3.2. The SSP receiver relationship to other receivers is defined in 4.3.3.

6.17.8.3 SSP_TIM (transmit interlocked frame monitor) state machine

The SSP_TIM state machine's function is to ensure that ACKs or NAKs are received for each transmitted frame before the ACK/NAK timeout. This state machine consists of one state.

This state machine monitors the number of frames transmitted with a Number Of Frames Transmitted counter and monitors the number of ACKs and NAKs received with a Number Of ACKs/NAKs Received counter. This state machine ensures that an ACK or NAK is received for each frame transmitted and reports an ACK/NAK timeout if they are not.

When the Number Of Frames Transmitted counter equals the Number Of ACKs/NAKs Received counter, the ACK/NAK count is balanced and this state machine shall send the Tx Balance Status (Balanced) message to the SSP_TF2:Tx_Wait state. When the Number Of Frames Transmitted counter does not equal the Number Of ACKs/NAKs Received counter, the ACK/NAK count is not balanced and this state machine shall send the Tx Balance Status (Not Balanced) message to the SSP_TF2:Tx_Wait state.

Each time a Frame Transmitted message is received, this state machine shall increment the Number Of Frames Transmitted counter.

If the ACK/NAK count is not balanced, then each time an ACK Received message is received, this state machine shall:

- a) increment the Number Of ACKs/NAKs Received counter; and
- b) send an ACK Received confirmation to the port layer.

If the ACK/NAK count is not balanced, then each time a NAK Received message is received, this state machine shall:

- a) increment the Number Of ACKs/NAKs Received counter; and
- b) send an NAK Received confirmation to the port layer.

If the ACK/NAK count is balanced, then the ACK Received message and NAK Received message shall be ignored and the ACK/NAK Timeout timer shall be stopped.

Each time the ACK/NAK count is not balanced, the ACK/NAK Timeout timer shall be initialized and started. The ACK/NAK Timeout timer shall be reinitialized each time the Number Of ACKs/NAKs Received counter is incremented. If the ACK/NAK Timeout timer expires, then this state machine shall send the ACK/NAK Timeout confirmation to the port layer and to the following states:

- a) SSP_TF1:Connected_Idle; and
- b) SSP_TF2:Tx_Wait state.

When this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message, the Number Of Frames Transmitted counter shall be set to zero and the Number Of ACKs/NAKs Received counter shall be set to zero.

6.17.8.4 SSP_TCM (transmit frame credit monitor) state machine

The SSP_TCM state machine's function is to ensure that transmit frame credit is available before a frame is transmitted. This state machine consists of one state.

This state machine shall keep track of the number of transmit frame credits available. This state machine shall add one transmit frame credit for each RRDY Received message received and subtract one transmit frame credit for each Tx Credit Used message received.

The CREDIT_BLOCKED Received message indicates that transmit frame credit is blocked. After receiving a CREDIT_BLOCKED Received message, this state machine may ignore additional RRDY Received messages until it receives a Request Close message or a Request Break message.

When transmit frame credit is available, this state machine shall send the Tx Credit Status (Available) message to the SSP_TF2:Tx_Wait state.

When transmit frame credit is not available and transmit frame credit is not blocked, this state machine shall send the Tx Credit Status (Not Available) message to the SSP_TF2:Tx_Wait state.

When transmit frame credit is not available and transmit frame credit is blocked, this state machine shall send the Tx Credit Status (Blocked) message to the SSP_TF2:Tx_Wait state.

When this state machine receives an Enable Disable SSP (Enable) message, a Request Close message, or a Request Break message, this state shall set transmit frame credit to not available and transmit frame credit shall be set to not blocked.

6.17.8.5 SSP_D (DONE control) state machine

The SSP_D state machine's function is to ensure a DONE has been received and transmitted before the SL_CC state machine disables the SSP state machines. This state machine consists of one state.

This state machine ensures that a DONE is received and transmitted before the connection is closed. The DONE may be transmitted and received in any order.

If a DONE Received message has been received before a Transmitted DONE message is received, then this state machine shall send a Request Close message to the SL_CC state machine (see 6.15) and all the SSP state machines after receiving the Transmitted DONE message.

If a DONE Received message, a Transmitted DONE (Normal) message, or a Transmitted DONE (Credit Timeout) message has not been received and an Rx Credit Status (Extended) message or an Rx Credit Control (Blocked) message has been received, then this state shall initialize and start the DONE Timeout timer after receiving a Transmitted DONE (Normal) message or a Transmitted DONE (Credit Timeout) message.

If a DONE Received message has not been received and a Transmitted DONE (Normal) message or a Transmitted DONE (Credit Timeout) message has been received, then this state machine shall initialize and start the DONE Timeout timer each time:

- a) a Rx Credit Status (Extended) message is received; or
- b) a Rx Credit Control (Blocked) message is received.

If a Transmitted DONE (Normal) message or a Transmitted DONE (Credit Timeout) message has been received, then the DONE Timeout timer shall be reinitialized each time an EOF Received message is received.

If a Transmitted DONE (Normal) message or a Transmitted DONE (Credit Timeout) message has been received, then the DONE Timeout timer shall be stopped after:

- a) an Rx Credit Status (Exhausted) message is received; and
- b) an Rx Credit Control (Blocked) message has not been received.

NOTE 53 Stopping the timer ensures that, if credit remains exhausted long enough that the Credit Timeout timer of the other phy in the connection expires, the other phy is able to transmit a DONE (CREDIT TIMEOUT).

If a DONE Received message has not been received and a Transmitted DONE (ACK/NAK Timeout) message has been received, then:

- a) this state machine shall initialize and start the DONE Timeout timer; and
- b) this state shall not reinitialize the DONE Timeout timer if an EOF Received message is received.

If a DONE Received message is received before the DONE Timeout timer expires, then this state machine shall send a Request Close message to the SL_CC state machine and all the SSP state machines.

If a DONE Received message is not received before the DONE Timeout timer expires, then this state machine shall send a Request Break message to the SL_CC state machine and all the SSP state machines.

Any time a DONE Received message is received, this state machine shall send a DONE Received confirmation to the port layer. A DONE Received (ACK/NAK Timeout) confirmation informs the port layer that the SSP transmitter is going to close the connection within 1 ms; other DONE Received confirmations (e.g., DONE Received (Normal) and DONE Received (Credit Timeout)) may be used by the SCSI application layer to decide when to reuse initiator port transfer tags (see 9.2.2).

NOTE 54 The DONE Timeout timer in one phy (e.g., phy A) may expire concurrently with the ACK/NAK Timeout timer in the other phy (e.g., phy B) in a connection.

For example, if phy A receives DONE (NORMAL) indicating phy B has no more frames to transmit, and phy A then transmits a series of non-interlocked frames where one or more of the SOFs is corrupted, then phy A waits to receive all the ACKs and/or NAKs after transmitting the series of non-interlocked frames. However, since phy B did not receive the full number of SOFs, it does not transmit as many ACKs and/or NAKs as phy A is expecting. The ACK/NAK Timeout timer in phy A expires and phy A transmits DONE (ACK/NAK TIMEOUT). Meanwhile, despite having transmitted DONE, phy B stops receiving frames while phy A is waiting for the final ACKs and/or NAKs. Since phy B does not receive DONE or any more frames, its DONE Timeout timer expires and phy B transmits a BREAK primitive sequence.

Since the timers may expire at different times (e.g., due to timer resolution differences), the DONE (ACK/NAK TIMEOUT) may be transmitted before, concurrently with, or after the BREAK primitive sequence. Nevertheless, the phys handle the link layer error (i.e., the ACK/NAK timeout or the DONE timeout) the same way (see 8.2.4.5 and 8.2.4.6).

6.17.8.6 SSP_TF (transmit frame control) state machine

6.17.8.6.1 SSP_TF state machine overview

The SSP_TF state machine's function is to control when the SSP transmitter transmits SOF, frame dwords, EOF, and DONE. This state machine consists of the following states:

- a) SSP_TF1:Connected_Idle (see 6.17.8.6.2) (initial state);
- b) SSP_TF2:Tx_Wait (see 6.17.8.6.3);
- c) SSP_TF3:Transmit_Frame (see 6.17.8.6.4); and
- d) SSP_TF4:Transmit_DONE (see 6.17.8.6.5).

This state machine shall start in the SSP_TF1:Connected_Idle state.

6.17.8.6.2 SSP_TF1:Connected_Idle state

6.17.8.6.2.1 State description

This state waits for a request to transmit a frame or to close the connection.

6.17.8.6.2.2 Transition SSP_TF1:Connected_Idle to SSP_TF2:Tx_Wait

This transition shall occur after a Tx Frame request is received or a Close Connection request is received.

If a Tx Frame (Balance Required) request was received, then this transition shall include a Transmit Frame Balance Required argument.

If a Tx Frame (Balance Not Required) request was received, then this transition shall include a Transmit Frame Balance Not Required argument.

If a Close Connection request was received, then this transition shall include a Close Connection argument.

6.17.8.6.2.3 Transition SSP_TF1:Connected_Idle to SSP_TF4:Transmit_DONE

This transition shall occur if an ACK/NAK Timeout message is received. This transition shall include an ACK/NAK Timeout argument.

6.17.8.6.3 SSP_TF2:Tx_Wait state

6.17.8.6.3.1 State description

This state monitors the Tx Balance Status message and the Tx Credit Status message to ensure that frames are transmitted and connections are closed at the proper time.

If this state is entered from the SSP_TF1:Connected_Idle state with a Transmit Frame Balance Required argument or a Transmit Frame Balance Not Required argument, and:

- a) if the last Tx Credit Status message received had an argument of Not Available, then this state shall initialize and start the Credit Timeout timer; or
- b) if the last Tx Credit Status message had an argument other than Not Available, then this state shall stop the Credit Timeout timer.

6.17.8.6.3.2 Transition SSP_TF2:Tx_Wait to SSP_TF3:Transmit_Frame

This transition shall occur if this state was entered from the SSP_TF1:Connected_Idle state with an argument of Transmit Frame Balance Required if:

- a) the last Tx Balance Status message received had an argument of Balanced; and
- b) the last Tx Credit Status message received had an argument of Available.

This transition shall occur if this state was entered from the SSP_TF1:Connected_Idle state with an argument of Transmit Frame Balance Not Required and if the last Tx Credit Status message received had an argument of Available.

This transition shall occur after sending a Tx Credit Used message to the SSP_TCM state machine.

6.17.8.6.3.3 Transition SSP_TF2:Tx_Wait to SSP_TF4:Transmit_DONE

This transition shall occur and include an ACK/NAK Timeout argument if an ACK/NAK Timeout message is received.

This transition shall occur and include a Close Connection argument if:

- a) this state was entered from the SSP_TF1:Connected_Idle state with an argument of Close Connection; and
- b) the last Tx Balance Status message received had an argument of Balanced.

This transition shall occur after sending a Credit Timeout confirmation and include a Credit Timeout argument if:

- a) this state was entered from the SSP_TF1:Connected_Idle state with a Transmit Frame Balance Required argument or a Transmit Frame Balance Not Required argument;
- b) the Credit Timeout timer expired before a Tx Credit Status message was received with an argument of Available, or the last Tx Credit Status message received had an argument of Blocked;
- c) a Tx Balance Status message was received with an argument of Balanced (i.e., the Credit Timeout argument shall not be included in this transition for this reason unless the ACK/NAK count is balanced); and
- d) an ACK/NAK Timeout message was not received.

6.17.8.6.4 SSP_TF3:Transmit_Frame state

6.17.8.6.4.1 State description

This state shall request a frame transmission by sending a Transmit Frame message to the SSP transmitter with an argument containing the frame contents. Each time a Transmit Frame message is sent to the SSP transmitter, one SSP frame (i.e., SOF, frame contents, CRC, and EOF) is transmitted.

In this state receiving a Frame Transmitted message indicates that the frame has been transmitted.

6.17.8.6.4.2 Transition SSP_TF3:Transmit_Frame to SSP_TF1:Connected_Idle

This transition shall occur after:

- a) receiving a Frame Transmitted message;
- b) sending an Frame Transmitted message to the SSP_TIM state machine; and
- c) sending a Frame Transmitted confirmation to the port layer.

6.17.8.6.5 SSP_TF4:Transmit_DONE state

This state shall send one of the following messages to an SSP transmitter:

- a) a Transmit DONE (Normal) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Close Connection;
- b) a Transmit DONE (ACK/NAK Timeout) message if this state was entered from the SSP_TF2:Tx_Wait state or the SSP_TF1:Connected_Idle state with an argument of ACK/NAK Timeout; or
- c) a Transmit DONE (Credit Timeout) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Credit Timeout.

NOTE 55 Possible livelock scenarios occur if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled and a SAS logical phy transmits a BREAK primitive sequence to break a connection (e.g., if its Done Timeout timer expires). SAS logical phys should respond to DONE faster than 1 ms to reduce susceptibility to this problem.

After a DONE Transmitted message is received, this state shall send the DONE Transmitted confirmation to the port layer and send one of the following messages to the SSP_D state machine:

- a) a Transmitted DONE (Normal) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Close Connection;
- b) a Transmitted DONE (ACK/NAK Timeout) message if this state was entered from the SSP_TF2:Tx_Wait state or the SSP_TF1:Connected_Idle state with an argument of ACK/NAK Timeout; or
- c) a Transmitted DONE (Credit Timeout) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Credit Timeout.

6.17.8.7 SSP_RF (receive frame control) state machine

The SSP_RF state machine's function is to receive frames and determine whether or not those frames were received without error. This state machine consists of one state.

This state machine:

- a) checks the frame to determine if the frame should be accepted or discarded;
- b) checks the frame to determine if an ACK or NAK should be transmitted; and
- c) sends a Frame Received confirmation to the port layer.

If this state receives a subsequent SOF Received message after receiving an SOF Received message but before receiving an EOF Received message (i.e., SOF, data dwords, SOF, data dwords, and EOF instead of SOF, data dwords, EOF, SOF, data dwords, and EOF), then this state shall discard the frame in progress.

This state shall discard the frame if:

- a) this state receives more than 263 Data Dword Received messages (i.e., 1 052 bytes) after an SOF Received message and before an EOF Received message;
- b) this state receives fewer than 7 Data Dword Received messages (i.e., 28 bytes) after an SOF Received message and before an EOF Received message;
- c) this state receives an Rx Credit Status (Credit Exhausted) message; or
- d) this state receives a DONE Received message.

If this state receives an Invalid Dword Received message or an ERROR Received message after receiving an SOF Received message and before receiving an EOF Received message, then this state machine shall either:

- a) ignore the invalid dword or ERROR; or
- b) discard the frame, send a Frame Received message to the SSP_RCM state machine, send a Frame Received message to the SSP_RIM state machine, and send a Frame Received (Unsuccessful) message to the SSP_TAN state machine.

If the frame is not discarded and the frame CRC is bad, then this state machine shall:

- a) send a Frame Received message to the SSP_RCM state machine;
- b) send a Frame Received message to the SSP_RIM state machine; and
- c) send a Frame Received (Unsuccessful) message to the SSP_TAN state machine.

If the frame is not discarded and the frame CRC is good, then this state machine shall send a Frame Received (Successful) message to the SSP_TAN state machine, and:

- a) send a Frame Received message to the SSP_RCM state machine;
- b) send a Frame Received message to the SSP_RIM state machine; and
- c) send a Frame Received (Successful) message to the SSP_TAN state machine, and:
 - A) if the last Rx Balance Status message received had an argument of Balanced, then send a Frame Received (ACK/NAK Balanced) confirmation to the port layer; or
 - B) if the last Rx Balance Status message received had an argument of Not Balanced, then send a Frame Received (ACK/NAK Not Balanced) confirmation to the port layer.

6.17.8.8 SSP_RCM (receive frame credit monitor) state machine

The SSP_RCM state machine's function is to ensure that there was credit given to the originator for every frame that is received. This state machine consists of one state.

This state machine monitors the receiver's resources and keeps track of the number of RRDYs transmitted versus the number of frames received.

Any time resources are released or become available, if this state machine has not sent the Rx Credit Control (Blocked) message to the SSP_TC state machine and the SSP_D state machine, then this state machine shall send the Rx Credit Control (Available) message to the SSP_TC state machine. This state machine shall only send the Rx Credit Control (Available) message to the SSP_TC state machine after frame receive resources become available. The specifications for when or how resources become available is outside the scope of this standard.

This state machine may send the Rx Credit Control (Blocked) message to the SSP_TC state machine and the SSP_D state machine when no further receive frame credit is going to become available within a credit timeout (i.e., less than 1 ms), even if frames are received per the existing receive frame credit. After sending the Rx Credit Control (Blocked) message to the SSP_TC state machine and the SSP_D state machine, this state machine shall not send the Rx Credit Control (Available) message to the SSP_TC state machine or the SSP_D state machine for the duration of the current connection.

This state machine shall indicate through the Rx Credit Control message only the amount of resources available to handle received frames (e.g., if this state machine has resources for five frames, then the maximum number of Rx Credit Control requests with the Available argument outstanding is five).

This state machine shall use the Credit Transmitted message to keep track of the number of RRDYs transmitted. This state machine shall use the Frame Received message to keep a track of the number of frames received.

Any time the number of Credit Transmitted messages received exceeds the number of Frame Received messages received this state machine shall send a Rx Credit Status (Extended) message to the SSP_RF state machine and the SSP_D state machine.

Any time the number of Credit Transmitted messages received equals the number of Frame Received messages received this state machine shall send a Rx Credit Status (Exhausted) message to the SSP_RF state machine and the SSP_D state machine.

If this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message, the frame receive resources shall be initialized to the no credit value for the current connection.

6.17.8.9 SSP_RIM (receive interlocked frame monitor) state machine

The SSP_RIM state machine's function is to inform the SSP_RF state machine when the number of ACKs and NAKs transmitted equals the number of the EOFs received. This state machine consists of one state.

This state machine monitors the number of frames received with a Number Of Frames Received counter and monitors the number of ACKs and NAKs transmitted with a Number Of ACKs/NAKs Transmitted counter.

Each time a Frame Received message is received, this state machine shall increment the Number Of Frames Received counter.

Each time an ACK Transmitted message or a NAK Transmitted message is received, this state machine shall increment the Number Of ACKs/NAKs Transmitted counter.

When the Number Of Frames Received counter equals the Number Of ACKs/NAKs Transmitted counter, this state machine shall send an Rx Balance Status (Balanced) message to the SSP_RF state machine.

When the Number Of Frames Received counter does not equal the Number Of ACKs/NAKs Transmitted counter, this state machine shall send an Rx Balance Status (Not Balanced) message to the SSP_RF state machine.

When this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message, the Number Of Frames Received counter shall be set to zero and the Number Of ACKs/NAKs Transmitted counter shall be set to zero.

6.17.8.10 SSP_TC (transmit credit control) state machine

The SSP_TC state machine's function is to control the sending of requests to transmit an RRDY or CREDIT_BLOCKED. This state machine consists of one state.

Any time this state machine receives a Rx Credit Control (Available) message, it shall send a number of Transmit RRDY (Normal) messages to the SSP transmitter as indicated by the amount of resources available to handle received frames (e.g., if the Available argument indicates five RRDYs are to be transmitted, then this state machine sends five Transmit RRDY (Normal) messages to the SSP transmitter).

Any time this state machine receives a RRDY Transmitted message, it shall send a Credit Transmitted message to the SSP_RCM state machine.

Any time this state machine receives a Rx Credit Control (Blocked) message, it shall send a Transmit CREDIT_BLOCKED message to the SSP transmitter.

6.17.8.11 SSP_TAN (transmit ACK/NAK control) state machine

The SSP_TAN state machine's function is to control the sending of requests to transmit an ACK or NAK to the SSP transmitter. This state machine consists of one state.

Any time this state machine receives a Frame Received (Successful) message, it shall send a Transmit ACK message to the SSP transmitter.

Any time this state machine receives a Frame Received (Unsuccessful) message, it shall send a Transmit NAK (CRC Error) message to the SSP transmitter.

If multiple Frame Received (Unsuccessful) messages and Frame Received (Successful) messages are received, then the order in which the Transmit ACK messages and Transmit NAK messages are sent to the SSP transmitter shall be the same order as the Frame Received (Unsuccessful) messages and Frame Received (Successful) messages were received.

Any time this state machine receives an ACK Transmitted message, it shall:

- a) send a Transmitted ACK message to the SSP_RIM state machine; and
- b) send an ACK Transmitted confirmation to the port layer.

Any time this state machine receives a NAK Transmitted argument, it shall send a Transmitted NAK message to the SSP_RIM state machine.

6.18 STP link layer

6.18.1 STP frame transmission and reception

STP frame transmission is defined by SATA. During an STP connection, frames are preceded by SATA_SOF and followed by SATA_EOF as shown in figure 124.

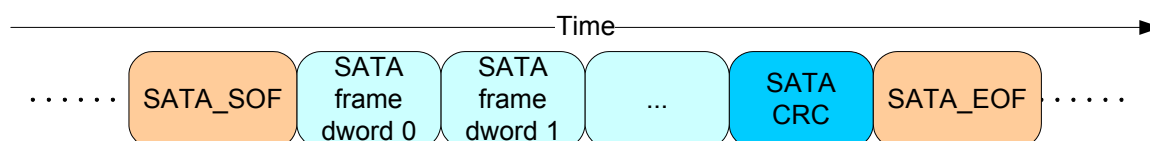


Figure 124 — STP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 6.5).

Other primitives may be interspersed during the connection as defined by SATA.

STP encapsulates SATA with connection management. Table 110 summarizes STP link layer differences from the SATA link layer (see SATA) that affect behavior during an STP connection.

Table 110 — STP link layer differences from SATA link layer during an STP connection

Feature	Description	Reference
STP flow control	Flow control through an STP connection is point-to-point, not end-to-end. Expander devices accept dwords in an STP flow control buffer after transmitting SATA_HOLD to avoid losing data en-route before the transmitting phy acknowledges the SATA_HOLD with SATA_HOLD_A.	6.18.2
Continued primitive sequence	Sustain the continued primitive sequence if a SATA_CONT appears after the continued primitive sequence has begun.	6.18.3

6.18.2 STP flow control

6.18.2.1 STP flow control overview

Each STP phy (i.e., STP initiator phy and STP target phy) and expander logical phy through which the STP connection is routed shall implement the SATA flow control protocol on each logical link in the pathway. The flow control primitives are not forwarded through expander devices like other dwords.

6.18.2.2 SATA frame buffering

When an STP phy or expander phy during an STP connection is receiving a SATA frame and its STP flow control buffer begins to fill up, that STP phy or expander phy shall transmit SATA_HOLD. After transmitting SATA_HOLD, the STP phy or expander phy shall accept at least the number of data dwords or SATA_EOFs for the SATA frame defined in 6.19.2.3 into its STP flow control buffer and receives SATA_HOLD_A within that number of data dwords or SATA_EOFs. While receiving SATA_HOLD_A, the STP phy or expander phy does not place any data dwords into the STP flow control buffer. The STP phy or expander phy shall stop transmitting SATA_HOLD when the STP flow control buffer empties enough to hold at least that number of data dwords or SATA_EOFs.

When an STP phy or expander phy during an STP connection is transmitting a SATA frame and receives SATA_HOLD, that STP phy or expander phy shall transmit no more than 20 data dwords or SATA_EOFs for the SATA frame and respond with SATA_HOLD_A.

When a SATA host phy in an STP/SATA bridge is receiving a SATA frame from a SATA physical link, that SATA host phy shall transmit a SATA_HOLD when it is only capable of receiving 21 more data dwords. The SATA host phy shall stop transmitting SATA_HOLD (e.g., return to transmitting SATA_R_IP) when it is capable of receiving at least 21 more data dwords.

NOTE 56 SATA requires that frame transmission cease and SATA_HOLD_A be transmitted within 20 data dwords of receiving SATA_HOLD. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

When a SATA host phy in an STP/SATA bridge is transmitting a SATA frame to a SATA physical link, that SATA host phy shall transmit no more than 19 dwords (e.g., including data dwords, deletable primitives, SATA_EOF, and SATA_HOLDs followed by data dwords) after receiving SATA_HOLD before responding with SATA_HOLD_A.

NOTE 57 SATA assumes that once a SATA_HOLD is transmitted, frame transmission ceases and SATA_HOLD_A arrives within 20 dwords. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

While transmitting SATA_HOLD or SATA_HOLD_A, the expander device is considered to be originating (see 6.3.2) rather than forwarding (see 6.3.3) for purposes of deletable primitive insertion.

6.18.2.3 STP flow control buffer size

All logical phys in a port shall support the same STP flow control buffer size. The STP flow control buffer size is determined for each connector category (see SAS-2.1) to which the phy is attached as follows:

- a) for the unmanaged passive connector category (see SAS-2.1) the STP flow control buffer size shall be at least:
 - A) 24 data dwords or SATA_EOFs at the 1.5 Gbps logical link rate;
 - B) 28 data dwords or SATA_EOFs at the 3 Gbps logical link rate; or
 - C) 36 data dwords or SATA_EOFs at the 6 Gbps logical link rate;

NOTE 58 - For connectors in the unmanaged connector category the STP flow control buffer requirements are based on $(20 + (4 \times 2^n))$ where n is zero for 1.5 Gbps, one for 3 Gbps, and two for 6 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see SATA). The (4×2^n) portion of this equation is based on:

- a) one-way propagation time on a 10 m cable = 53 ns (see SAS-2.1);
 - b) round-trip propagation time on a 10 m cable = 106 ns (e.g., time to send SATA_HOLD and receive SATA_HOLD_A);
 - c) time to transmit a 1.5 Gbps dword = $(0.6 \text{ ns/bit unit interval (see SAS-2.1)}) \times (40 \text{ bits/dword}) = 26.6 \text{ ns}$;
 - d) number of 1.5 Gbps dwords on the wire during round-trip propagation time = $(106 \text{ ns} / 26.6 \text{ ns}) = 3.98$ data dwords; and
 - e) rounding to 4 data dwords.
- b) for the unmanaged active connector category (see SAS-2.1) the STP flow control buffer size shall be at least:
 - A) 30 data dwords or SATA_EOFs at the 1.5 Gbps logical link rate;
 - B) 40 data dwords or SATA_EOFs at the 3 Gbps logical link rate; or
 - C) 60 data dwords or SATA_EOFs at the 6 Gbps logical link rate;

NOTE 59 - For connectors in the unmanaged active connector category the STP flow control buffer requirements are based on $(20 + (9.375 \times 2^n))$ where n is zero for 1.5 Gbps, one for 3 Gbps, and two for 6 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see SATA). The (9.375×2^n) portion of this equation is based on:

- a) one-way propagation time on a 25 m cable = 133 ns (see SAS-2.1);
 - b) round-trip propagation time on a 25 m cable = 266 ns (e.g., time to send SATA_HOLD and receive SATA_HOLD_A);
 - c) time to transmit a 1.5 Gbps dword = $(0.6 \text{ ns/bit unit interval (see SAS-2.1)}) \times (40 \text{ bits/dword}) = 26.6 \text{ ns}$; and
 - d) number of 1.5 Gbps dwords on the wire during round-trip propagation time = $(266 \text{ ns} / 26.6 \text{ ns}) = 10$ data dwords.
- c) for the managed connector category (see SAS-2.1) the total cable assembly propagation delay is reported through the management protocol (see SAS-2.1). If the propagation delay is less than or equal to 53 ns, then the STP flow control buffer size is 24 data dwords, 28 data dwords, or 36 data dwords depending on the logical link rate (see a)). If the propagation delay exceeds 53 ns then the minimum STP flow control buffer size in dwords shall be calculated using the following equation:

$$\text{Minimum STP flow control buffer size} = (20 + ((2 \times P_d \times R) / 40)),$$

where:

P_d = Propagation delay of cable assembly (e.g., in ns) (see SAS-2.1); and

R = Nominal logical link rate (e.g., in Gbps).

If the minimum buffer size is not an integer, then the minimum buffer size value shall be rounded to the next highest integer value.

NOTE 60 - For connectors in the managed active connector category where the cable reports total propagation delay of 250 ns and connection rate at 6 Gbps then the STP flow control buffer requirements are based on Minimum buffer size = $[20 + ((2 \times P_d \times R) / 40)]$. The 20 portion of this equation is based on

the frame transmitter requirements (see SATA). The $[(2 \times P_d \times R) / 40]$ portion of this equation is the number of dwords on the wire during a round trip: $2 \times P_d$ is the round trip propagation time, and $40 / R$ is the time to transmit a dword. For example, for a 100 m optical cable:

- a) one-way propagation time = 500 ns;
- b) round-trip propagation time = 1 000 ns (e.g., time to send SATA_HOLD and receive SATA_HOLD_A);
- c) time to transmit a 6 Gbps dword = $(0.16 \text{ ns/bit unit interval}) \times (40 \text{ bits/dword}) = 6.6 \text{ ns}$;
- d) number of 6 Gbps dwords on the wire during round-trip propagation time = $(1\,000 \text{ ns} / 6.6 \text{ ns}) = 150$ data dwords; and
- e) 20 is the buffer size required by SATA, resulting in a minimum STP flow control buffer size = 170 data dwords.

6.18.2.4 STP flow control example

Figure 125 shows STP flow control between:

- a) an STP initiator phy receiving a frame;
- b) an expander device (the first expander device);
- c) an expander device with an STP/SATA bridge (the second expander device); and
- d) a SATA device phy transmitting a frame.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

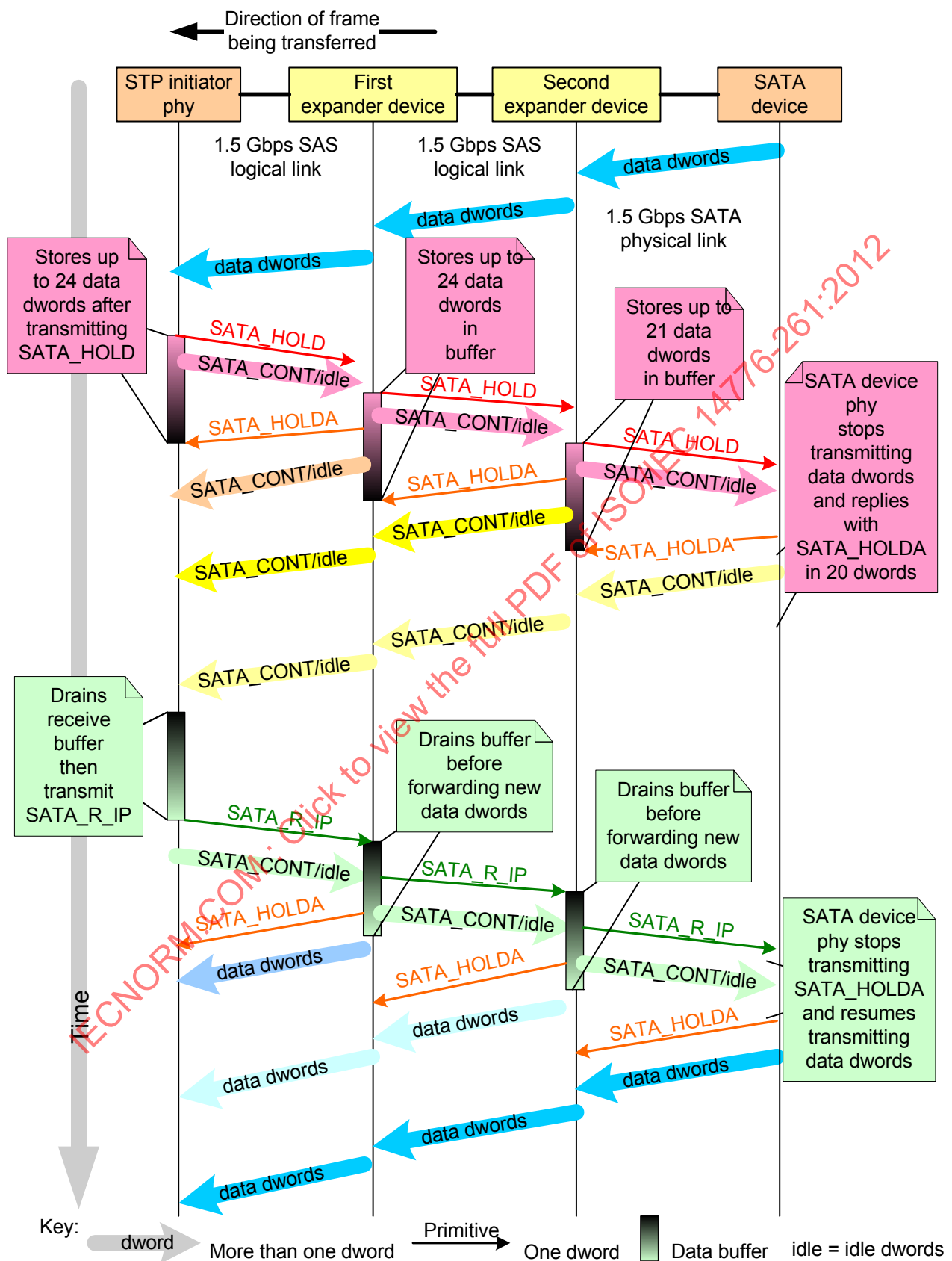


Figure 125 — STP flow control

After the STP initiator phy transmits SATA_HOLD, it receives a SATA_HOLDA reply from the first expander device within 24 dwords. The first expander device transmits SATA_HOLD to the second expander device and receives SATA_HOLDA within 24 dwords, buffering data dwords in the STP flow control buffer that the first expander device is no longer able to forward to the STP initiator phy. The second expander device transmits SATA_HOLD to the SATA device phy and receives SATA_HOLDA within 21 dwords, buffering data dwords in the STP flow control buffer that it is no longer able to forward to the first expander device. When the SATA device phy stops transmitting data dwords, its previous data dwords are stored in the STP flow control buffers in both expander devices and the STP initiator phy.

After the STP initiator phy drains its STP flow control buffer and transmits SATA_R_IP, it receives data dwords from the first expander device's STP flow control buffer, followed by data dwords from the second expander device's STP flow control buffer, followed by data dwords from the SATA device phy.

6.18.2.5 STP insufficient buffer support

If an STP phy is attached to a connector in the managed connector category (see SAS- 2.1) and that STP phy does not support the minimum buffer size for the logical link rate, then all phys in the STP port shall perform a link reset sequence indicating the STP protocol is not supported with:

- a) the STP INITIATOR PORT bit in the IDENTIFY address frame set to zero (see 6.8.2); and
- b) the STP TARGET PORT bit in the IDENTIFY address frame set to zero (see 6.8.2).

An STP initiator port shall not originate an STP connection request to a destination STP target port on a pathway that contains an expander device reporting insufficient buffer for STP support.

If an expander phy is attached to a connector in the managed connector category (see SAS-2.1) and that expander phy does not support the minimum buffer size for the logical link rate, then all phys in the expander port shall perform a link reset sequence indicating the STP protocol is not supported with:

- a) the STP BUFFER TOO SMALL bit in the SMP DISCOVER response set to one (see 9.4.3.10); and
- b) the STP BUFFER TOO SMALL bit in the SMP DISCOVER LIST response set to one (see 9.4.3.15).

If an expander phy does not support the minimum buffer size for the logical link rate, then that expander phy shall respond to an STP connection request with OPEN_REJECT (PROTOCOL NOT SUPPORTED).

6.18.3 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be:

- 1) transmitted two times;
- 2) be followed by SATA_CONT, if needed; and
- 3) be followed by vendor specific scrambled data dwords, if needed.

Deletable primitives may be transmitted inside continued primitive sequences as described in 6.2.4.1.

After the SATA_CONT, during the vendor specific scrambled data dwords:

- a) a SATA_CONT continues the continued primitive sequence; and
- b) any other STP primitive, including the primitive that is being continued, ends the continued primitive sequence.

Figure 126 shows an example of transmitting a continued primitive sequence.

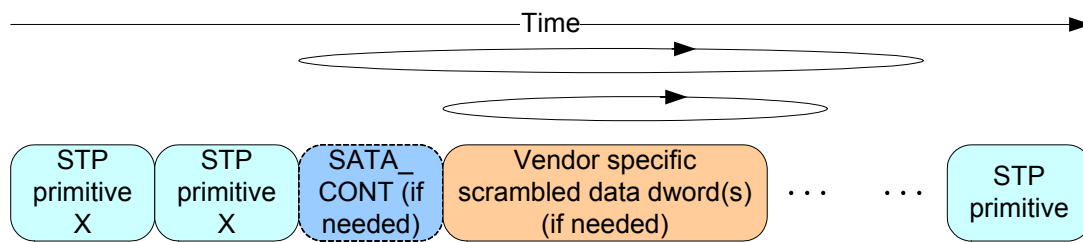


Figure 126 — Transmitting a continued primitive sequence

Receivers shall detect a continued primitive sequence after at least one primitive is received. The primitive may be followed by one or more of the same primitive. The primitive may be followed by one or more SATA_CONTs, each of which may be followed by vendor specific data dwords. Receivers shall ignore invalid dwords before, during, or after the SATA_CONT(s). Receivers do not count the number of times the continued primitive, the SATA_CONTs, or the vendor specific data dwords are received (i.e., receivers are in the state of receiving the primitive).

Expanders forwarding dwords may or may not detect an incoming sequence of the same primitive and convert the incoming sequence into a continued primitive sequence.

Figure 127 shows an example of receiving a continued primitive sequence.

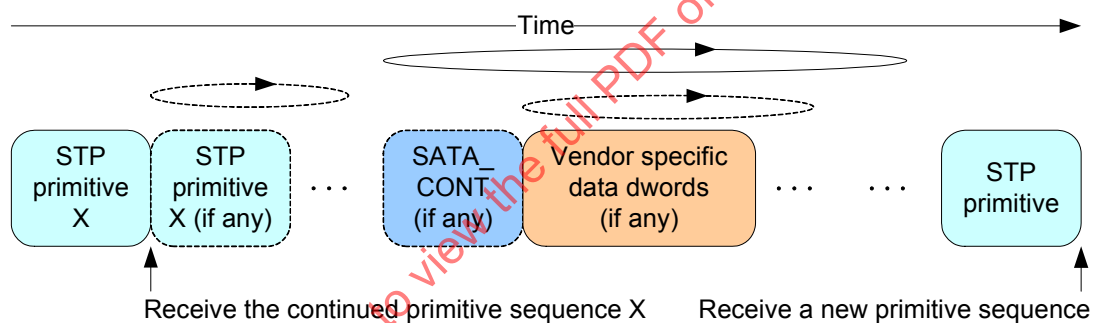


Figure 127 — Receiving a continued primitive sequence

An expander device forwarding a continued primitive sequence may transmit more dwords in the continued primitive sequence than it receives (i.e., expand) or transmit fewer dwords in the continued primitive sequence than it receives (i.e., contract). While transmitting a continued primitive sequence, the expander device is considered to be originating (see 6.3.2) rather than forwarding (see 6.3.3) for purposes of deletable primitive insertion.

6.18.4 Affiliations

The STP target port shall provide coherent access to a set of registers called an affiliation context for each STP initiator port from which the STP target port accepts connections. An affiliation is a state entered by an STP target port in which the STP target port refuses to accept connection requests from STP initiator ports other than those that have established an affiliation.

An STP target port shall implement one of the affiliation policies defined in table 111.

Table 111 — Affiliation policies

Affiliation policy	Description
No affiliations	An unlimited number of STP initiator ports are allowed to access the STP target port concurrently. The STP target port is cognizant of the SAS address of the STP initiator port that sends each ATA command.
Multiple affiliations	The STP target port implements more than one affiliation, so a limited number of STP initiator ports are allowed to access the STP target port concurrently. The STP target port implements no more than one affiliation context per STP initiator port.
Single affiliation	The STP target port implements one affiliation, so one STP initiator port is allowed to access the STP target port at a time.

An STP/SATA bridge that supports either no affiliations or multiple affiliations shall:

- ensure that the SATA NCQ tags in commands issued to the SATA device are unique across all affiliations;
- ensure that a non-queued command received in one affiliation context is not issued to the SATA device while another affiliation context has a queued command outstanding to the SATA device (e.g., the STP target port shall allow all queued commands in the SATA device to complete prior to issuing the non-queued command);
- ensure that a non-queued command received in one affiliation context is not issued to the SATA device while another affiliation context has a non-queued command outstanding to the SATA device (e.g., the STP target port shall allow the non-queued command in the SATA device to complete prior to issuing the new non-queued command); and
- ensure that a queued command received in one affiliation context is not issued to the SATA device while another affiliation context has a non-queued command outstanding to the SATA device (e.g., the STP target port shall allow any non-queued command in the SATA device to complete prior to issuing the queued commands).

An STP/SATA bridge that supports multiple affiliations may modify the queue depth reported in the ATA IDENTIFY DEVICE data (see ACS-2) to each STP initiator port to ensure that all the STP initiator ports with affiliations do not send more commands than the SATA device supports.

An STP target port that supports affiliations shall establish an affiliation whenever it accepts a connection request from an STP initiator port that does not already have an affiliation. When the maximum number of affiliations have been established (i.e., all affiliation contexts are in use), the STP target port shall reject all subsequent connection requests from other STP initiator ports with OPEN_REJECT (STP RESOURCES BUSY).

An STP target port shall maintain an affiliation until any of the following occurs:

- power on;
- the management device server receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of HARD RESET (see 9.4.3.28) from any SMP initiator port;
- the management device server receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of TRANSMIT SATA PORT SELECTION SIGNAL (see 9.4.3.28) from any SMP initiator port;
- the management device server receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of CLEAR AFFILIATION (see 9.4.3.28) from the same SAS initiator port that has the affiliation.

If a connection is already established to the STP target port on one phy while an SMP PHY CONTROL request specifying a phy operation of CLEAR AFFILIATION is processed by an SMP target port on another phy, then the affiliation shall be cleared and the STP target port shall respond to new connection attempts with:

- A) an AIP (WAITING ON CONNECTION) primitive sequence and/or OPEN_REJECT (RETRY), if the STP target port is in an expander device; or
- B) OPEN_REJECT (RETRY), if the STP target port is in a SAS device, rather than OPEN_REJECT (STP RESOURCES BUSY);
- e) an STP connection to a phy in the STP target port is closed with a CLOSE (CLEAR AFFILIATION) primitive sequence; or
- f) the STP target port encounters an I_T nexus loss.

The STP initiator port shall maintain an affiliation starting with the connection in which a command is transmitted until all frames for the command have been delivered. An STP initiator port implementing command queuing (see ACS-2 and SATA) shall maintain an affiliation while any commands are outstanding. STP initiator ports should not keep affiliations while commands are not outstanding.

An STP target port that implements affiliations shall implement at least one affiliation context per STP target port. Multiple phys on the same STP target port shall use the same set of affiliation contexts. Support for affiliations is indicated in the SMP REPORT PHY SATA response (see 9.4.3.12).

An STP target port implementing multiple affiliations shall sort the affiliation contexts in a vendor specific order. In the SMP REPORT PHY SATA response, if the SMP initiator port has the same SAS address as an affiliated STP initiator port, then the management device server shall report the affiliation for that SAS address as relative identifier 0 and shall report all additional affiliations with incrementing relative identifiers following the sorted order. If the SMP initiator port does not have the same SAS address as an affiliated STP initiator port, the management device server shall report the affiliation contexts in the vendor specific order.

For example, if the STP target port supports four affiliation contexts sorted in order A, B, C, and D, when returning the SMP REPORT PHY SATA response to an SMP initiator port, then the management device server reports the affiliation contexts as described in table 112.

Table 112 — Affiliation context relative identifier example

Affiliation context containing the SAS address of the SMP initiator port	Affiliation context relative identifier assignment			
	0	1	2	3
A	A	B	C	D
B	B	C	D	A
C	C	D	A	B
D	D	A	B	C
None	A	B	C	D

6.18.5 Opening an STP connection

When the SATA host port in an STP/SATA bridge receives a SATA_X_RDY from the attached SATA device, the STP target port in the STP/SATA bridge shall establish an STP connection to the appropriate STP initiator port. If there is no affiliation, the SATA host port may either:

- a) perform a link reset on the SATA physical link; or
- b) wait for an affiliation to be established.

If an STP/SATA bridge receives a connection request for a SATA device that has delivered the initial Register – Device to Host FIS in error, then it shall return an OPEN_REJECT (NO DESTINATION).

If there is a problem receiving the expected initial Register - Device to Host FIS, then the STP/SATA bridge should use SATA_R_ERR to retry until the FIS is received without error. In the DISCOVER response, the ATTACHED SATA DEVICE bit is set to one and the ATTACHED SAS ADDRESS field is valid, but the ATTACHED DEVICE TYPE field is set to 000b (i.e., no device attached) during this time.

If an STP/SATA bridge that retrieves IDENTIFY (PACKET) DEVICE data receives a connection request for a SATA device before it has retrieved the IDENTIFY (PACKET) DEVICE data, then it shall return an OPEN_REJECT (NO DESTINATION). If the STP/SATA bridge has a problem retrieving the IDENTIFY (PACKET) DEVICE data (e.g., word 255 (i.e., the Integrity Word) is not correct), then it shall set the ATTACHED DEVICE NAME field to zero, set the ATTACHED DEVICE TYPE field to 001b (i.e, end device), and start accepting connections.

A wide STP initiator port shall not request more than one connection at a time to a specific STP target port.

While a wide STP initiator port is waiting for a response to a connection request to an STP target port, a SAS phy in the STP initiator port shall not reject an incoming connection request from that STP target port with OPEN_REJECT (RETRY) because the SAS port containing that SAS phy needs an outgoing connection request to be accepted. The SAS phy may reject an incoming connection request from that STP target port with OPEN_REJECT (RETRY) for any reason that is not dependent on the SAS port containing that SAS phy having an outgoing connection request accepted (e.g., because of a temporary buffer full condition).

If a wide STP initiator port receives an incoming connection request from an STP target port while it has a connection established with that STP target port, then the wide STP initiator port shall reject the request with OPEN_REJECT (RETRY).

A wide STP target port shall not request more than one connection at a time to a specific STP initiator port.

While a wide STP target port is waiting for a response to a connection request or has established a connection to an STP initiator port, the wide STP target port shall:

- a) reject incoming connection requests from that STP initiator port with OPEN_REJECT (RETRY); and
- b) if affiliations are supported and the maximum number of affiliations has been established (i.e., all affiliation contexts are in use), reject incoming connection requests from other STP initiator ports that do not have affiliations with OPEN_REJECT (STP RESOURCES BUSY).

A SAS phy may reject an incoming connection request (i.e., an OPEN address frame) to an STP target port with OPEN_REJECT (RETRY) for any reason, including because the SAS port containing that SAS phy needs an outgoing connection request to be accepted (e.g., to transmit a frame and empty a buffer).

An expander device should not allow its STP ports (e.g., the STP target ports in STP/SATA bridges and any STP initiator ports in the expander device) to attempt to establish more connections to a specific destination port than the destination port width or the width of the narrowest physical link on the pathway to the destination port. This does not apply to connection requests being forwarded by the expander device.

An expander device should not allow its STP ports (e.g., the STP target ports in STP/SATA bridges and any STP initiator ports in the expander device) to attempt to establish more connections than the width of the narrowest common physical link on the pathways to the destination ports of those connections. This does not apply to connection requests being forwarded by the expander device.

Figure 128 shows an example of the simultaneous connection recommendations for an expander device containing STP ports. Multiplexing is disabled in this example.

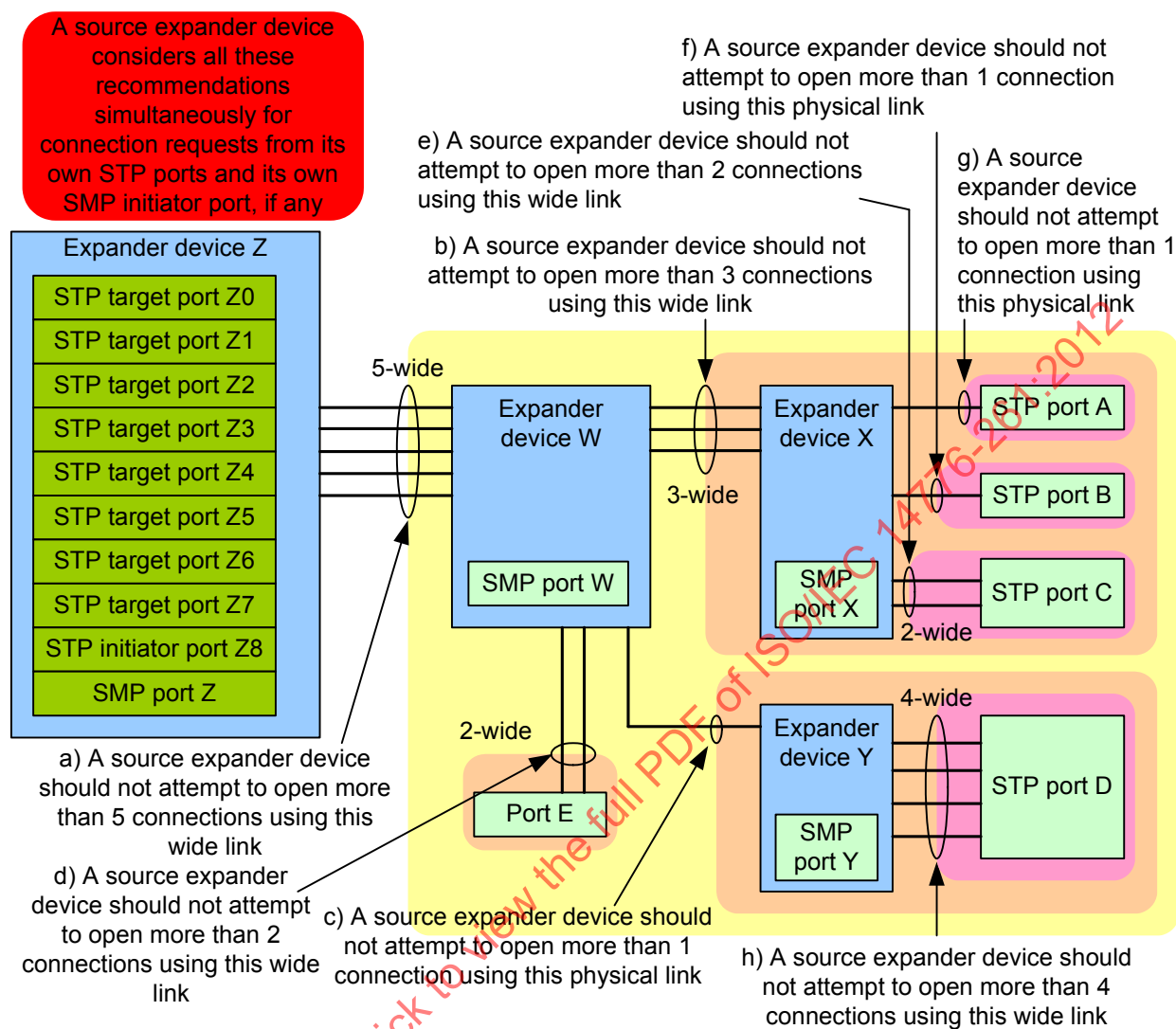


Figure 128 — Example simultaneous connection recommendations for an expander device

In figure 128, some of the recommendations are combined as follows:

- recommendations a), b), and e) together specify that expander device Z should not attempt to open more than 2 connections to port C;
- recommendations a), b), e), f), and g) together specify that if expander device Z has 2 connections open to ports A, B, and X, it should not attempt to open more than 1 connection to port C. If it has 4 connections open to ports A, B, D, E, W, X, and Y, it should not attempt to open more than 1 connection to port C; and
- recommendations a), c), and h) together specify that expander device Z should not attempt to open more than 1 connection to port D. If it has a connection open to port Y, it should not attempt to open another connection to port D until the first connection is closed.

The first dword that an STP phy sends inside an STP connection after OPEN_ACCEPT that is not a deletable primitive shall be an STP primitive (e.g., SATA_SYNC).

6.18.6 Closing an STP connection

Either STP port (i.e., either the STP initiator port or the STP target port) may originate closing an STP connection. An STP port shall not originate closing an STP connection after sending a SATA_X_RDY or SATA_R_RDY until after both sending and receiving SATA_SYNC. An STP port shall transmit a CLOSE

primitive sequence after receiving a CLOSE primitive sequence if it has not already transmitted a CLOSE primitive sequence.

If an STP port receives a CLOSE primitive sequence after transmitting a SATA_X_RDY but before receiving a SATA_R_RDY, then the STP port shall complete closing the connection (i.e., transmit a CLOSE primitive sequence) and retransmit the SATA_X_RDY in a new connection.

When an STP initiator port closes an STP connection, it shall transmit a CLOSE (NORMAL) primitive sequence or CLOSE (CLEAR AFFILIATION) primitive sequence. When an STP target port closes an STP connection, it shall transmit a CLOSE (NORMAL) primitive sequence.

An STP initiator port may issue a CLOSE (CLEAR AFFILIATION) primitive sequence in place of a CLOSE (NORMAL) primitive sequence to cause the STP target port to clear the affiliation (see 6.18.4) along with closing the connection. If an STP target port receives a CLOSE (CLEAR AFFILIATION) primitive sequence, then the STP target port shall clear the affiliation for the STP initiator port from which the a CLOSE (CLEAR AFFILIATION) primitive sequence was received.

See 6.13.7 for additional details on closing connections.

An STP/SATA bridge shall break an STP connection if its SATA host phy loses dword synchronization (see 6.13.8).

6.18.7 STP connection management examples

The STP/SATA bridge adds the outgoing OPEN address frames and CLOSEs so the STP initiator port sees an STP target port. The STP/SATA bridge removes incoming OPEN address frame and CLOSEs so the SATA device port sees only a SATA host port. While the connection is open, the STP/SATA bridge passes through all dwords without modification. Both STP initiator port and STP target port use SATA, with SATA flow control (see 6.18.2), while the connection is open.

Figure 129 shows an STP initiator port opening a connection, transmitting a single SATA frame, and closing the connection.

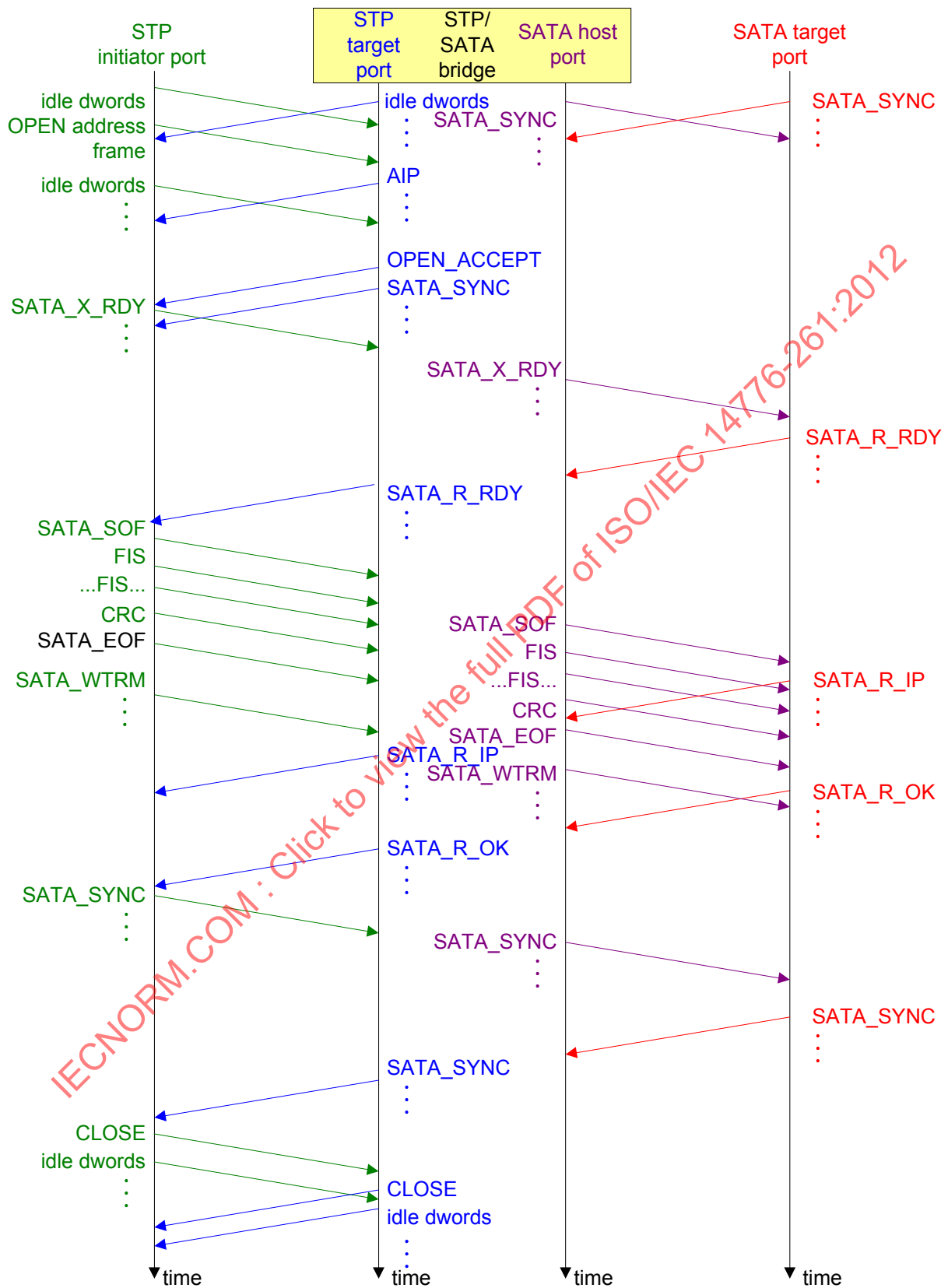


Figure 129 — STP initiator port opening an STP connection

Figure 130 shows a SATA device transmitting a SATA frame. In this example, the STP target port in the STP/SATA bridge opens a connection to an STP initiator port to send just one frame, then closes the connection.

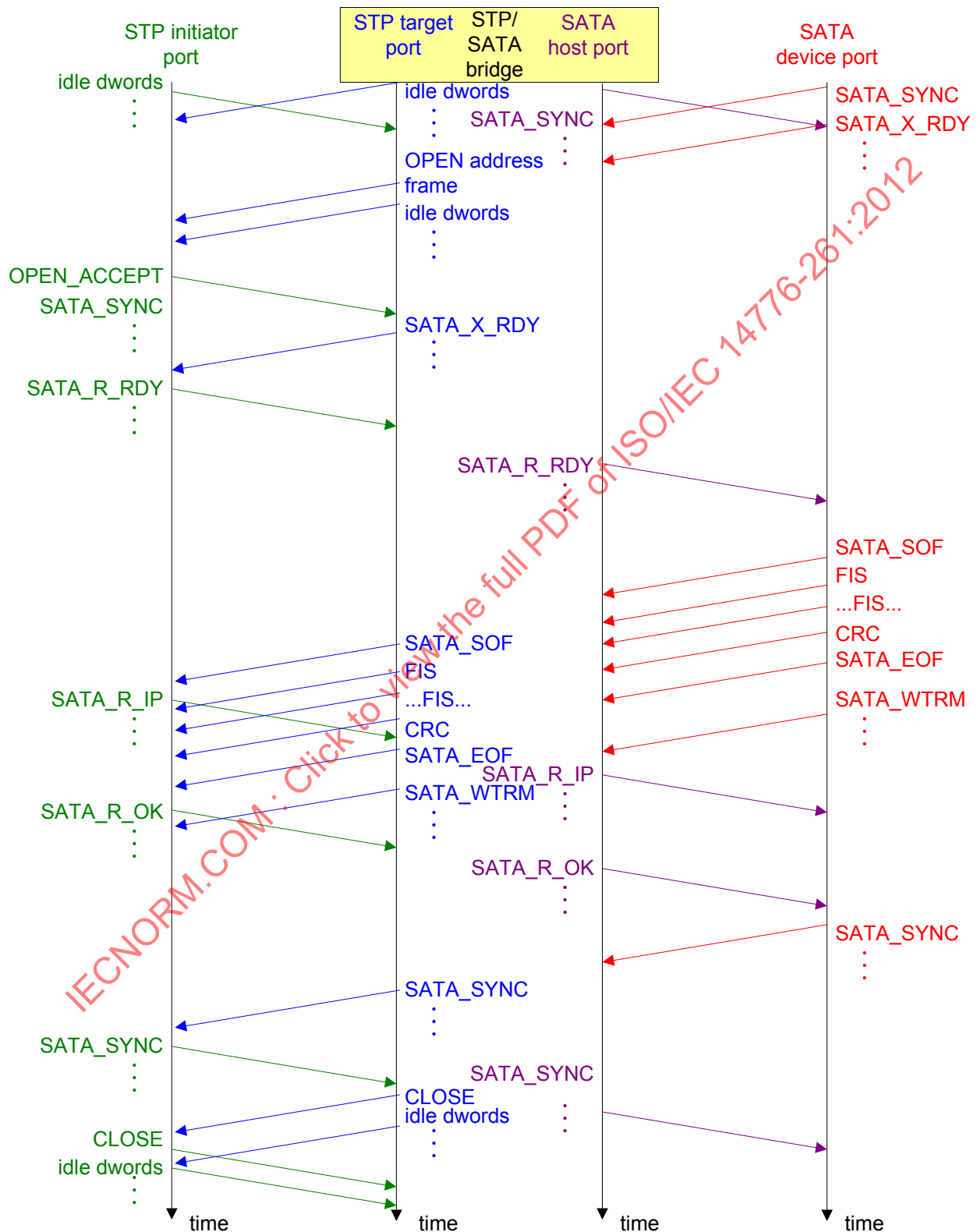


Figure 130 — STP target port opening an STP connection

6.18.8 STP (link layer for STP phys) state machines

The STP link layer uses the SATA link layer state machines (see SATA), modified to:

- a) communicate with the port layer rather than directly with the transport layer;
- b) interface with the SL state machines for connection management (e.g., to select when to open and close STP connections, and to tolerate idle dwords between an OPEN address frame and the first SATA primitive);
- c) communicate with an STP transmitter and receiver; and
- d) support an affiliation policy (see 6.18.4).

These modifications are not described in this standard.

The STP transmitter relationship to other transmitters is defined in 4.3.2. The STP receiver relationship to other receivers is defined in 4.3.3.

6.18.9 SMP target port support

A SAS device that contains an STP target port shall also contain an SMP target port.

6.19 SMP link layer

6.19.1 SMP frame transmission and reception

Inside an SMP connection, the SMP initiator phy transmits a single SMP_REQUEST frame within 100 µs and the SMP target phy responds with a single SMP_RESPONSE frame (see 8.4) within 1 900 µs.

Frames are surrounded by SOF and EOF as shown in figure 131. See 6.19.5 for error handling details.

NOTE 61 Unlike SSP, there is no acknowledgement of SMP frames with ACK and NAK and there is no credit exchange with RRDY.

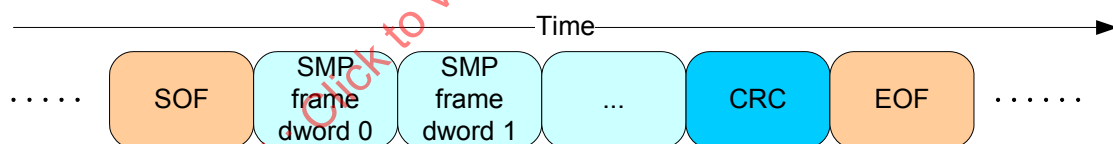


Figure 131 — SMP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 6.5). The SMP link layer state machine checks that the frame is not too short and that the CRC is valid (see 6.19.5).

6.19.2 SMP flow control

By accepting an SMP connection, the SMP target phy indicates it is ready to receive one SMP_REQUEST frame.

After the SMP initiator phy transmits one SMP_REQUEST frame, it shall be ready to receive one SMP_RESPONSE frame.

6.19.3 Opening an SMP connection

An SMP target port shall not attempt to establish an SMP connection.

A SAS phy may reject an incoming connection request (i.e., OPEN address frame) to an SMP target port with OPEN_REJECT (RETRY) for any reason, including because the SAS port containing that SAS phy needs an outgoing connection request to be accepted (e.g., to transmit a frame and empty a buffer).

6.19.4 Closing an SMP connection

After receiving the SMP_RESPONSE frame, the SMP initiator phy shall transmit a CLOSE (NORMAL) primitive sequence to close the connection.

After transmitting the SMP_RESPONSE frame, the SMP target phy shall reply with a CLOSE (NORMAL) primitive sequence.

See 6.13.7 for additional details on closing connections.

6.19.5 SMP (link layer for SMP phys) state machines

6.19.5.1 SMP state machines overview

The SMP state machines control the flow of dwords on the physical link during an SMP connection. The SMP state machines are as follows:

- a) SMP_IP (link layer for SMP initiator phys) state machine (see 6.19.5.3); and
- b) SMP_TP (link layer for SMP target phys) state machine (see 6.19.5.4).

6.19.5.2 SMP transmitter and receiver

The SMP transmitter receives the following messages from the SMP state machines specifying dwords and frames to transmit:

- a) Transmit Idle Dword; and
- b) Transmit Frame with an argument containing the frame contents.

In response to the Transmit Frame message, the SMP transmitter transmits:

- 1) SOF;
- 2) the frame contents;
- 3) CRC; and
- 4) EOF.

The SMP transmitter sends the following message to the SMP state machines based on dwords that have been transmitted:

- a) Frame Transmitted.

When there is no outstanding message specifying a dword to transmit, the SMP transmitter shall transmit idle dwords.

The SMP receiver sends the following messages to the SMP state machines indicating primitive sequences and dwords received from the SP_DWS receiver (see 5.10.2):

- a) SOF Received;
- b) Data Dword Received;
- c) EOF Received;
- d) ERROR Received; and
- e) Invalid Dword Received.

The SMP receiver shall ignore all other dwords.

The SMP transmitter relationship to other transmitters is defined in 4.3.2. The SMP receiver relationship to other receivers is defined in 4.3.3.

6.19.5.3 SMP_IP (link layer for SMP initiator phys) state machine

6.19.5.3.1 SMP_IP state machine overview

The SMP_IP state machine's function is to transmit an SMP request frame and then receive the corresponding response frame. This state machine consists of the following states:

- a) SMP_IP1:Idle (see 6.19.5.3.2) (initial state);

- b) SMP_IP2:Transmit_Frame (see 6.19.5.3.3); and
- c) SMP_IP3:Receive_Frame (see 6.19.5.3.4).

This state machine shall start in the SMP_IP1:Idle state on receipt of an Enable Disable SMP (Enable) message from the SL state machines (see 6.15).

The SMP_IP state machine shall terminate after receiving an Enable Disable SMP (Disable) message from the SL state machines.

Figure 132 shows the SMP_IP state machine.

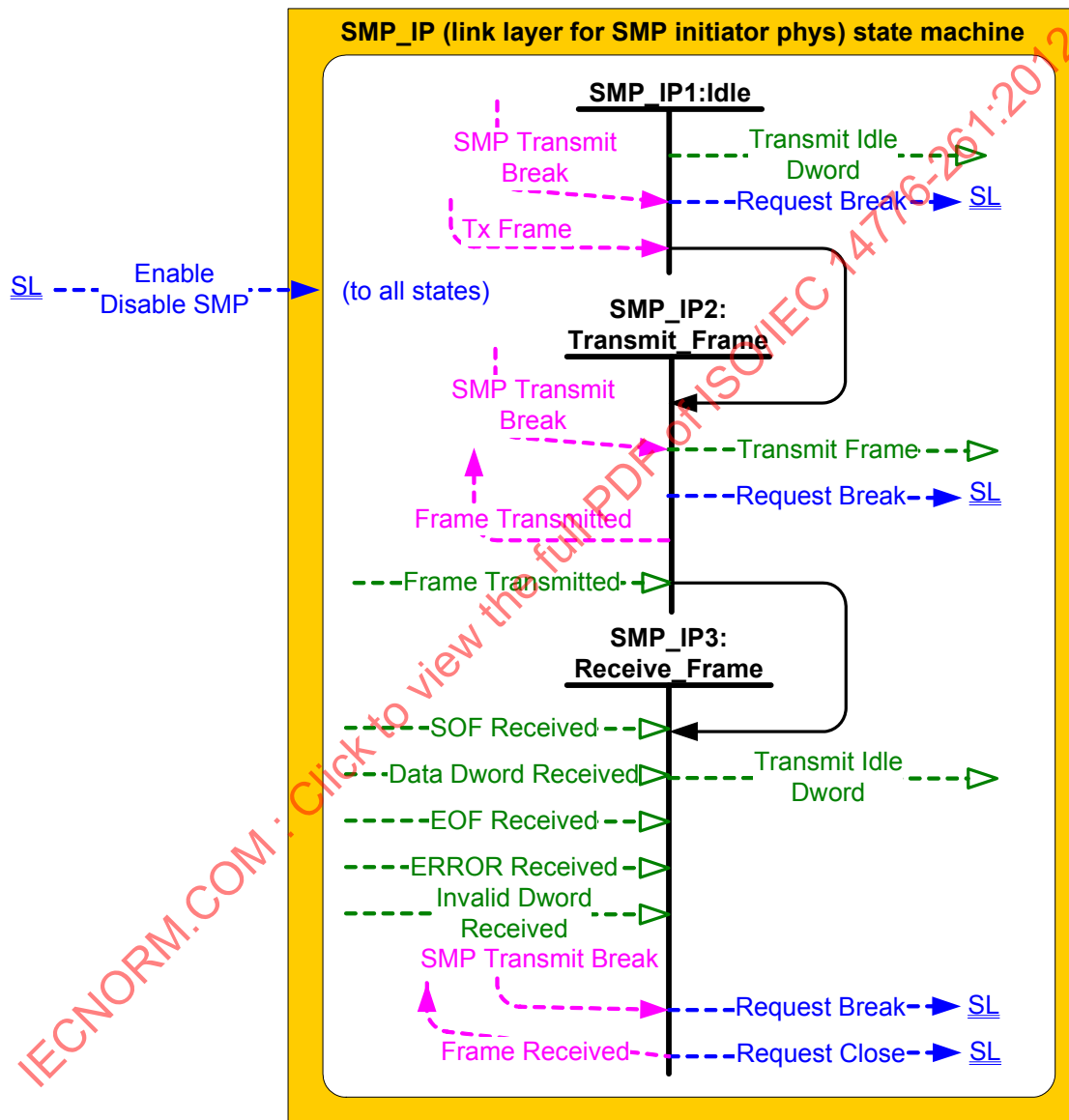


Figure 132 — SMP_IP (link layer for SMP initiator phys) state machine

6.19.5.3.2 SMP_IP1:Idle state

6.19.5.3.2.1 State description

This state is the initial state.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

If an SMP Transmit Break request is received, then this state shall send a Request Break message to the SL state machines (see 6.15).

6.19.5.3.2 Transition SMP_IP1:Idle to SMP_IP2:Transmit_Frame

This transition shall occur after a Tx Frame request is received.

6.19.5.3.3 SMP_IP2:Transmit_Frame state

6.19.5.3.3.1 State description

This state shall send a Transmit Frame message to the SMP transmitter with an argument containing the frame contents.

If an SMP Transmit Break request is received, then this state shall send a Request Break message to the SL state machines (see 6.15) and terminate this state machine.

After the Frame Transmitted message is received, this state shall send a Frame Transmitted confirmation to the port layer.

6.19.5.3.3.2 Transition SMP_IP2:Transmit_Frame to SMP_IP3:Receive_Frame

This transition shall occur after sending a Frame Transmitted confirmation to the port layer.

6.19.5.3.4 SMP_IP3:Receive_Frame state

This state checks the SMP response frame and determines if the SMP response frame was received without error (e.g., no CRC error).

If this state receives a subsequent SOF Received message after receiving an SOF Received message but before receiving an EOF Received message (i.e., SOF, data dwords, SOF, data dwords, and EOF instead of SOF, data dwords, EOF, SOF, data dwords, and EOF), then this state shall discard the frame in progress.

This state shall discard the frame, send a Frame Received (SMP Unsuccessful) confirmation to the port layer, send a Request Break message to the SL state machines, and terminate this state machine if:

- a) this state receives more than 257 Data Dword Received messages (i.e., 1 028 bytes) after an SOF Received message and before an EOF Received message; or

NOTE 62 SMP target phys compliant with previous versions of the standard may send vendor specific SMP frames containing 258 data dwords (i.e., 1 032 bytes).

- b) this state receives fewer than 2 Data Dword Received messages (i.e., 8 bytes) after an SOF Received message and before an EOF Received message.

If this state receives an Invalid Dword Received message or an ERROR Received message after an SOF Received message and before an EOF Received message, then this state machine shall either:

- a) ignore the invalid dword or ERROR; or
- b) discard the frame, send a Frame Received (SMP Unsuccessful) confirmation to the port layer, send a Request Break message to the SL state machines, and terminate this state machine.

If the SMP response frame is received with a CRC error, then this state shall discard the frame, send a Frame Received (SMP Unsuccessful) confirmation to the port layer, send a Request Break message to the SL state machines, and terminate this state machine.

If the SMP response frame is received with no CRC error and the SMP response frame is valid, then this state shall:

- a) send a Frame Received (SMP Successful) confirmation to the port layer; and
- b) send a Request Close message to the SL state machines (see 6.15).

If an SMP Transmit Break request is received, then this state shall send a Request Break message to the SL state machines and terminate this state machine.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

6.19.5.4 SMP_TP (link layer for SMP target phys) state machine

6.19.5.4.1 SMP_TP state machine overview

The SMP_TP state machine's function is to receive an SMP request frame and then transmit the corresponding SMP response frame. The SMP_TP state machine consists of the following states:

- SMP_TP1:Receive_Frame (see 6.19.5.4.2) (initial state); and
- SMP_TP2:Transmit_Frame (see 6.19.5.4.3).

This state machine shall start in the SMP_TP1:Receive_Frame state after receiving an Enable Disable SMP (Enable) message from the SL state machines (see 6.15).

This state machine shall terminate after receiving an Enable Disable SMP (Disable) message from the SL state machines.

Figure 133 shows the SMP_TP state machine.

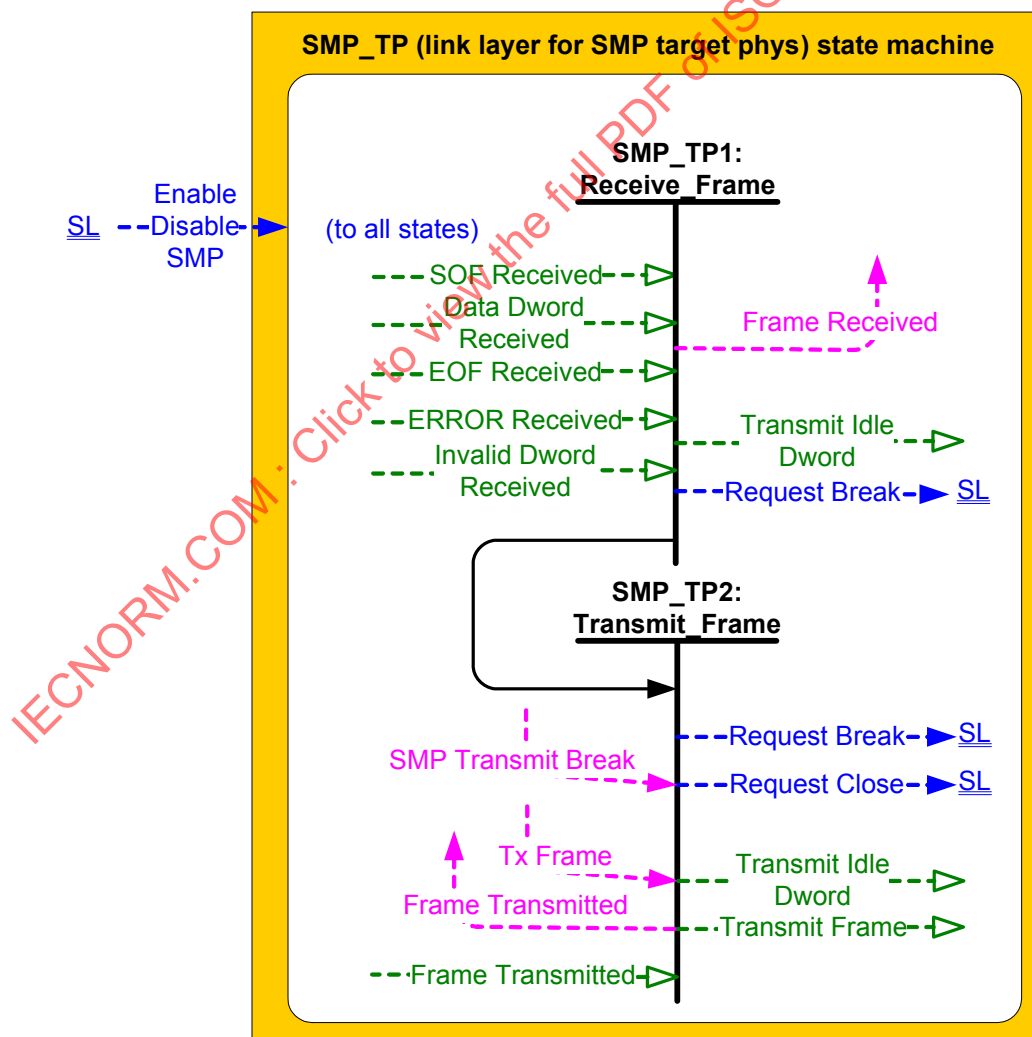


Figure 133 — SMP_TP (link layer for SMP target phys) state machine

6.19.5.4.2 SMP_TP1:Receive_Frame state

6.19.5.4.2.1 State description

This state waits for an SMP frame and determines if the SMP frame was received without error (e.g., no CRC error).

If this state receives a subsequent SOF Received message after receiving an SOF Received message but before receiving an EOF Received message (i.e., SOF, data dwords, SOF, data dwords, and EOF instead of SOF, data dwords, EOF, SOF, data dwords, and EOF), then this state shall discard the frame in progress.

This state shall discard the frame, send a Request Break message to the SL state machines (see 6.15) and shall terminate this state machine if:

- a) this state receives more than 257 Data Dword Received messages (i.e., 1 028 bytes) after an SOF Received message and before an EOF Received message; or

NOTE 63 SMP initiator phys compliant with previous versions of the standard may send vendor specific SMP frames containing 258 data dwords (i.e., 1 032 bytes).

- b) this state receives fewer than 2 Data Dword Received messages (i.e., 8 bytes) after an SOF Received message and before an EOF Received message.

If this state receives an Invalid Dword Received message or an ERROR Received message after an SOF Received message and before an EOF Received message, then this state machine shall either:

- a) ignore the invalid dword or ERROR; or
- b) discard the frame, send a Request Break message to the SL state machines (see 6.15) and shall terminate this state machine.

If the SMP request frame is received with a CRC error, then this state shall discard the frame, send a Request Break message to the SL state machines (see 6.15) and shall terminate this state machine.

Otherwise, this state shall send a Frame Received (SMP Successful) confirmation to the port layer.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

6.19.5.4.2.2 Transition SMP_TP1:Receive_Frame to SMP_TP2:Transmit_Frame

This transition shall occur after sending a Frame Received (SMP Successful) confirmation to the port layer.

6.19.5.4.3 SMP_TP2:Transmit_Frame state

If this state receives an SMP Transmit Break request, then this state shall send a Request Break message to the SL state machines and terminate this state machine.

If this state receives a Tx Frame request, then this state shall send a Transmit Frame message to the SMP transmitter with an argument containing the frame contents, then wait for a Frame Transmitted message. After receiving a Frame Transmitted message, this state shall send a Frame Transmitted confirmation to the port layer, send a Request Close message to the SL state machines (see 6.15) and terminate this state machine.

After sending Transmit Frame message to the SMP transmitter, this state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

7 Port layer

7.1 Port layer overview

The port layer (PL) state machines interface with one or more SAS link layer state machines and one or more SSP, SMP, and STP transport layer state machines to establish port connections and disconnections. The port layer state machines also interpret or pass transmit data, receive data, commands, and confirmations between the link and transport layers.

7.2 PL (port layer) state machines

7.2.1 PL state machines overview

The PL (port layer) consists of state machines that run in parallel and perform the following functions:

- a) receive requests from the SSP, SMP, and STP transport layer state machines for connection management (e.g., requests to open or close connections) and frame transmission;
- b) send requests to the SAS link layer state machines for connection management and frame transmission;
- c) receive confirmation from the SAS link layer state machines; and
- d) send confirmations to the SSP, SMP, and STP transport layer state machines.

The port layer state machines are as follows:

- a) PL_OC (port layer overall control) state machines (see 7.2.2); and
- b) PL_PM (port layer phy manager) state machines (see 7.2.3).

There is one PL_OC state machine per port (see 4.1.4). There is one PL_PM state machine for each phy contained in the port. Phys are assigned to ports by the management application layer. More than one port in a SAS device may have the same SAS address if the ports are in different SAS domains (see 4.2.9).

Figure 134 shows examples of the port layer state machines and their interaction with the transport and link layers.

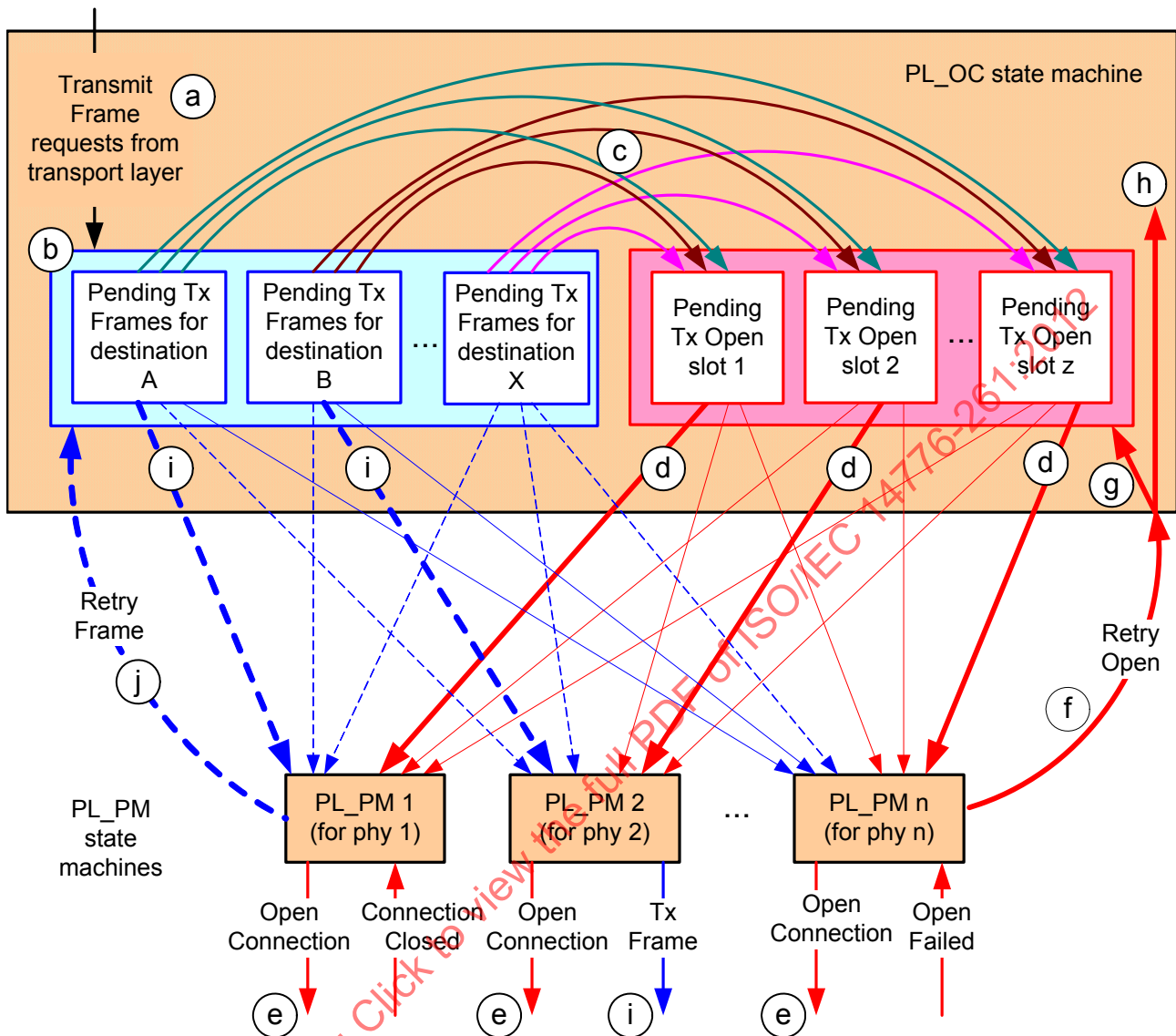


Figure 134 — Port layer examples

The following is a description of the example processes in figure 134. These example processes do not describe all of the possible condition or actions.

- a) Transmit Frame requests are received by the PL_OC state machine;
- b) the PL_OC state machine converts Transmit Frame requests into pending Tx Frame messages associated with the destination SAS address;
- c) the PL_OC state machine generates a pending Tx Open message for a pending Tx Frame message when there is a pending Tx Open slot available (i.e., the number of pending Tx Open messages is less than or equal to the number of destination SAS addresses);
- d) the PL_OC state machine sends a pending Tx Open message as a Tx Open message to a PL_PM state machine when a PL_PM machine is available; a slot is then available for a new pending Tx Open message;
- e) when a PL_PM state machine receives a Tx Open message, the PL_PM state machine attempts to establish a connection with the destination SAS address through the link layer;
- f) if a PL_PM state machine is unable to establish a connection with the destination SAS address, then the PL_PM state machine sends a Retry Open message to the PL_OC state machine;
- g) if there is a pending Tx Open slot available, then the PL_OC state machine converts a Retry Open message to a pending Tx Open message with the pathway blocked count and arbitration wait time

context from the Retry Open message applied to the pending Tx Open message, and may start the Reject To Open Limit timer;

- h) if the PL_OC state machine does not convert a Retry Open to a pending Tx Open frame, then the PL_OC discards the Retry Open message. The PL_OC state machine may create a new Tx Open message for the same pending Tx Frame at a later time or send the appropriate Transmission Status confirmation to the transport layer. If the PL_OC state machine discards a Retry Open message, then the pathway blocked count and arbitration wait time context from the Retry Open message are also discarded;
- i) after the Reject To Open Limit timer, if any, has expired and after a PL_PM state machine establishes a connection with a destination SAS address, the PL_OC state machine sends pending Tx Frame messages for the destination to the PL_PM state machine as Tx Frame messages;
- j) if a PL_PM state machine is unable to send a Tx Frame message to the link layer as a Tx Frame request (e.g., due to a credit timeout), then the PL_PM state machine sends a Retry Frame message to the PL_OC state machine, and the PL_OC state machine converts the Retry Frame message into a pending Tx Frame message; and
- k) if the PL_PM state machine is able to send a Tx Frame message as a Tx Frame request to the link layer, then the PL_PM state machine sends a Transmission Status confirmation to the transport layer.

The Transmission Status confirmation from either the PL_OC state machine or a PL_PM state machine shall include the following as arguments:

- a) initiator port transfer tag;
- b) Destination SAS Address; and
- c) Source SAS Address.

7.2.2 PL_OC (port layer overall control) state machine

7.2.2.1 PL_OC state machine overview

The PL_OC state machine:

- a) receives requests from the SSP, SMP, and STP transport layers;
- b) sends messages to the PL_PM state machine;
- c) receives messages from the PL_PM state machine;
- d) selects frames to transmit;
- e) selects phys on which to transmit frames;
- f) receives confirmations from the link layer;
- g) sends confirmations to the transport layer;
- h) has Arbitration Wait Time timers;
- i) has I_T Nexus Loss timers; and
- j) may have Reject To Open Limit timers.

This state machine consists of the following states:

- a) PL_OC1:Idle (see 7.2.2.2) (initial state); and
- b) PL_OC2:Overall_Control (see 7.2.2.3).

This state machine shall start in the PL_OC1:Idle state after power on.

This state machine shall maintain:

- a) a pool of pending Tx Frame messages for each destination SAS address; and
- b) a pool of pending Tx Open message slots. There shall only be at most a single pending Tx Open message slot for each destination SAS address. There may be fewer total pending Tx Open message slots than the total number of destination SAS addresses.

This state machine shall maintain the timers listed in table 113.

Table 113 — PL_OC state machine timers

Timer	Maximum number of timers	Initial value
I_T Nexus Loss timer	One per destination SAS address	Depending on the protocol used by the port: a) for SSP target ports, the value in the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page (see 9.2.7.4); b) for SSP initiator ports, the value in the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page for the SSP target port with that destination SAS address (see 9.2.7.4); c) for STP target ports, the value in the STP SMP I_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function (see 9.4.3.18); d) for STP initiator ports, the value in the STP SMP I_T NEXUS LOSS TIME field in the SMP REPORT GENERAL function (see 9.4.3.4) for the STP target port with that destination SAS address; or e) for SMP initiator ports, the value in the STP SMP I_T NEXUS LOSS TIME field in the SMP REPORT GENERAL function (see 9.4.3.4).
Arbitration Wait Time timer	One per pending Tx Open message	0000h, a vendor specific value less than 8000h (see 6.13.3), or the value received with a Retry Open message.
Reject To Open Limit timer	One per Retry Open message	Depending on the protocol used by the port: a) for SSP target ports, the value in the REJECT TO OPEN LIMIT field in the Protocol-Specific Port mode page (see 9.2.7.4); b) for SSP initiator ports, a vendor specific value; c) for STP target ports, the value in the STP REJECT TO OPEN LIMIT field in the SMP CONFIGURE GENERAL function (see 9.4.3.18); or d) for STP initiator ports, a vendor specific value.

Figure 135 shows the PL_OC state machine.

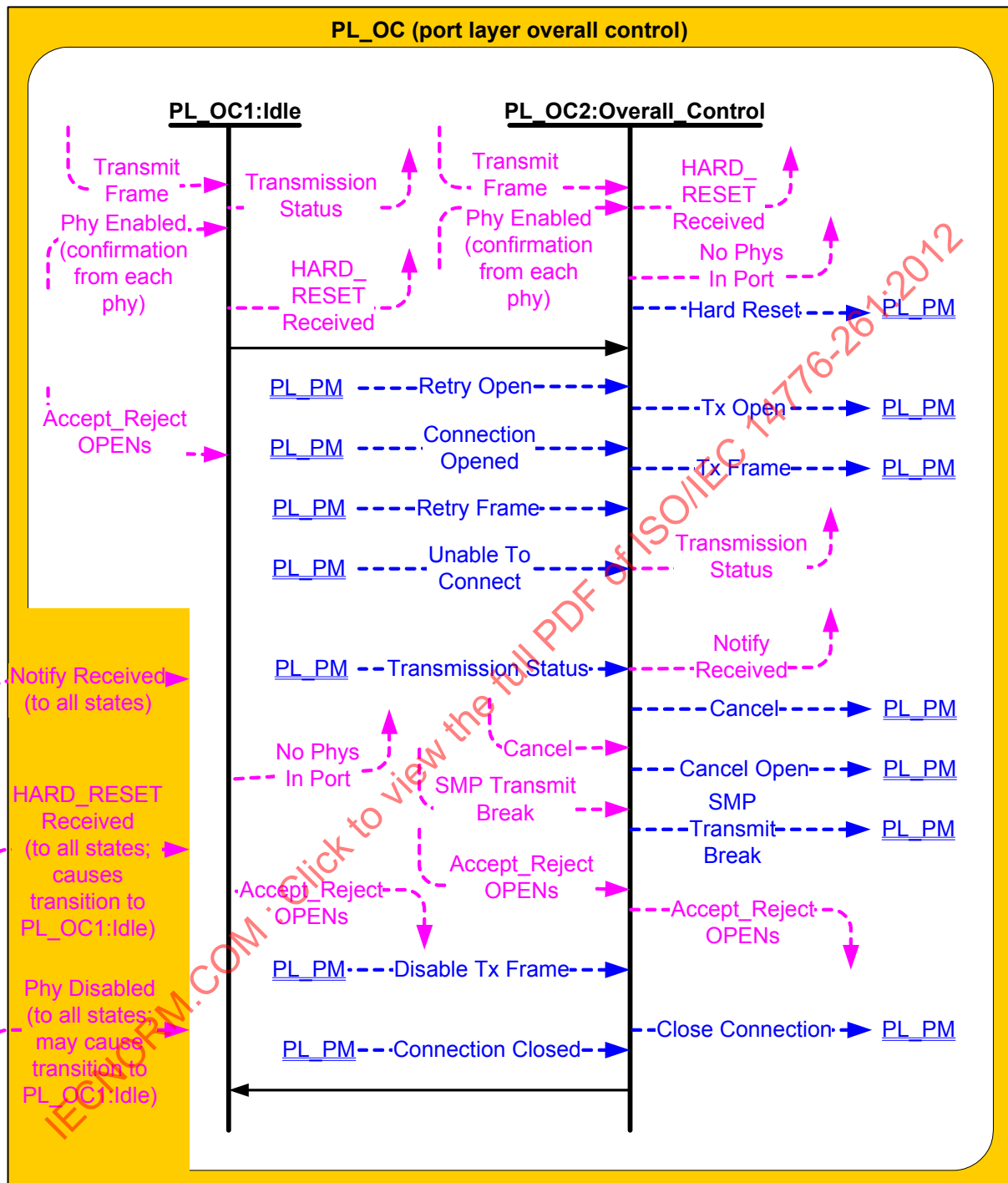


Figure 135 — PL_OC (port layer overall control) state machine

7.2.2.2 PL_OC1:Idle state

7.2.2.2.1 PL_OC1:Idle state description

This state is the initial state of the PL_OC state machine.

If this state receives a HARD_RESET Received confirmation, then this state shall send a HARD_RESET Received confirmation to the transport layer.

If this state receives a Notify Received (Power Loss Expected) confirmation, then this state shall send a Notify Received (Power Loss Expected) confirmation to the transport layer.

If this state receives an Accept_Reject OPENs request, then this state shall send an Accept_Reject OPENs request with the same arguments to all link layers in the port.

If this state receives a Transmit Frame request, then this state shall send a No Phys In Port confirmation to the transport layer.

If an I_T Nexus Loss timer expires for a destination SAS address, then this state shall perform the following:

- a) delete the I_T Nexus Loss timer for the SAS address;
- b) send a Transmission Status (I_T Nexus Loss) confirmation for each pending Tx Frame message for the SAS address; and
- c) discard each pending Tx Frame message for the SAS address and any corresponding pending Tx Open messages.

If the port is an STP target port or an STP initiator port, then the port shall handle all pending commands as described in 4.5.

7.2.2.2.2 Transition PL_OC1:Idle to PL_OC2:Overall_Control

This transition shall occur after a Phy Enabled confirmation is received for at least one phy assigned to the port.

7.2.2.3 PL_OC2:Overall_Control state

7.2.2.3.1 PL_OC2:Overall_Control state overview

This state may receive Transmit Frame requests from the transport layers (i.e., SSP and SMP) and Retry frame messages from PL_PM state machines. This state shall create a pending Tx Frame message for each received Transmit Frame request and Retry Frame message. There may be more than one pending Tx Frame message at a time for each SSP transport layer. There shall be only one pending Tx Frame message at a time for each SMP transport layer.

This state selects PL_PM state machines through which connections are established. This state shall only attempt to establish connections through PL_PM state machines whose phys are enabled. In a vendor specific manner, this state selects PL_PM state machines on which connections are established to transmit frames. This state shall receive a response to a message from a PL_PM state machine before sending another message to that PL_PM state machine.

This state also:

- a) receives connection management requests from the transport layers;
- b) sends connection management messages to PL_PM state machines;
- c) receives connection management messages from PL_PM state machines; and
- d) sends connection management confirmations to the transport layers.

After receiving a Transmit Frame request for a destination SAS address for which there is no connection established and for which no I_T Nexus Loss timer has been created, this state shall create an I_T Nexus Loss timer for that SAS address if:

- a) the protocol is SSP, the port is an SSP target port, the Protocol-Specific Port mode page is implemented, and the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page (see 9.2.7.4) is not set to 0000h;
- b) the protocol is STP, the port is an STP target port, and the STP SMP I_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function is not set to 0000h; or
- c) the protocol is SMP, the port is an SMP initiator port, and the STP SMP I_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function is not set to 0000h.

This state may create an I_T Nexus Loss timer for that SAS address if:

- a) the protocol is SSP and the port is an SSP initiator port; or;
- b) the protocol is STP and the port is an STP initiator port.

When this state creates an I_T Nexus Loss timer it shall:

- 1) initialize the I_T Nexus Loss timer as specified in table 113 (see 7.2.2.1); and
- 2) not start the I_T Nexus Loss timer.

If this state machine is in an SSP initiator port, then this state may create an I_T Nexus Loss timer for the SAS address. If a state machine in an SSP initiator port creates an I_T Nexus Loss timer, then the state machine should use the value in the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page for the SSP target port (see 9.2.7.4) as the initial value for its I_T Nexus Loss timer.

If there are no pending Tx Frame messages for a destination SAS address and an I_T Nexus Loss timer has been created for that destination SAS address, then this state shall delete the I_T Nexus Loss timer for that destination SAS address.

If this state receives a HARD_RESET Received confirmation, then this state shall:

- a) discard all pending Tx Frame messages;
- b) discard all pending Tx Open messages;
- c) delete all timers (e.g., I_T Nexus Loss timers, Arbitration Wait Time timers, and Reject To Open Limit timers);
- d) send a Hard Reset message to each PL_PM state machine; and
- e) send a HARD_RESET Received confirmation to the transport layer.

If this state receives a Notify Received (Power Loss Expected) confirmation, then this state shall:

- a) discard all pending Tx Frame messages;
- b) discard all pending Tx Open messages;
- c) delete all timers (e.g., I_T Nexus Loss timers, Arbitration Wait Time timers, and Reject To Open Limit timers);
- d) send a Close Connection message to each of the PL_PM state machines;
- e) send a Cancel Open message to each of the PL_PM state machines; and
- f) send a Notify Received (Power Loss Expected) confirmation to the transport layer.

If this state receives a Phy Disabled confirmation from all the link layers in the port, then this state shall:

- a) discard all pending Tx Frame messages;
- b) discard all pending Tx Open messages;
- c) delete all timers (e.g., I_T Nexus Loss timers, Arbitration Wait Time timers, and Reject To Open Limit timers); and
- d) send a No Phys In Port confirmation to the transport layer.

7.2.2.3.2 PL_OC2:Overall_Control state establishing connections

This state receives Phy Enabled confirmations indicating when a phy is available.

This state receives Retry Open messages from a PL_PM state machine.

This state creates pending Tx Open messages based on pending Tx Frame messages and Retry Open messages. Pending Tx Open messages are sent to a PL_PM state machine as Tx Open messages. This state shall discard a pending Tx Open message if there are no pending Tx Frame messages for that destination (e.g., after accepting an incoming connection in which the other phy provides credit).

See 6.13.2.1 for additional requirements and recommendations on how this state decides to create pending Tx Open messages.

If this state receives a Retry Open (Retry) message or a Retry Open (Low Phy Power Condition) message, then this state shall process the Retry Open message.

If this state receives a Retry Open (No Destination) message or a Retry Open (Open Timeout Occurred) message and an I_T Nexus Loss timer has not been created for the destination SAS address (e.g., an SSP target port does not support the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page, or the field is set to 0000h), then this state shall process the Retry Open message as either a Retry Open message or an Unable To Connect message. This selection is vendor specific.

If this state receives a Retry Open (Pathway Blocked) message or Retry Open (Break Received) message, and an I_T Nexus Loss timer has not been created for the destination SAS address, then this state shall process the Retry Open message.

If this state receives a Retry Open (No Destination) message, a Retry Open (Open Timeout Occurred) message, Retry Open (Break Received) message, or a Retry Open (Pathway Blocked) message, and an I_T Nexus Loss timer has been created for the destination SAS address with an initial value of FFFFh, then this state shall process the Retry Open message (i.e., the Retry Open message is never processed as an Unable to Connect message).

If:

- a) this state receives a Retry Open (No Destination) message, Retry Open (Break Received) message, or a Retry Open (Open Timeout Occurred) message;
- b) an I_T Nexus Loss timer has been created for the destination SAS address; and
- c) there is no connection established with the destination SAS address,

then this state shall check the I_T Nexus Loss timer, and:

- a) if the I_T Nexus Loss timer is not running, the I_T nexus loss time is not set to FFFFh, and the CONFIGURING bit is set to zero in the REPORT GENERAL response (see 9.4.3.4) for each expander device between this port and the destination port that is two or more levels away from this port, then this state shall start the timer;

NOTE 64 The port layer may require assistance from the management application layer to determine the values of the CONFIGURING bits; this interaction is not specified by this standard. Consequently, the I_T Nexus Loss timer may not start immediately after this state receives a Retry Open (No Destination) message or a Retry Open (Open Timeout Occurred) message.

- b) if the I_T Nexus Loss timer is running, then this state shall not stop the timer; and
- c) if the I_T Nexus Loss timer has expired, then this state shall process the Retry Open message as if it were an Unable To Connect message (see 7.2.2.3.3.3).

If this state:

- a) receives a Retry Open (Pathway Blocked) message, Retry Open (Low Phy Power Condition) message, or Retry Open (Break Received) message;
- b) an I_T Nexus Loss timer has been created for the destination SAS address; and
- c) there is no connection established with the destination SAS address,

then this state shall check the I_T Nexus Loss timer, and:

- a) if the I_T Nexus Loss timer is running, then this state shall not stop the timer; and
- b) if the I_T Nexus Loss timer has expired, then this state shall process the Retry Open message as if it were an Unable To Connect message (see 7.2.2.3.3.3).

If this state receives a Retry Open (Retry) message and an I_T Nexus Loss timer is running for the destination SAS address, then this state shall:

- a) stop the I_T Nexus Loss timer (if the timer has been running); and
- b) initialize the I_T Nexus Loss timer.

This state shall create a pending Tx Open message if:

- a) this state has a pending Tx Frame message or has received a Retry Open message;
- b) there is a pending Tx Open message slot available for the destination SAS address;
- c) there is no pending Tx Open message for the destination SAS address; and
- d) there is no connection established with the destination SAS address.

This state may create a pending Tx Open message if:

- a) this state has a pending Tx Frame message, or this state has received a Retry Open message and has not processed the message by sending a confirmation; and
- b) there is a pending Tx Open message slot available for the destination SAS address.

If this state receives a Retry Open message and there are pending Tx Frame messages for which pending Tx Open messages have not been created, then this state should create a pending Tx Open message from the Retry Open message.

If this state does not create a pending Tx Open message from a Retry Open message (e.g., there is not an available pending Tx Open message slot for the destination SAS address), then this state shall discard the Retry Open message. This state may create a new pending Tx Open message at a later time for the pending Tx Frame message that resulted in the Retry Open message.

If this state receives a Retry Open (Opened By Destination) message, and the initiator port bit and protocol arguments match those in the Tx Open messages that resulted in the Retry Open message, then this state may discard the Retry Open message and use the established connection to send pending Tx Frame messages as Tx Frame messages to the destination SAS address. If this state receives a Retry Open (Opened By Destination) message and state has a pending Tx Open slot available, then this state may create a pending Tx Open message from the Retry Open message.

NOTE 65 If a connection is established by another port as indicated by a Retry Open (Opened By Destination) message, credit may not be granted for frame transmission. In this case this state may create a pending Tx Open message from a Retry Open message in order to establish a connection where credit is granted.

This state shall send a pending Tx Open message as a Tx Open message to a PL_PM state machine that has an enabled phy and does not have a connection established. If there is more than one pending Tx Open message, then this state should send a Tx Open message for the pending Tx Open message that has been pending for the longest time first.

If low phy power conditions (see 4.10.1) are enabled, then this state should use the following precedence in selecting the PL_PM state machine to send a pending Tx Open message:

- 1) PL_PM state machine with its phy in the active phy power condition (see 4.10.2);
- 2) PL_PM state machine with its phy in the partial phy power condition (see 4.10.1.3); and
- 3) PL_PM state machine with its phy in the slumber phy power condition (see 4.10.1.4).

If this state creates a pending Tx Open message from one of the following messages:

- a) Retry Open (Opened By Destination);
- b) Retry Open (Opened By Other);
- c) Retry Open (Collided);
- d) Retry Open (Pathway Blocked);
- e) Retry Open (Low Phy Power Condition); or
- f) Retry Open (Retry) if the CONTINUE AWT bit is set to one in the Protocol-specific Port mode page (see 9.2.7.4),

then this state shall:

- 1) create an Arbitration Wait Time timer for the pending Tx Open message;
- 2) set the Arbitration Wait Time timer for the pending Tx Open message to the arbitration wait time argument from the Retry Open message; and
- 3) start the Arbitration Wait Time timer for the pending Tx Open message.

When a pending Tx Open message is sent to a PL_PM state machine as a Tx Open message, the Tx Open message shall contain the following arguments to be used in an OPEN address frame:

- a) Initiator Port Bit from the Transmit Frame request;
- b) Protocol from the Transmit Frame request;
- c) Connection Rate from the Transmit Frame request;
- d) Initiator Connection Tag from the Transmit Frame request;
- e) Destination SAS Address from the Transmit Frame request;
- f) Source SAS Address from the Transmit Frame request;
- g) Pathway Blocked Count; and
- h) Arbitration Wait Time.

If this state creates a pending Tx Open message from one of the following:

- a) a Transmit Frame request;
- b) a Retry Open (No Destination) message;
- c) a Retry Open (Open Timeout Occurred) message;
- d) a Retry Open (Break Received) message; or
- e) a Retry Open (Retry) message if the CONTINUE AWT bit is set to zero in the Protocol-specific Port mode page (see 9.2.7.4),

then this state shall:

- a) set the pathway blocked count argument in the Tx Open message to zero; and
- b) set the arbitration wait time argument in the Tx Open message to zero or a vendor specific value less than 8000h (see 6.13.3).

If a pending Tx Open message was created as the result of this state receiving a Retry Open (Retry) message and:

- a) the protocol for the connection is SSP, the Protocol-Specific Port mode page is implemented, and the REJECT TO OPEN LIMIT field in the Protocol-Specific Port mode page (see 9.2.7.4) is not set to zero; or
- b) the protocol for the connection is STP and the STP REJECT TO OPEN LIMIT field is not set to zero in the SMP REPORT GENERAL response (see 9.4.3.4),

then this state shall:

- 1) create a Reject To Open Limit timer associated with the pending Tx Open message that received the Retry Open (Retry) message;
- 2) initialize the Reject To Open Limit timer as specified in table 113 (see 7.2.2.1);
- 3) start the Reject To Open Limit timer; and
- 4) wait at least until the Reject To Open Limit timer expires before sending a Tx Open message.

If a pending Tx Open message was created as the result of this state receiving a Retry Open (Pathway Blocked) message, then:

- a) if the Retry Open message pathway blocked count argument is FFh, then this state shall set the Tx Open pathway blocked count argument to FFh; or
- b) if the Retry Open pathway blocked count argument is less than FFh, then this state shall set the Tx Open pathway blocked count argument to the Retry Open pathway blocked count argument plus 01h.

If a pending Tx Open message was created as the result of this state receiving one of the following:

- a) a Retry Open (Opened By Destination) message;
- b) a Retry Open (Opened By Other) message;
- c) a Retry Open (Collided) message;
- d) a Retry Open (Pathway Blocked) message;
- e) a Retry Open (Low Phy Power Condition) message; or
- f) a Retry Open (Retry) message if the CONTINUE AWT bit is set to one in the Protocol-specific Port mode page (see 9.2.7.4),

then this state shall set the arbitration wait time argument in the Tx Open message to be the value from the Arbitration Wait Time timer created as a result of the Retry Open message.

After this state sends a Tx Open message, this state shall discard the pending Tx Open message from which the Tx Open message was created. After this state discards a pending Tx Open message, this state may create a new pending Tx Open message.

If this state receives a Connection Opened message and the initiator port bit and protocol arguments match those in a pending Tx Open message, then any Reject To Open Limit timer associated with that pending Tx Open message shall be discarded.

If this state receives a Connection Opened message and the initiator port bit and protocol arguments match those in any pending Tx Frame messages, then this state may use the established connection to send pending Tx Frame messages as Tx Frame messages to the destination SAS address.

7.2.2.3.3 PL_OC2:Overall_Control state connection established

7.2.2.3.3.1 General

If this state receives a Connection Opened message or a Retry Open (Opened By Destination) message for a SAS address, and an I_T Nexus Loss timer has been created for the SAS address, then this state shall:

- if the I_T Nexus Loss timer for the SAS address has been running, then stop the timer; and
- initialize the I_T Nexus Loss timer.

7.2.2.3.3.2 PL_OC2:Overall_Control state unable to establish a connection - Unable To Connect message

If this state receives an Unable To Connect message shown in table 114, then this state shall establish an I_T nexus loss event (see 7.2.2.3.3.5).

Table 114 — Confirmations from Unable To Connect messages

Message received	Confirmation to be sent to transport layer
Unable to Connect (Bad Destination)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Connection Rate Not Supported)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Protocol Not Supported)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Reserved Abandon 1)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Reserved Abandon 2)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Reserved Abandon 3)	Transmission Status (I_T Nexus Loss)
Unable To Connect (STP Resources Busy)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Wrong Destination)	Transmission Status (I_T Nexus Loss)
Unable To Connect (Zone Violation)	Transmission Status (I_T Nexus Loss)

7.2.2.3.3.3 PL_OC2:Overall_Control state unable to establish a connection - Unable To Connect message - Retry Open message processed as an Unable To Connect message

If this state receives a Retry Open (No Destination) message, a Retry Open (Open Timeout Occurred) message, Retry Open (Low Phy Power Condition), or a Retry Open (Pathway Blocked) message and processes it as an Unable To Connect message, then this state shall establish an I_T nexus loss event (see 7.2.2.3.3.5).

7.2.2.3.3.4 PL_OC2:Overall_Control state unable to establish a connection - I_T Nexus Loss timer expires

If this state receives a Retry Open (No Destination) message, a Retry Open (Open Timeout Occurred) message, Retry Open (Break Received), Retry Open (Low Phy Power Condition), or a Retry Open (Pathway Blocked) message and the I_T Nexus Loss timer for the SAS address has expired, then this state shall establish an I_T nexus loss event (see 7.2.2.3.3.5).

7.2.2.3.3.5 PL_OC2:Overall_Control state - I_T nexus loss

If an I_T nexus loss event occurs, then this state shall perform the following:

- delete the I_T Nexus Loss timer for the SAS address;
- discard the Retry Open message;

- c) send a Transmission Status (I_T Nexus Loss) confirmation for the pending Tx Frame message from which the Retry Open message resulted;
- d) discard the pending Tx Frame message from which the Retry Open message resulted;
- e) if this state has any pending Tx Frame messages with the same destination SAS address and protocol as the Retry Open message, and this state has not sent a Tx Open message to a PL_PM state machine for the messages, then this state shall send a Transmission Status (I_T Nexus Loss) confirmation for each pending Tx Frame message and discard the pending Tx Frame messages and any corresponding pending Tx Open messages; and
- f) if this state has any pending Tx Frame messages with the same destination SAS address and protocol as the Retry Open message, and this state has sent a Tx Open message to a PL_PM state machine for a message, then this state shall send a Cancel Open message to each PL_PM state machine to which it has sent a Tx Open message. After receiving an Unable To Connect (Arb Stopped) message from a PL_PM state machine in response to the Cancel Open message, this state shall send a Transmission Status (I_T Nexus Loss) confirmation for each pending Tx Frame message and discard the pending Tx Frame messages and any corresponding pending Tx Open messages.

7.2.2.3.4 PL_OC2:Overall_Control state connection management

If this state receives an Accept_Reject OPENs request, then this state shall send an Accept_Reject OPENs request with the same arguments to all phys in the port.

If this state receives an SMP Transmit Break request, then this state shall send an SMP Transmit Break message to the PL_PM state machine associated with the corresponding SMP transport state machine. If there is no PL_PM state machine associated with the request, then the PL_OC state machine shall ignore the request.

If this state receives one of the following:

- a) a Connection Closed (Close Timeout) message;
- b) a Connection Closed (Break Requested) message; or
- c) a Connection Closed (Break Received) message,

then this state shall not send a Tx Open or Tx Frame message to the PL_PM state machine that sent the message until this state receives a Connection Closed (Transition to Idle) message from that PL_PM state machine.

If this state receives a Connection Closed (Normal) message or a Connection Closed (Transition to Idle) message indicating that a connection with a destination SAS address is no longer open and this state has pending Tx Open messages, then this state may send a Tx Open message to the PL_PM state machine that sent the Connection Closed message.

If this port is a wide SSP port, then this state shall not reject an incoming connection request on one phy because it has an outgoing connection request on another phy.

If this port is an SSP port, there are no pending Tx Frame messages for a destination SAS address with which a PL_PM state machine has established a connection, and the connection was established by a message from this state, then this state should send a Close Connection message to the PL_PM state machine.

If this port is an SSP port, has no pending Tx Frame messages for a destination SAS address with which a PL_PM state machine has established a connection, and the connection was established by the destination, then this state may wait a vendor specific time and then shall send a Close Connection message to the PL_PM state machine.

If this state has received a Disable Tx Frame message from a PL_PM state machine, then this state should send a Close Connection message to the PL_PM state machine.

NOTE 66 The PL_PM state machine sends a Close Connection request to the link layer upon receipt of a Close Connection message or on expiration of the Bus Inactivity Time Limit timer (see 7.2.3.4.1).

7.2.2.3.5 PL_OC2:Overall_Control state frame transmission

In order to prevent livelocks, if this port is a wide SSP port, has multiple connections established, and has a pending Tx Frame message, then this state shall send at least one Tx Frame message to a PL_PM state machine before sending a Close Connection message to the PL_PM state machine.

After this state receives a Connection Opened message from a PL_PM state machine, this state selects pending Tx Frame messages for the destination SAS address with the same initiator port bit and protocol arguments, and, as an option, the same connection rate argument, and sends the messages to the PL_PM state machine as Tx Frame messages.

This state may send a Tx Frame message to any PL_PM state machine that has established a connection with the destination SAS address when the initiator port bit and protocol arguments match those in the Tx Frame message.

After this state sends a Tx Frame message to a PL_PM state machine, this state shall not send another Tx Frame message to that PL_PM state machine until it receives a Transmission Status (Frame Transmitted) message.

This state shall not send a Tx Frame message containing a Request Fence argument or Response Fence argument to any PL_PM state machine until this state has received one of the following messages for each Tx Frame message with the same nexus as specified by that Request Fence argument or Response Fence argument:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

After this state sends a Tx Frame message containing a Request Fence argument or Response Fence argument, this state shall not send another Tx Frame message with the same nexus as specified by that Request Fence argument or Response Fence argument until this state has received one of the following messages:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

Once this state has sent a Tx Frame message containing a Non-Interlocked argument to a PL_PM state machine, this state shall not send a Tx Frame message containing a Non-Interlocked argument with the same I_T_L_Q nexus to another PL_PM state machine until this state has received one of the following messages for each Tx Frame message containing a Non-Interlocked argument for the same I_T_L_Q nexus:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

For a bidirectional command, frames with the Non-Interlocked argument for an I_T_L_Q nexus may be transmitted on one phy at the same time as frames with the Non-Interlocked argument for the same I_T_L_Q nexus are received on the same phy or on a different phy.

If this port is an SMP initiator port, then this state shall send the Tx Frame message containing the SMP REQUEST frame to the PL_PM state machine on which the connection was established for the Tx Open message. If this port is an SMP target port, then this state shall send the Tx Frame message containing the SMP RESPONSE frame to the PL_PM state machine on which the connection was established for the Tx Open message. See 6.19 for additional information about SMP connections.

Characteristics of STP connections are defined by SATA (also see 6.18).

The following arguments shall be included with the Tx Frame message:

- a) the frame to be transmitted; and
- b) Balance Required or Balance Not Required.

A Balance Not Required argument shall only be included if:

- a) the request was a Transmit Frame (Non-Interlocked) request (i.e., the request included a DATA frame); and
- b) the last Tx Frame message sent to this PL_PM state machine while this connection has been established was for a DATA frame having the same logical unit number and initiator port transfer tag as the DATA frame in this Tx Frame message.

If a Balance Not Required argument is not included in the Tx Frame message, then a Balance Required argument shall be included.

If this state receives a Disable Tx Frames message from a PL_PM state machine, then this state should send no more Tx Frame messages to that state machine until a new connection is established.

7.2.2.3.6 PL_OC2:Overall_Control state frame transmission cancellations

Cancel requests cause this state to cancel previous Transmit Frame requests. A Cancel request includes the following arguments:

- a) Destination SAS Address; and
- b) Initiator Port Transfer Tag.

If this state receives a Cancel request and has not already sent a Tx Frame message for the Transmit Frame request to a PL_PM state machine for the Transmit Frame request specified by the Cancel request, then this state shall:

- a) discard all Transmit Frame requests for the specified destination SAS address and initiator port transfer tag; and
- b) send a Transmission Status (Cancel Acknowledge) confirmation to the transport layer.

If this state receives a Cancel request and has already sent a Tx Frame message to a PL_PM state machine for the Transmit Frame request specified by the Cancel request, then this state shall send a Cancel message to the PL_PM state machine to which the Tx Frame message was sent. The Cancel message shall include the initiator port transfer tag.

7.2.2.3.7 Transition PL_OC2:Overall_Control to PL_OC1:Idle

This transition shall occur after:

- a) sending a HARD_RESET Received confirmation to the transport layer;
- b) a Phy Disabled confirmation is received from all of the link layers in the port; or
- c) sending a Notify Received (Power Loss Expected) confirmation to the transport layer.

7.2.3 PL_PM (port layer phy manager) state machine

7.2.3.1 PL_PM state machine overview

A PL_PM state machine:

- a) receives messages from the PL_OC state machine;
- b) sends requests to the link layer;
- c) receives confirmations from the link layer;
- d) sends confirmations to the transport layer;
- e) sends messages to PL_OC state machine;
- f) has an Arbitration Wait Time timer;
- g) may have a Bus Inactivity Time Limit timer; and
- h) may have Maximum Connect Time Limit timer.

This state machine consist of the following states:

- a) PL_PM1:Idle (see 7.2.3.2) (initial state);
- b) PL_PM2:Req_Wait (see 7.2.3.3);
- c) PL_PM3:Connected (see 7.2.3.4); and

d) PL_PM4:Wait_For_Close (see 7.2.3.5).

This state machine shall start in the PL_PM1:Idle state after power on.

This state machine shall maintain the timers listed in table 115.

Table 115 — PL_PM state machine timers

Timer	Initial value
Arbitration Wait Time timer	The arbitration wait time argument from a Retry Open message (see 7.2.2.3.1).
Bus Inactivity Time Limit timer	Depending on the protocol used by the port: a) for SSP target ports, the value in the BUS INACTIVITY TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2); b) for STP target ports, the value in the STP BUS INACTIVITY TIME LIMIT field in the SMP REPORT GENERAL response (see 9.4.3.4); c) for SSP initiator ports, the value in the BUS INACTIVITY TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2) of the destination SSP target port; or d) for STP initiator ports, the value in the STP BUS INACTIVITY TIME LIMIT field in the SMP REPORT GENERAL response (see 9.4.3.4) of the destination STP target port.
Maximum Connect Time Limit timer	Depending on the protocol used by the port: a) for an SSP target port, the value in the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2); b) for an STP target port, the value in the STP MAXIMUM CONNECT TIME LIMIT field in the SMP REPORT GENERAL response; c) for an SMP target port, 2 ms; d) for an SSP initiator port, the value in the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2) of the destination SSP target port; or e) for an STP initiator port, the value in the STP MAXIMUM CONNECT TIME LIMIT field in the SMP REPORT GENERAL response of the destination STP target port.

Figure 136 shows part 1 of the PL_PM state machine.

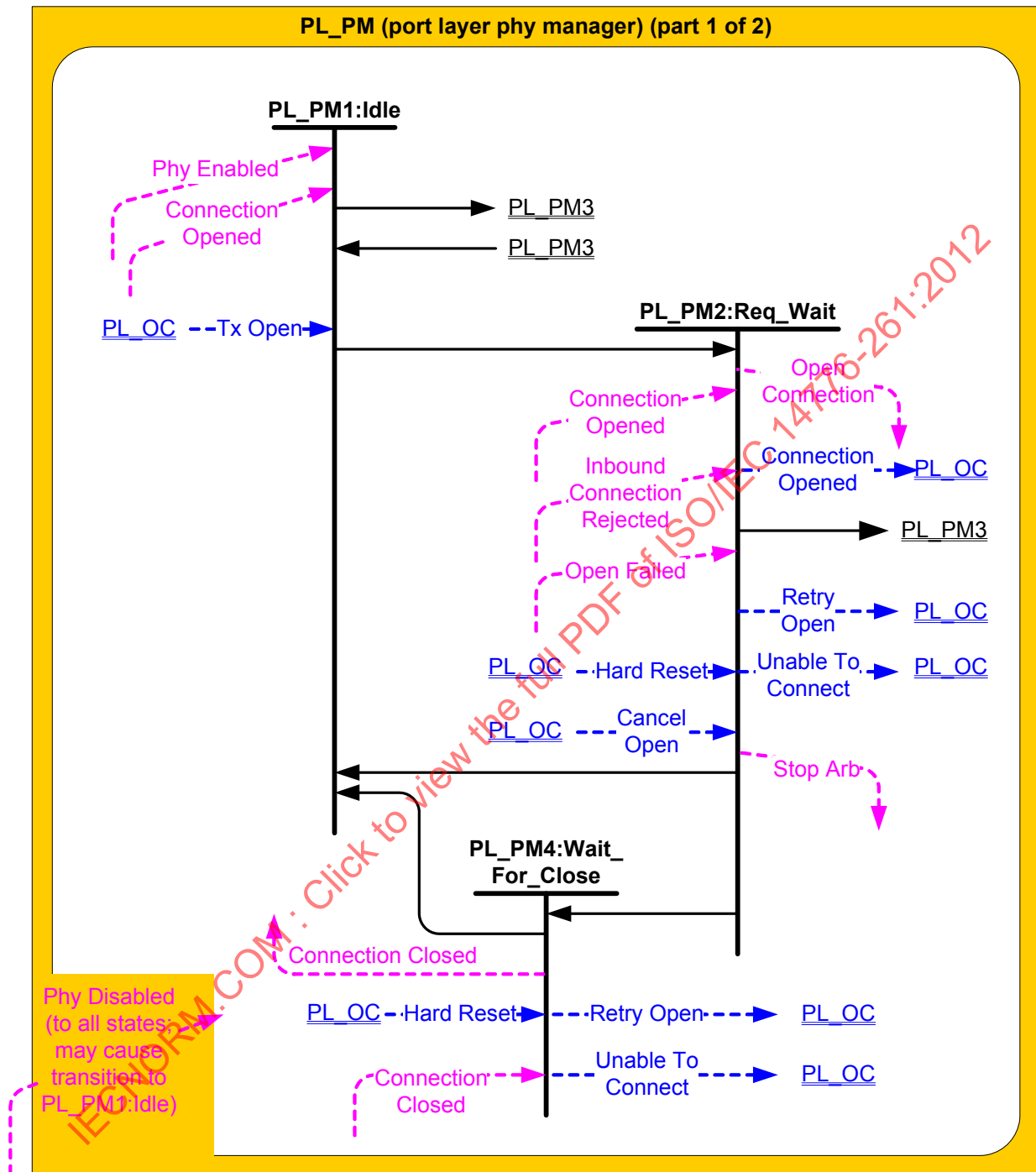


Figure 136 — PL_PM (port layer phy manager) state machine (part 1)

Figure 137 shows part 2 of the PL_PM state machine.

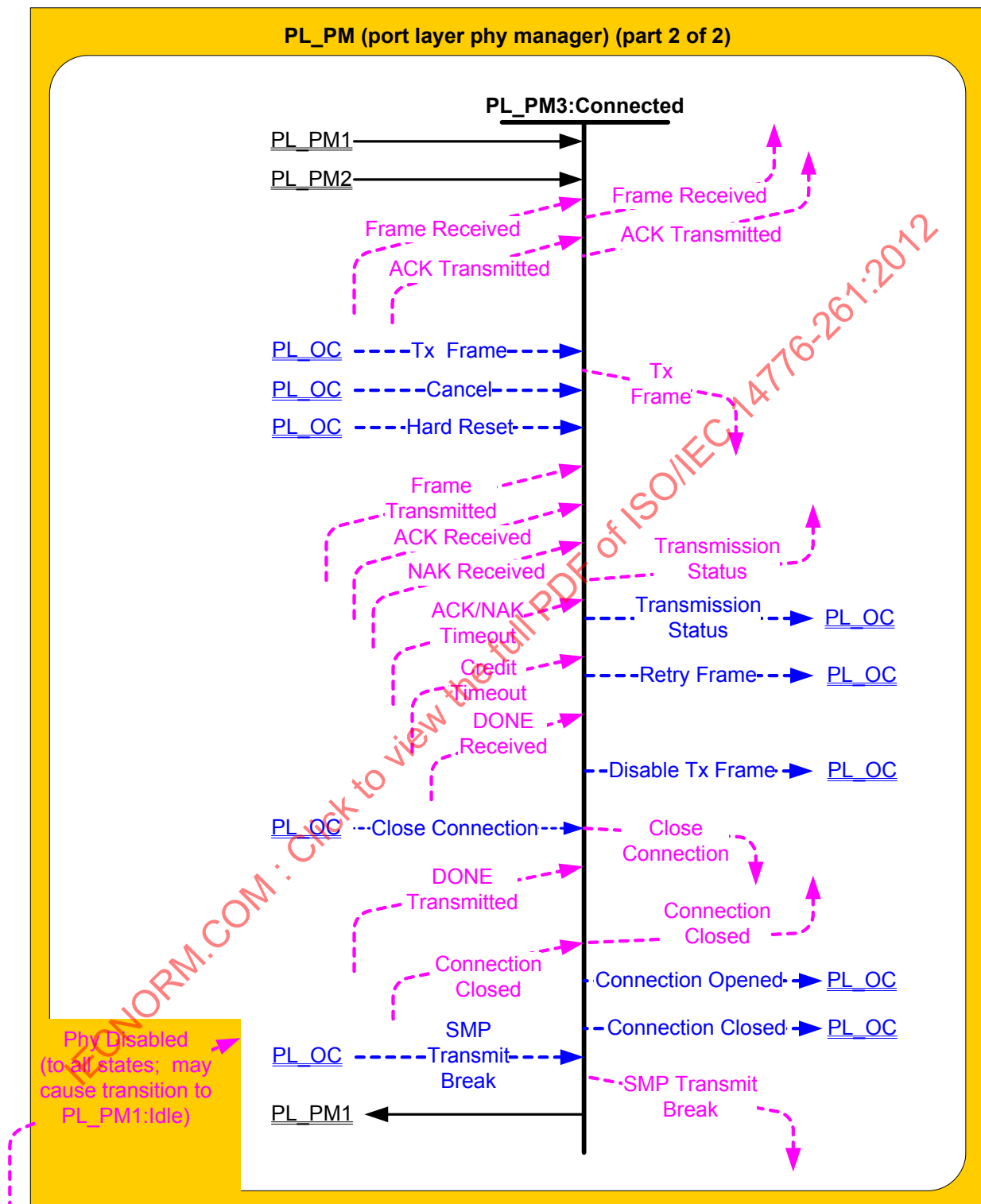


Figure 137 — PL_PM (port layer phy manager) state machine (part 2)

7.2.3.2 PL_PM1:Idle state

7.2.3.2.1 PL_PM1:Idle state description

This is the initial state of the PL_PM state machine.

7.2.3.2.2 Transition PL_PM1:Idle to PL_PM2:Req_Wait

This transition shall occur after:

- a) a Phy Enabled confirmation is received; and
- b) a Tx Open message is received.

7.2.3.2.3 Transition PL_PM1:Idle to PL_PM3:Connected

This transition shall occur after a Connection Opened confirmation is received. The transition shall include the received OPEN address frame.

7.2.3.3 PL_PM2:Req_Wait state

7.2.3.3.1 PL_PM2:Req_Wait state overview

This state sends an Open Connection request to the link layer and waits for a confirmation. This state sends and receives connection management messages to and from the PL_OC state machine.

If this state receives a Hard Reset message, then this state shall terminate all operations.

7.2.3.3.2 PL_PM2:Req_Wait establishing a connection

Upon entry into this state, this state shall:

- 1) create an Arbitration Wait Time timer;
- 2) initialize the Arbitration Wait Time timer to the arbitration wait time argument received with the Tx Open message;
- 3) start the Arbitration Wait Time timer; and
- 4) send an Open Connection request to the link layer.

The Open Connection request shall contain the following arguments from the Tx Open message to be used in an OPEN address frame:

- a) Initiator Port Bit;
- b) Protocol;
- c) Connection Rate;
- d) Initiator Connection Tag;
- e) Destination SAS Address;
- f) Source SAS Address;
- g) Pathway Blocked Count; and
- h) Arbitration Wait Time.

7.2.3.3.3 PL_PM2:Req_Wait connection established

If this state receives a Connection Opened confirmation, then this state shall send a Connection Opened message to the PL_OC state machine.

If this state receives a Connection Opened confirmation and the confirmation was not in response to an Open Connection request from this state (i.e., the connection was established in response to an OPEN address frame from another device), then this state shall discard any Open Connection request and send a Retry Open message to the PL_OC state machine. If the Connection Opened confirmation was from the destination of the Open Connection request, then this state shall send a Retry Open (Opened By Destination) message. If the Connection Opened confirmation was from a destination other than the destination of the Open Connection request, then this state shall send a Retry Open (Opened By Other) message.

A Retry Open (Opened By Destination) or Retry Open (Opened By Other) message shall contain the following arguments:

- a) Initiator Port Bit set to the value received with the Tx Open message;
- b) Protocol set to the value received with the Tx Open message;
- c) Connection Rate set to the value received with the Tx Open message;
- d) Initiator Connection Tag set to the value received with the Tx Open message;
- e) Destination SAS Address set to the value received with the Tx Open message;
- f) Source SAS Address set to the value received with the Tx Open message;
- g) Pathway Blocked Count set to the value received with the Tx Open message; and
- h) Arbitration Wait Time set to the value of the Arbitration Wait Time timer.

7.2.3.3.4 PL_PM2:Req_Wait unable to establish a connection

If this state receives one of the Open Failed confirmations listed in table 116, then this state shall send either a Retry Open message or an Unable To Connect message to the PL_OC state machine.

Table 116 defines the message to be sent for each Open Failed confirmation.

Table 116 — Messages from Open Failed confirmations

Confirmation received	Message to be sent to PL_OC
Open Failed (No Destination)	Retry Open (No Destination)
Open Failed (Pathway Blocked)	Retry Open (Pathway Blocked)
Open Failed (Reserved Continue 0)	Retry Open (Retry)
Open Failed (Reserved Continue 1)	Retry Open (Retry)
Open Failed (Reserved Initialize 0)	Retry Open (No Destination)
Open Failed (Reserved Initialize 1)	Retry Open (No Destination)
Open Failed (Reserved Stop 0)	Retry Open (Pathway Blocked)
Open Failed (Reserved Stop 1)	Retry Open (Pathway Blocked)
Open Failed (Retry)	Retry Open (Retry)
Open Failed (Low Phy Power Condition)	Retry Open (Low Phy Power Condition)
Open Failed (Bad Destination)	Unable To Connect (Bad Destination)
Open Failed (Connection Rate Not Supported)	Unable To Connect (Connection Rate Not Supported)
Open Failed (Protocol Not Supported)	Unable To Connect (Protocol Not Supported)
Open Failed (Reserved Abandon 1)	Unable To Connect (Reserved Abandon 1)
Open Failed (Reserved Abandon 2)	Unable To Connect (Reserved Abandon 2)
Open Failed (Reserved Abandon 3)	Unable To Connect (Reserved Abandon 3)
Open Failed (STP Resources Busy)	Unable To Connect (STP Resources Busy)
Open Failed (Wrong Destination)	Unable To Connect (Wrong Destination)
Open Failed (Zone Violation)	Unable To Connect (Zone Violation)

If this state receives an Inbound Connection Rejected confirmation after sending an Open Connection request, then this state shall discard the Open Connection request and send a Retry Open (Collided) message to the PL_OC state machine.

A Retry Open message shall include the following arguments:

- a) Initiator Port Bit set to the value received with the Tx Open message;
- b) Protocol set to the value received with the Tx Open message;

- c) Connection Rate set to the value received with the Tx Open message;
- d) Initiator Connection Tag set to the value received with the Tx Open message;
- e) Destination SAS Address set to the value received with the Tx Open message;
- f) Source SAS Address set to the value received with the Tx Open message;
- g) Pathway Blocked Count argument set to the value received with the Tx Open message; and
- h) Arbitration Wait Time set to the value of the Arbitration Wait Time timer.

An Unable To Connect message shall include the following arguments:

- a) Initiator Connection Tag set to the value received with the Tx Open message;
- b) Destination SAS Address set to the value received with the Tx Open message; and
- c) Source SAS Address set to the value received with the Tx Open message.

7.2.3.3.5 PL_PM2:Req_Wait connection management

If this state receives a Cancel Open message and a Connection Opened confirmation has not been received, then this state shall send a Stop Arb request to the link layer.

7.2.3.3.6 Transition PL_PM2:Req_Wait to PL_PM1:Idle

This transition shall occur after:

- a) a Retry Open message is sent to the PL_OC state machine;
- b) an Unable To Connect message is sent to the PL_OC state machine;
- c) all operations have been terminated after receiving a Hard Reset message; or
- d) a Phy Disabled confirmation is received.

7.2.3.3.7 Transition PL_PM2:Req_Wait to PL_PM3:Connected

This transition shall occur after a Connection Opened confirmation is received.

7.2.3.3.8 Transition PL_PM2:Req_Wait to PL_PM4:Wait_For_Close

This transition shall occur after one of the following confirmations is received:

- a) an Open Failed (Open Timeout Occurred);
- b) an Open Failed (Break Received); or
- c) an Open Failed (Arb Stopped).

7.2.3.4 PL_PM3:Connected state

7.2.3.4.1 PL_PM3:Connected state description

If this state was entered from the PL_PM1:Idle state, then this state shall send a Connection Opened message to the PL_OC state machine that includes as an argument the received OPEN address frame.

If:

- a) the protocol for the connection is SSP, the port is an SSP target port, the Disconnect-Reconnect mode page is implemented, and the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2) is not set to zero;
- b) the protocol for the connection is SMP and the port is an SMP target port; or
- c) the protocol for the connection is STP, the port is an STP target port, and the STP MAXIMUM CONNECT TIME LIMIT field is not set to zero in the SMP REPORT GENERAL response (see 9.4.3.4),

then, upon entry into this state, this state shall:

- 1) create a Maximum Connect Time Limit timer;
- 2) initialize the Maximum Connect Time Limit timer as specified in table 115 (see 7.2.3.1); and
- 3) start the Maximum Connect Time Limit timer.

If:

- a) the protocol for the connection is SSP, the port is an SSP initiator port, and the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2) for the destination SSP target port is not set to zero; or
- b) the protocol for the connection is STP, the port is an STP initiator port, and the STP MAXIMUM CONNECT TIME LIMIT field is not set to zero in the SMP REPORT GENERAL response (see 9.4.3.4) for the destination STP target port,

then, upon entry into this state, this state may:

- 1) create a Maximum Connect Time Limit timer;
- 2) initialize the Maximum Connect Time Limit timer as specified in table 115 (see 7.2.3.1); and
- 3) start the Maximum Connect Time Limit timer.

If:

- a) the protocol for the connection is SSP, the port is an SSP target port, and the BUS INACTIVITY TIME LIMIT field in the Disconnect-Reconnect mode page (see 9.2.7.2) is set to a non-zero value; or
- b) the protocol for the connection is STP, the port is an STP initiator port, and the STP BUS INACTIVITY TIME LIMIT field is not set to zero in the SMP REPORT GENERAL response for the destination STP target port,

then, upon entry into this state, this state shall:

- 1) create a Bus Inactivity Time Limit timer;
- 2) initialize the Bus Inactivity Time Limit timer as specified in table 115 (see 7.2.3.1); and
- 3) start the Bus Inactivity Time Limit timer.

If a Bus Inactivity Time Limit timer has been created and:

- a) the connection is SSP and this state receives a Tx Frame message; or
- b) the connection is STP and the phy is not both transmitting and receiving SATA_SYNC,

then this state shall:

- 1) stop the Bus Inactivity Time Limit timer, if it is running;
- 2) initialize the Bus Inactivity Time Limit timer as specified in table 115 (see 7.2.3.1); and
- 3) start the Bus Inactivity Time Limit timer.

If this state receives a Tx Frame message, then this state shall send a Tx Frame request to the link layer. The following arguments from the Tx Frame message shall be included with the Tx Frame request:

- a) the frame to be transmitted; and
- b) if this port is an SSP port, Balance Required or Balance Not Required.

For STP connections, this state connects the STP transport layer to the STP link layer.

If a Bus Inactivity Time Limit timer expires:

- a) if the connection is SSP and there is no Tx Frame request outstanding (i.e., this state is not waiting for an ACK Received or NAK Received confirmation), then this state shall send a Close Connection request to the link layer;
- b) if the connection is SSP and there is a Tx Frame request outstanding (i.e., this state is waiting for an ACK Received or NAK Received confirmation), then this state shall send a Close Connection request to the link layer after receiving an ACK Received or NAK Received confirmation; or
- c) if the connection is STP, then this state shall send a Close Connection request to the link layer.

If a Maximum Connect Time Limit timer expires:

- a) if the connection is SSP and there is no Tx Frame request outstanding (i.e., this state is not waiting for an ACK Received or NAK Received confirmation), then this state shall send a Close Connection request to the link layer;
- b) if the connection is SSP and there is a Tx Frame request outstanding (i.e., this state is waiting for an ACK Received or NAK Received confirmation), then this state shall send a Close Connection request to the link layer after receiving an ACK Received or NAK Received confirmation;
- c) if the connection is SMP, then this state shall send an SMP Transmit Break request to the link layer; or

- d) if the connection is STP, then this state shall send a Close Connection request to the link layer after the phy is both transmitting and receiving SATA_SYNC.

If this state receives a Tx Frame message after sending a Close Connection request but before receiving a Connection Closed confirmation, then this state shall send a Retry Frame message to the PL_OC state machine.

If this state receives a Frame Received confirmation, then this state shall send a Frame Received confirmation to the transport layer. The confirmation shall include the arguments received with the confirmation (e.g., the frame).

If this state receives an ACK Transmitted confirmation, then this state shall send an ACK Transmitted confirmation to the transport layer including the initiator port transfer tag of the frame that was ACKed.

If this state receives a Frame Transmitted confirmation, then this state shall send:

- a) a Transmission Status (Frame Transmitted) confirmation to the transport layer; and
- b) a Transmission Status (Frame Transmitted) message to the PL_OC state machine.

If this state receives an ACK Received confirmation, then this state shall send:

- a) a Transmission Status (ACK Received) confirmation to the transport layer; and
- b) a Transmission Status (ACK Received) message to the PL_OC state machine.

If this state receives a NAK Received confirmation, then this state shall send:

- a) a Transmission Status (NAK Received) confirmation to the transport layer; and
- b) a Transmission Status (NAK Received) message to the PL_OC state machine.

If this state receives an ACK/NAK Timeout confirmation, then this state shall send:

- a) a Transmission Status (ACK/NAK Timeout) confirmation to the transport layer; and
- b) a Transmission Status (ACK/NAK Timeout) message to the PL_OC state machine.

If this state receives a Cancel message, then this state shall:

- a) discard all Tx Frame requests for the specified initiator port transfer tag;
- b) send a Transmission Status (Cancel Acknowledge) confirmation to the transport layer including the destination SAS address and the initiator port transfer tag as arguments; and
- c) discard any subsequent confirmations for previous Tx Frame requests sent for the initiator port transfer tag.

If this state receives a Close Connection message from the PL_OC state machine, then this state shall send a Close Connection request to the link layer.

If this state receives one of the following:

- a) a Connection Closed (Normal) confirmation;
- b) a Connection Closed (Close Timeout) confirmation;
- c) a Connection Closed (Break Requested) confirmation;
- d) a Connection Closed (Break Received) confirmation; or
- e) a Connection Closed (Transition to Idle) confirmation,

then this state shall send a Connection Closed message to the PL_OC state machine including the argument received with the confirmation.

If this state receives a Connection Closed (Transition to Idle) confirmation after receiving one of the following:

- a) a Connection Closed (Break Received) confirmation; or
- b) a Connection Closed (Break Requested) confirmation,

then this state shall send a Transmission Status (Break Received) confirmation to the transport layer.

If this state receives a Connection Closed (Normal) confirmation, a Connection Closed (Transition to Idle) confirmation, or a Phy Disabled confirmation after sending a Transmission Status (Frame Transmitted) confirmation, but before this state receives an ACK Received or NAK Received confirmation, then this state shall send:

- a) a Transmission Status (Connection Lost Without ACK/NAK) confirmation to the transport layer; and

- b) a Transmission Status (Connection Lost Without ACK/NAK) message to the PL_OC state machine.

If this state receives a Connection Closed (Normal) confirmation, a Connection Closed (Transition to Idle) confirmation, or a Phy Disabled confirmation after sending a Tx Frame request but before receiving a Frame Transmitted confirmation, then this state shall send a Retry Frame message to the PL_OC state machine.

If this state receives a Connection Closed confirmation during an SMP connection, then this state shall send a Connection Closed confirmation to the transport layer.

If this state receives a Credit Timeout confirmation, then this state shall send a Retry Frame message to the PL_OC state machine.

A Retry Frame message shall include the following arguments from the Tx Frame message:

- a) Initiator Port Bit;
- b) Protocol;
- c) Connection Rate;
- d) Initiator Connection Tag;
- e) Destination SAS Address;
- f) Source SAS Address; and
- g) Frame.

After this state receives a DONE Received (Normal) or DONE Received (Credit Timeout) confirmation, if this state does not receive a Tx Frame message within 1 ms, then this state shall send a Disable Tx Frames message to the PL_OC state machine.

If this state receives a DONE Received (ACK/NAK Timeout) or DONE Transmitted confirmation, then this state shall send a Disable Tx Frames message to the PL_OC state machine.

If this state receives an SMP Transmit Break message, then this state shall send an SMP Transmit Break request to the link layer.

If this state receives a Hard Reset message, then this state shall terminate all operations.

7.2.3.4.2 Transition PL_PM3:Connected to PL_PM1:Idle

This transition shall occur after:

- a) a Connection Closed (Transition to Idle) message is sent to the PL_OC state machine; or
- b) all operations are terminated after receiving a Hard Reset message.

7.2.3.5 PL_PM4:Wait_For_Close state

7.2.3.5.1 PL_PM4:Wait_For_Close state description

After receiving a Connection Closed (Transition to Idle) confirmation:

- a) if this state was entered as the result of the PL_PM2:Req_Wait state receiving an Open Failed (Open Timeout Occurred) confirmation, then this state shall send a Retry Open (Open Timeout Occurred) message to the PL_OC state machine; or
- b) if this state was entered as the result of the PL_PM2:Req_Wait state receiving an Open Failed (Break Received) confirmation, then this state shall send an Retry Open (Break Received) message to the PL_OC state machine.

The Retry Open message shall include the following arguments:

- a) Initiator Port Bit set to the value received with the Tx Open message;
- b) Protocol set to the value received with the Tx Open message;
- c) Connection Rate set to the value received with the Tx Open message;
- d) Initiator Connection Tag set to the value received with the Tx Open message;
- e) Destination SAS Address set to the value received with the Tx Open message;
- f) Source SAS Address set to the value received with the Tx Open message;
- g) Pathway Blocked Count set to the value received with the Tx Open message; and
- h) Arbitration Wait Time set to the value of the Arbitration Wait Time timer.

If this state receives a Connection Closed confirmation and the connection request was for an SMP connection, then this state shall send a Connection Closed confirmation to the transport layer.

After receiving a Connection Closed (Transition to Idle) confirmation, if this state was entered after the PL_PM2:Req_Wait state received an Open Failed (Arb Stopped) confirmation (i.e., as the result of the PL_PM2:Req_Wait state sending a Stop Arb request), then this state shall send an Unable to Connect (Arb Stopped) message to the PL_OC state machine.

If this state receives a Hard Reset message, then this state shall terminate all operations.

7.2.3.5.2 Transition PL_PM4:Wait_For_Close to PL_PM1:Idle

This transition shall occur after:

- a) a Retry Open or Unable To Connect message is sent to the PL_OC state machine; or
- b) all operations are terminated after receiving a Hard Reset message.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

8 Transport layer

8.1 Transport layer overview

The transport layer defines frame formats and how frames are processed by this layer. Transport layer state machines interface to the application layer and port layer and construct and parse frame contents. For SSP, the transport layer only receives frames from the port layer for which an ACK is going to be transmitted by the link layer.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Byte\Bit	7	6	5	4	3	2	1	0
0	FRAME TYPE							
1	(MSB) HASHED DESTINATION SAS ADDRESS (LSB)							
3								
4	Reserved							
5	(MSB) HASHED SOURCE SAS ADDRESS (LSB)							
7								
8	Reserved							
9	Reserved							
10	Reserved			TLR CONTROL		RETRY DATA FRAMES	RETRANSMIT	CHANGING DATA POINTER
11	Reserved						NUMBER OF FILL BYTES	
12	Reserved							
13	Reserved							
15								
16	(MSB) INITIATOR PORT TRANSFER TAG (LSB)							
17								
18	(MSB) TARGET PORT TRANSFER TAG (LSB)							
19								
20	(MSB) DATA OFFSET (LSB)							
23								
24	INFORMATION UNIT							
m	(e.g., see table 120, table 122, table 124, table 125, or table 126)							
	Fill bytes, if needed							
n - 3	(MSB) CRC (LSB)							
n								

Table 118 defines the FRAME TYPE field, which defines the format of the INFORMATION UNIT field.

Table 118 — FRAME TYPE field

Code	Name of frame	Type of information unit	Originator	Information unit size (bytes)	Reference
01h	DATA frame (i.e., write DATA frame or read DATA frame)	Data information unit (i.e., write Data information unit or read Data information unit)	SSP initiator port or SSP target port	1 to 1 024	8.2.2.4
05h	XFER_RDY frame	Transfer Ready information unit	SSP target port	12	8.2.2.3
06h	COMMAND frame	Command information unit	SSP initiator port	28 to 280	8.2.2.1
07h	RESPONSE frame	Response information unit	SSP target port	24 to 1 024	8.2.2.5
16h	TASK frame	Task Management Function information unit	SSP initiator port	28	8.2.2.2
F0h to FFh	Vendor specific				
All others	Reserved				

The HASHED DESTINATION SAS ADDRESS field contains the hashed value of the destination SAS address (see 4.2.5). See 8.2.6.2.2 and 8.2.6.3.2 for transport layer requirements on checking this field.

The HASHED SOURCE SAS ADDRESS field contains the hashed value of the source SAS address (see 4.2.5). See 8.2.6.2.2 and 8.2.6.3.2 for transport layer requirements on checking this field.

Table 119 defines the TLR CONTROL field for COMMAND frames. The TLR CONTROL field is reserved for all other frame types.

Table 119 — TLR CONTROL field for COMMAND frames

Code ^a	Description
00b or 11b	The SSP target port shall use the TRANSPORT LAYER RETRIES bit in the Protocol-Specific Logical Unit mode page (see 9.2.7.3) to enable or disable transport layer retries for this command as follows: a) if the TRANSPORT LAYER RETRIES bit is set to one, then the SSP target port shall set the RETRY DATA FRAMES bit to one in any XFER_RDY frames that the SSP target port transmits for this command; or b) if the TRANSPORT LAYER RETRIES bit is set to zero, then the SSP target port shall set the RETRY DATA FRAMES bit to zero in any XFER_RDY frames that the SSP target port transmits for this command.
01b	The SSP target port may enable transport layer retries for this command. If the SSP target port enables transport layer retries, then it shall set the RETRY DATA FRAMES bit to one in any XFER_RDY frames that it transmits for this command. If the SSP target port does not enable transport layer retries, then it shall set the RETRY DATA FRAMES bit to zero in any XFER_RDY frames that it transmits for this command.
10b	The SSP target port shall: a) disable transport layer retries for this command; and b) set the RETRY DATA FRAMES bit to zero in any XFER_RDY frames that it transmits for this command.
^a If the SSP target port receives a non-zero value in the TLR CONTROL field and does not support non-zero values in the TLR CONTROL field, then it shall reply with a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 02h (i.e., INVALID FRAME).	

If an SSP initiator port supports transport layer retries, then it shall set the TLR CONTROL field to 01b in each COMMAND frame that it sends unless it has determined that the I_T_L nexus does not support the TLR CONTROL field.

If an SSP initiator port does not support transport layer retries, then it shall set the TLR CONTROL field to 10b in each COMMAND frame that it sends unless it has determined that the I_T_L nexus does not support the TLR CONTROL field.

An SSP initiator port determines that an I_T_L nexus does not support the TLR CONTROL field if it sends a COMMAND frame with the TLR CONTROL field set to 01b or 10b and receives a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 02h (i.e., INVALID FRAME). After determining that an I_T_L nexus does not support the TLR CONTROL field, the SSP initiator port shall set the TLR CONTROL field to 00b for subsequent COMMAND frames for that I_T_L nexus.

NOTE 67 Initiator ports compliant with SAS-1.1 always set the TLR CONTROL field to 00b.

NOTE 68 The TLR CONTROL SUPPORTED field in the Protocol-Specific Logical Unit Information VPD page (see 9.2.11.3) indicates if the SSP target port supports the TLR CONTROL field set to a non-zero value.

An SSP target port sets the RETRY DATA FRAMES bit in an XFER_RDY frame based on the TLR CONTROL field received in the COMMAND frame for the command (see table 119) and the TRANSPORT LAYER RETRIES bit in the Protocol-Specific Logical Unit mode page (see 9.2.7.3).

A RETRY DATA FRAMES bit set to one in an XFER_RDY frame specifies that the SSP initiator port shall enable transport layer retries for write DATA transfers related to this XFER_RDY.

A RETRY DATA FRAMES bit set to zero in an XFER_RDY frame specifies that the SSP initiator port shall disable transport layer retries for write DATA transfers related to this XFER_RDY.

The RETRY DATA FRAMES bit is reserved for frames other than XFER_RDY frames.

A RETRANSMIT bit set to one specifies that the frame is a retransmission after the SSP port failed in its previous attempt to transmit the frame (e.g., the SSP port received a NAK for the frame transmitted in the previous attempt). The RETRANSMIT bit is set to one for TASK frames, RESPONSE frames, and XFER_RDY frames under the conditions defined in 8.2.4 and shall be set to zero for all other frame types.

A CHANGING DATA POINTER bit set to one specifies that the frame is a retransmission after the SSP port failed in its previous attempt to transmit the frame or a subsequent frame and the DATA OFFSET field of the frame may not be sequentially increased from that of the previous frame. The CHANGING DATA POINTER bit is set to one for DATA frames under the conditions defined in 8.2.4.5 and shall be set to zero for all other frame types.

The NUMBER OF FILL BYTES field specifies the number of fill bytes between the INFORMATION UNIT field and the CRC field. The NUMBER OF FILL BYTES field shall be set to 00b for all frame types except DATA frames as specified in 8.2.2.4 and RESPONSE frames as specified in 8.2.2.5 (i.e., all other frame types are four-byte aligned).

The INITIATOR PORT TRANSFER TAG field contains a value that allows the SSP initiator port to establish a context for commands and task management functions.

For COMMAND frames and TASK frames, the SSP initiator port shall set the INITIATOR PORT TRANSFER TAG field to a value that is unique for the I_T nexus established by the connection (see 6.13). An SSP initiator port shall not reuse the same initiator port transfer tag when transmitting COMMAND frames or TASK frames to different LUNs in the same SSP target port. An SSP initiator port may reuse an initiator port transfer tag when transmitting frames to different SSP target ports. An SSP initiator port does not reuse an initiator port transfer tag until it receives indication from the SSP target port that the initiator port transfer tag is no longer in use (see 8.2.4, 8.2.5, and 9.2.2).

The INITIATOR PORT TRANSFER TAG field in a COMMAND frame contains the command identifier defined in SAM-5. The INITIATOR PORT TRANSFER TAG field in a TASK frame is the association between a Send Task Management Request (see 9.2.1.12) and a Received Task Management Function Executed (see 9.2.1.15). The number space used in the INITIATOR PORT TRANSFER TAG fields is shared across COMMAND frames and TASK frames (e.g., if an initiator port transfer tag is used for a COMMAND frame, then it is not also used for a concurrent TASK frame).

For DATA, XFER_RDY, and RESPONSE frames, the SSP target port shall set the INITIATOR PORT TRANSFER TAG field to the initiator port transfer tag of the command or task management function to which the frame pertains.

The TARGET PORT TRANSFER TAG field provides an optional method for an SSP target port to establish the write data context when receiving a write DATA frame (i.e., determine the command to which the write data corresponds). Unlike the INITIATOR PORT TRANSFER TAG field, which was assigned by the SSP initiator port, the TARGET PORT TRANSFER TAG field in a write DATA frame contains a value assigned by the SSP target port that was delivered to the SSP initiator port in the XFER_RDY frame requesting the write data.

NOTE 69 The TARGET PORT TRANSFER TAG field may be useful when the SSP target port has more than one XFER_RDY frame outstanding (i.e., the SSP target port has transmitted an XFER_RDY frame for each of two or more commands and has not yet received all the write data for them).

SSP target ports may set the TARGET PORT TRANSFER TAG field to any value when transmitting any SSP frame. SSP target ports that use this field should set the TARGET PORT TRANSFER TAG field in every XFER_RDY frame to a value that is unique for the L_Q portion of the I_T_L_Q nexus (i.e., that is unique for every XFER_RDY that is outstanding from the SSP target port).

SSP initiator ports shall set the TARGET PORT TRANSFER TAG field as follows:

- a) for each write DATA frame that is sent in response to an XFER_RDY frame, the SSP initiator port shall set the TARGET PORT TRANSFER TAG field to the value that was in the corresponding XFER_RDY frame;
- b) for each write DATA frame that is sent containing first burst data (see 8.2.2.4), the SSP initiator port shall set the TARGET PORT TRANSFER TAG field to FFFFh; and
- c) for frames other than write DATA frames, the SSP initiator port shall set the TARGET PORT TRANSFER TAG field to FFFFh.

For DATA frames, the DATA OFFSET field is described in 8.2.2.4. For all other frame types, the DATA OFFSET field shall be ignored.

The INFORMATION UNIT field contains the information unit, the format of which is defined by the FRAME TYPE field (see table 118). The maximum size of the INFORMATION UNIT field is 1 024 bytes, making the maximum size of the frame 1 052 bytes (i.e., 24 bytes of header + 1 024 bytes of data + 4 bytes of CRC).

Fill bytes shall be included after the INFORMATION UNIT field so the CRC field is aligned on a four byte boundary. The number of fill bytes are specified by the NUMBER OF FILL BYTES field. The contents of the fill bytes are vendor specific.

The CRC field contains a CRC value (see 6.5) that is computed over the entire SSP frame prior to the CRC field including the fill bytes (i.e., all data dwords between the SOF and the CRC field). The CRC field is checked by the link layer (see 6.17), not the transport layer.

8.2.2 Information units

8.2.2.1 COMMAND frame - Command information unit

A COMMAND frame is sent by an SSP initiator port to request that a command be processed by the device server in a logical unit (see 8.2.3.3, 8.2.3.4, 8.2.3.5, and 8.2.3.6).

Table 120 defines the Command information unit used in a COMMAND frame.

Table 120 — COMMAND frame - Command information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	LOGICAL UNIT NUMBER							
7								
8	Reserved							
9	ENABLE FIRST BURST	COMMAND PRIORITY				TASK ATTRIBUTE		
10	Reserved							
11	ADDITIONAL CDB LENGTH (n dwords)						Reserved	
12	CDB							
27								
28	ADDITIONAL CDB BYTES							
27+n×4								

The LOGICAL UNIT NUMBER field specifies the logical unit number of the logical unit to which the task router shall route the command. The structure of the LOGICAL UNIT NUMBER field shall be as defined in SAM-5. If the addressed logical unit does not exist, then the task router shall follow the rules for selection of incorrect logical units defined in SAM-5.

An ENABLE FIRST BURST bit set to one specifies that the SSP target port shall expect first burst data for the command as defined by the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 9.2.7.2). An ENABLE FIRST BURST bit set to zero specifies that the SSP target port shall not expect first burst data for the

command (i.e., that the FIRST BURST SIZE field in the Disconnect-Reconnect mode page shall be ignored). Application clients shall only set the ENABLE FIRST BURST bit to one if:

- a) the FIRST BURST SIZE field in the Disconnect-Reconnect mode page is set to a value other than 0000h; and
- b) the logical unit and SSP target port comply with this standard (e.g., as reported in the standard INQUIRY data version descriptors (see SPC-4)).

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

The TASK ATTRIBUTE field is defined in table 121.

Table 121 — TASK ATTRIBUTE field

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

The ADDITIONAL CDB LENGTH field contains the length in dwords (i.e., four bytes) of the ADDITIONAL CDB field.

The CDB and ADDITIONAL CDB BYTES fields together contain the CDB to be interpreted by the addressed logical unit. Any bytes after the end of the actual CDB within the two fields shall be ignored (e.g., a six-byte CDB occupies the first six bytes of the CDB field, the remaining ten bytes of the CDB field are ignored, and the ADDITIONAL CDB BYTES field is not present).

The contents of the CDB are defined in the SCSI command standards (e.g., SPC-4).

8.2.2.2 TASK frame - Task Management Function information unit

A TASK frame is sent by an SSP initiator port to request that a task management function be processed by the task manager in a logical unit (see 8.2.3.2).

Table 122 defines the Task Management Function information unit used in a TASK frame.

Table 122 — TASK frame - Task Management Function information unit

Byte\Bit	7	6	5	4	3	2	1	0						
0	LOGICAL UNIT NUMBER													
7														
8	Reserved													
9	Reserved													
10	TASK MANAGEMENT FUNCTION													
11	Reserved													
12	(MSB)	INITIATOR PORT TRANSFER TAG TO MANAGE												
13								(LSB)						
14	Reserved													
27														

The LOGICAL UNIT NUMBER field specifies the logical unit number of the logical unit, if any, to which the task router shall route the task management function. The structure of the LOGICAL UNIT NUMBER field shall be as defined in SAM-5. If the addressed logical unit does not exist, then the task router shall return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and its RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER.

Table 123 defines the TASK MANAGEMENT FUNCTION field.

Table 123 — TASK MANAGEMENT FUNCTION field (part 1 of 2)

Code	Task management function	Support ^a	Uses LOGICAL UNIT NUMBER field	Uses INITIATOR PORT TRANSFER TAG TO MANAGE field	Description ^b
01h	ABORT TASK	M	yes	yes	The task manager shall perform the ABORT TASK task management function with L set to the value of the LOGICAL UNIT NUMBER field and Q set to the value of the INITIATOR PORT TRANSFER TAG TO MANAGE field (see SAM-5).
02h	ABORT TASK SET	M	yes	no	The task manager shall perform the ABORT TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
04h	CLEAR TASK SET	M	yes	no	The task manager shall perform the CLEAR TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
08h	LOGICAL UNIT RESET	M	yes	no	The logical unit shall perform the LOGICAL UNIT RESET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
10h	I_T NEXUS RESET	M	no	no	The SSP target device shall perform the I_T NEXUS RESET task management function (see SAM-5).
20h	Reserved				
40h	CLEAR ACA	X	yes	no	The task manager shall perform the CLEAR ACA task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
80h	QUERY TASK	M	yes	yes	The task manager shall perform the QUERY TASK task management function with L set to the value of the LOGICAL UNIT NUMBER field and Q set to the value of the INITIATOR PORT TRANSFER TAG TO MANAGE field (see SAM-5).
^a M = implementation is mandatory, X = implementation requirements are specified by SAM-5. ^b The task manager, logical unit, or SSP target device shall perform the specified task management function with the I and T arguments set to the SSP initiator port and SSP target port involved in the connection used to deliver the TASK frame.					

Table 123 — TASK MANAGEMENT FUNCTION field (part 2 of 2)

Code	Task management function	Support ^a	Uses LOGICAL UNIT NUMBER field	Uses INITIATOR PORT TRANSFER TAG TO MANAGE field	Description ^b
81h	QUERY TASK SET	M	yes	no	The task manager shall perform the QUERY TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
82h	QUERY ASYNCHRONOUS EVENT	M	yes	no	The task manager shall perform the QUERY ASYNCHRONOUS EVENT task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-5).
All others	Reserved				

^a M = implementation is mandatory, X = implementation requirements are specified by SAM-5.
^b The task manager, logical unit, or SSP target device shall perform the specified task management function with the I and T arguments set to the SSP initiator port and SSP target port involved in the connection used to deliver the TASK frame.

If the TASK MANAGEMENT FUNCTION field contains a reserved or unsupported value, then the task manager shall return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and its RESPONSE CODE field set to TASK MANAGEMENT FUNCTION NOT SUPPORTED.

If the TASK MANAGEMENT FUNCTION field is set to ABORT TASK or QUERY TASK, then the INITIATOR PORT TRANSFER TAG TO MANAGE field specifies the INITIATOR PORT TRANSFER TAG value from the COMMAND frame that contained the command to be aborted or checked. For all other task management functions, the INITIATOR PORT TRANSFER TAG TO MANAGE field shall be ignored.

8.2.2.3 XFER_RDY frame - Transfer Ready information unit

A XFER_RDY frame is sent by an SSP target port to request write data from the SSP initiator port during a write command or a bidirectional command (see 8.2.3.4 and 8.2.3.6).

Table 124 defines the Transfer Ready information unit used in a XFER_RDY frame.

Table 124 — XFER_RDY frame - Transfer Ready information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
3	REQUESTED OFFSET							(LSB)
4	(MSB)							
7	WRITE DATA LENGTH							(LSB)
8	Reserved							
11								

The REQUESTED OFFSET field contains the application client buffer offset of the segment of write data in the data-out buffer that the SSP initiator port may transmit to the logical unit using write DATA frames. The requested offset shall be a multiple of four (i.e., each write DATA frame shall begin transferring data on a dword boundary).

The REQUESTED OFFSET field shall be set to 00000000h for the first XFER_RDY frame of a command unless:

- a) the ENABLE FIRST BURST bit in the COMMAND frame (see 8.2.2.1) was set to one; and
- b) the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 9.2.7.2.5) is set to a value other than 0000h.

If the ENABLE FIRST BURST bit in the COMMAND frame (see 8.2.2.1) was set to one, then in the initial XFER_RDY frame for the command, the SSP target port shall set the REQUESTED OFFSET field to the application client buffer offset of the segment of write data following the first burst data defined by the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 9.2.7.2.5).

If any additional XFER_RDY frames are required for the command and transport layer retries are not being used, then the REQUESTED OFFSET field shall be set to the sum of the requested offset and write data length of the previous XFER_RDY frame.

The WRITE DATA LENGTH field contains the number of bytes of write data the SSP initiator port may transmit to the logical unit using write DATA frames from the application client data-out buffer starting at the requested offset. The SSP target port shall set the WRITE DATA LENGTH field to a value greater than or equal to 00000001h. If the value in the MAXIMUM BURST SIZE field in the Disconnect-Reconnect mode page is not 0000h, then the SSP target port shall set the WRITE DATA LENGTH field to a value less than or equal to the number of bytes specified by the MAXIMUM BURST SIZE field (see 9.2.7.2.4).

If an SSP target port transmits an XFER_RDY frame containing a WRITE DATA LENGTH field set to a value that is not divisible by four, then the SSP target port shall not transmit any subsequent XFER_RDY frames for that command (i.e., only the last XFER_RDY for a command may request a non-dword multiple write data length).

The value in the REQUESTED OFFSET field plus the value of the WRITE DATA LENGTH field shall not be greater than 1_00000000h (i.e., a SCSI command shall not transfer more than 2^{32} bytes of write data).

8.2.2.4 DATA frame - Data information unit

During a write command or a bidirectional command (see 8.2.3.4 and 8.2.3.6), one or more write DATA frames are sent by an SSP initiator port to deliver write data.

During a read command or a bidirectional command (see 8.2.3.5 and 8.2.3.6), one or more read DATA frames are sent by an SSP target port to deliver read data.

Table 125 defines the Data information unit used in a DATA frame.

Table 125 — DATA frame - Data information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	DATA							
n								

The DATA field contains the read data (i.e., data to the application client's data-in buffer) or write data (i.e., data from the application client's data-out buffer).

The size of the DATA field (i.e., the data length) is determined by subtracting the following values from the DATA frame size (i.e., the number of bytes between SOF and EOF (see 6.17.3)):

- a) the number of bytes in frame header (i.e., 24);
- b) the number of bytes in the CRC field (i.e., 4); and
- c) the number of fill bytes, specified by the NUMBER OF FILL BYTES field in the frame header (see 8.2.1).

The maximum size of the Data information unit (i.e., the DATA field) is the maximum size of any information unit in an SSP frame (see 8.2.1). The minimum size of the Data information unit is one byte.

An SSP initiator port shall only transmit a write DATA frame:

- a) in response to an XFER_RDY frame; or
- b) after transmitting a COMMAND frame if the ENABLE FIRST BURST bit in the COMMAND frame was set to one (see 8.2.2.1) and the FIRST BURST SIZE field in the Disconnect-Reconnect mode page is set to a value other than 0000h (see 9.2.7.2.5).

If the value in the MAXIMUM BURST SIZE field on the Disconnect-Reconnect mode page is not zero, then the maximum amount of data that is transferred at one time by an SSP target port per I_T_L_Q nexus is limited by the value in the MAXIMUM BURST SIZE field (see 9.2.7.2.4).

A write DATA frame shall only contain write data for a single XFER_RDY frame.

An SSP initiator port shall set the NUMBER OF FILL BYTES field to 00b in the frame header (see 8.2.1) in all write DATA frames that it transmits in response to an XFER_RDY frame except the last write DATA frame for that XFER_RDY frame. An SSP initiator port may set the NUMBER OF FILL BYTES field to a non-zero value in the last DATA frame that it transmits in response to an XFER_RDY.

NOTE 70 Combined with the restrictions on the WRITE DATA LENGTH field in the XFER_RDY frame (see 8.2.2.3), this ensures that only the last write DATA frame for a command may have data with a length that is not a multiple of four).

An SSP target port shall set the NUMBER OF FILL BYTES field to 00b in the frame header (see 8.2.1) in all read DATA frames for a command except the last read DATA frame for that command. The SSP target port may set the NUMBER OF FILL BYTES field to a non-zero value in the last read DATA frame for a command (i.e., only the last read DATA frame for a command may contain data with a length that is not a multiple of four).

An SSP initiator port shall not transmit a write DATA frame for a given I_T_L_Q nexus after it has sent a TASK frame that terminates that command (e.g., an ABORT TASK).

The DATA OFFSET field in the frame header (see 8.2.1) contains the application client buffer offset as described by SAM-5. For read DATA frames, this is the offset into the application client's data-in buffer; for write DATA frames, this is the offset into the application client's data-out buffer. The data offset shall be a multiple of four (i.e., each DATA frame shall transfer data beginning on a dword boundary).

The DATA OFFSET field shall be set to 00000000h in the initial read DATA frame for a command. If any additional read DATA frames are required for the command and transport layer retries are not being used, then the DATA OFFSET field shall be set to the sum of the data offset and data length of the previous read DATA frame.

The DATA OFFSET field shall be set to 00000000h in the initial write DATA frame for a command. If any additional write DATA frames are required for the command and transport layer retries are not being used,

Table 127 defines the DATAPRES field, which specifies the format and content of the STATUS field, the STATUS QUALIFIER field, the SENSE DATA LENGTH field, the RESPONSE DATA LENGTH field, the RESPONSE DATA field, and the SENSE DATA field.

Table 127 — DATAPRES field

Code	Name	Description	Reference
00b	NO_DATA	No response data or sense data present	8.2.2.5.2
01b	RESPONSE_DATA	Response data present	8.2.2.5.3
10b	SENSE_DATA	Sense data present	8.2.2.5.4
11b	Reserved		

An SSP target port shall set the DATAPRES field to NO_DATA in a RESPONSE frame sent for a command that completes without response data or sense data.

An SSP target port shall set the DATAPRES field to RESPONSE_DATA in a RESPONSE frame sent for:

- every TASK frame; and
- every COMMAND frame for which errors occur while the transport layer is processing the frame (see 8.2.5.3).

An SSP target port shall set the DATAPRES field to SENSE_DATA in a RESPONSE frame sent for a command that completes with sense data to return (e.g., CHECK CONDITION status).

If the DATAPRES field is set to a reserved value, then the SSP initiator port shall discard the RESPONSE frame.

8.2.2.5.2 Response information unit - NO_DATA format

If the DATAPRES field is set to NO_DATA, then:

- the SSP target port shall set the STATUS field to the status code for a completed command (see SAM-5 for a list of status codes);
- the SSP target port shall set the STATUS QUALIFIER field to the status qualifier for the command (see SAM-5);
- the SSP target port shall set the SENSE DATA LENGTH field to 00000000h and the RESPONSE DATA LENGTH field to 00000000h;
- the SSP initiator port shall ignore the SENSE DATA LENGTH field and the RESPONSE DATA LENGTH field; and
- the SSP target port shall not include the SENSE DATA field and the RESPONSE DATA field.

8.2.2.5.3 Response information unit - RESPONSE_DATA format

If the DATAPRES field is set to RESPONSE_DATA, then:

- the SSP target port shall set the STATUS field to zero, the STATUS QUALIFIER field to 0000h, and the SENSE DATA LENGTH field to 00000000h;
- the SSP initiator port shall ignore the STATUS field, the STATUS QUALIFIER field, and the SENSE DATA LENGTH field;
- the SSP target port shall not include the SENSE DATA field;
- the SSP target port shall set the RESPONSE DATA LENGTH field to 00000004h; and
- the SSP target port shall include the RESPONSE DATA field.

Table 128 defines the RESPONSE DATA field. The RESPONSE DATA field shall be present if the SSP target port detects any of the conditions described by a non-zero value in the RESPONSE CODE field and shall be present for a RESPONSE frame sent in response to a TASK frame.

Table 128 — RESPONSE DATA field

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDITIONAL RESPONSE INFORMATION							
2								
3	RESPONSE CODE							

The ADDITIONAL RESPONSE INFORMATION field contains additional response information for certain task management functions (e.g., QUERY ASYNCHRONOUS EVENT) as defined in SAM-5. If the task management function does not define additional response information or the logical unit does not support additional response information, then the SSP target port shall set the ADDITIONAL RESPONSE INFORMATION field to 000000h.

Table 129 defines the RESPONSE CODE field, which specifies the error condition or the completion status of a task management function. See 9.2.1.5 and 9.2.1.15 for the mapping of these response codes to SCSI service responses.

Table 129 — RESPONSE CODE field

Code	Description
00h	TASK MANAGEMENT FUNCTION COMPLETE ^a
02h	INVALID FRAME
04h	TASK MANAGEMENT FUNCTION NOT SUPPORTED ^a
05h	TASK MANAGEMENT FUNCTION FAILED ^a
08h	TASK MANAGEMENT FUNCTION SUCCEEDED ^a
09h	INCORRECT LOGICAL UNIT NUMBER ^a
0Ah	OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED ^b
All others	Reserved
^a Only valid when responding to a TASK frame. ^b Returned in case of command/task management function or task management function/task management function initiator port transfer tag conflicts.	

8.2.2.5.4 Response information unit - SENSE_DATA format

If the DATAPRES field is set to SENSE_DATA, then:

- the SSP target port shall set the STATUS field to the status code for a completed command (see SAM-5 for a list of status codes);
- the SSP target port shall set the STATUS QUALIFIER field to the status qualifier for the command (see SAM-5);
- the SSP target port shall set the RESPONSE DATA LENGTH field to 00000000h;
- the SSP initiator port shall ignore the RESPONSE DATA LENGTH field;
- the SSP target port shall not include the RESPONSE DATA field;
- the SSP target port shall set the SENSE DATA LENGTH field to a non-zero value indicating the number of bytes in the SENSE DATA field. The value in the SENSE DATA LENGTH field shall not be greater than 1 000 (see table 118 in 8.2.1); and
- the SSP target port shall set the SENSE DATA field to the sense data (see SAM-5).

The value in the SENSE DATA LENGTH field is not required to be a multiple of four. If the value is not a multiple of four, then the value in the NUMBER OF FILL BYTES field in the SSP frame header is non-zero and fill bytes are present.

8.2.3 Sequences of SSP frames

8.2.3.1 Sequences of SSP frames overview

Table 130 lists the sequences of SSP frames supporting the SCSI transport protocol services described in 9.2.1.

Table 130 — Sequences of SSP frames

Sequence	Reference
Task management function	8.2.3.2
Non-data command	8.2.3.3
Write command	8.2.3.4
Read command	8.2.3.5
Bidirectional command	8.2.3.6

When multiple commands and/or task management functions are outstanding, frames from each of the individual sequences may be interleaved in any order. RESPONSE frames may be returned in any order (i.e., the order in which TASK frames and COMMAND frames are sent has no effect on the order that RESPONSE frames are returned).

Frames in a sequence may be transmitted during one or more connections (see 6.13) (e.g., for a write command using a single XFER_RDY frame, the COMMAND frame may be transmitted in a connection originated by the SSP initiator port, the XFER_RDY frame in a connection originated by the SSP target port, the DATA frames in one or more connections originated by the SSP initiator port, and the RESPONSE frame in a connection originated by the SSP target port. Alternatively, all the frames may be transmitted in one connection).

8.2.3.2 Task management function sequence of SSP frames

Figure 138 shows the sequence of SSP frames for a task management function (e.g., ABORT TASK (see SAM-5)), including the transport protocol services (see 9.2.1) invoked by the SCSI application layer.

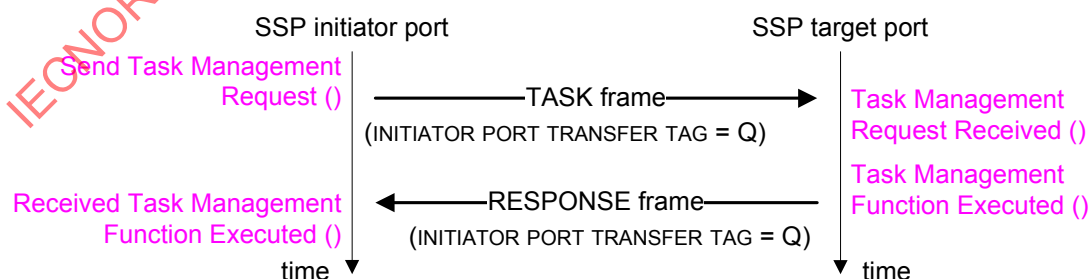


Figure 138 — Task management function sequence of SSP frames

8.2.3.3 Non-data command sequence of SSP frames

Figure 139 shows the sequence of SSP frames for a non-data command (e.g., TEST UNIT READY (see SPC-4)), including the transport protocol services (see 9.2.1) invoked by the SCSI application layer.

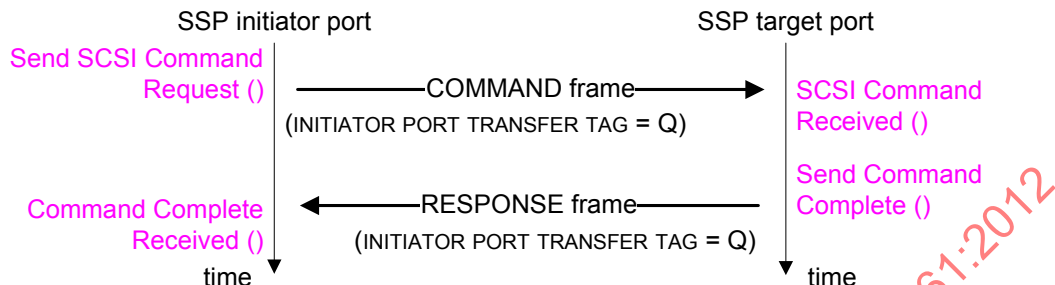


Figure 139 — Non-data command sequence of SSP frames

8.2.3.4 Write command sequence of SSP frames

Figure 140 shows the sequence of SSP frames for a write command (e.g., MODE SELECT (see SPC-4)), including the transport protocol services (see 9.2.1) invoked by the SCSI application layer.

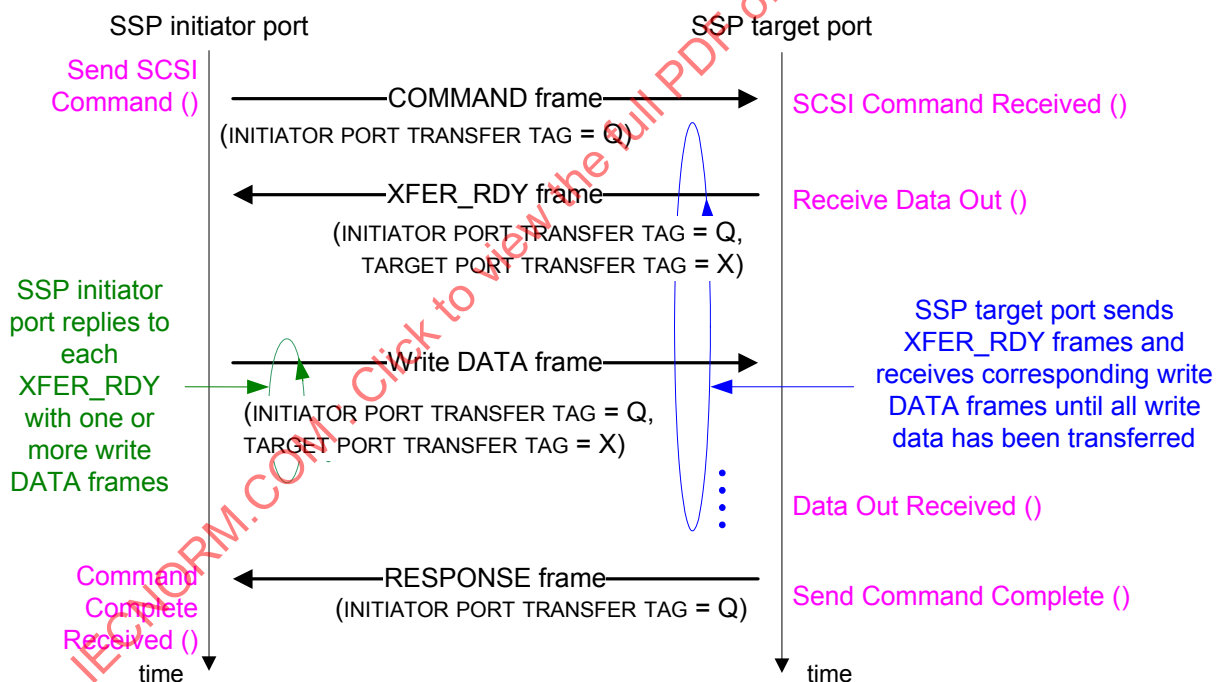


Figure 140 — Write command sequence of SSP frames

8.2.3.5 Read command sequence of SSP frames

Figure 141 shows the sequence of SSP frames for a read command (e.g., INQUIRY, REPORT LUNS, or MODE SENSE (see SPC-4)), including the transport protocol services (see 9.2.1) invoked by the SCSI application layer.

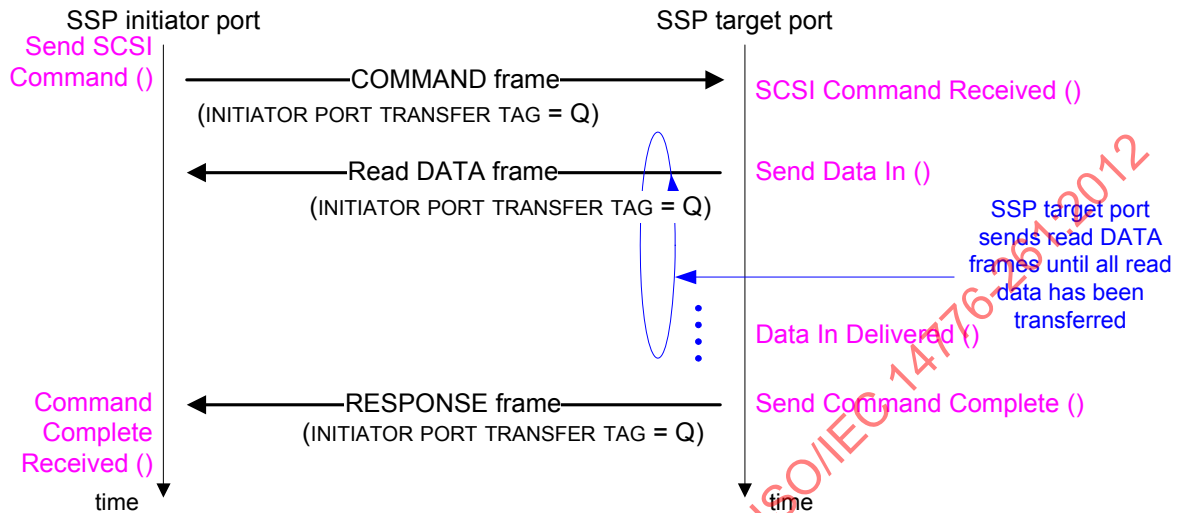


Figure 141 — Read command sequence of SSP frames

8.2.3.6 Bidirectional command sequence of SSP frames

Figure 142 shows the sequence of SSP frames for a bidirectional command (e.g., XDWRITEREAD (see SBC-3)), including the transport protocol services (see 9.2.1) invoked by the SCSI application layer.

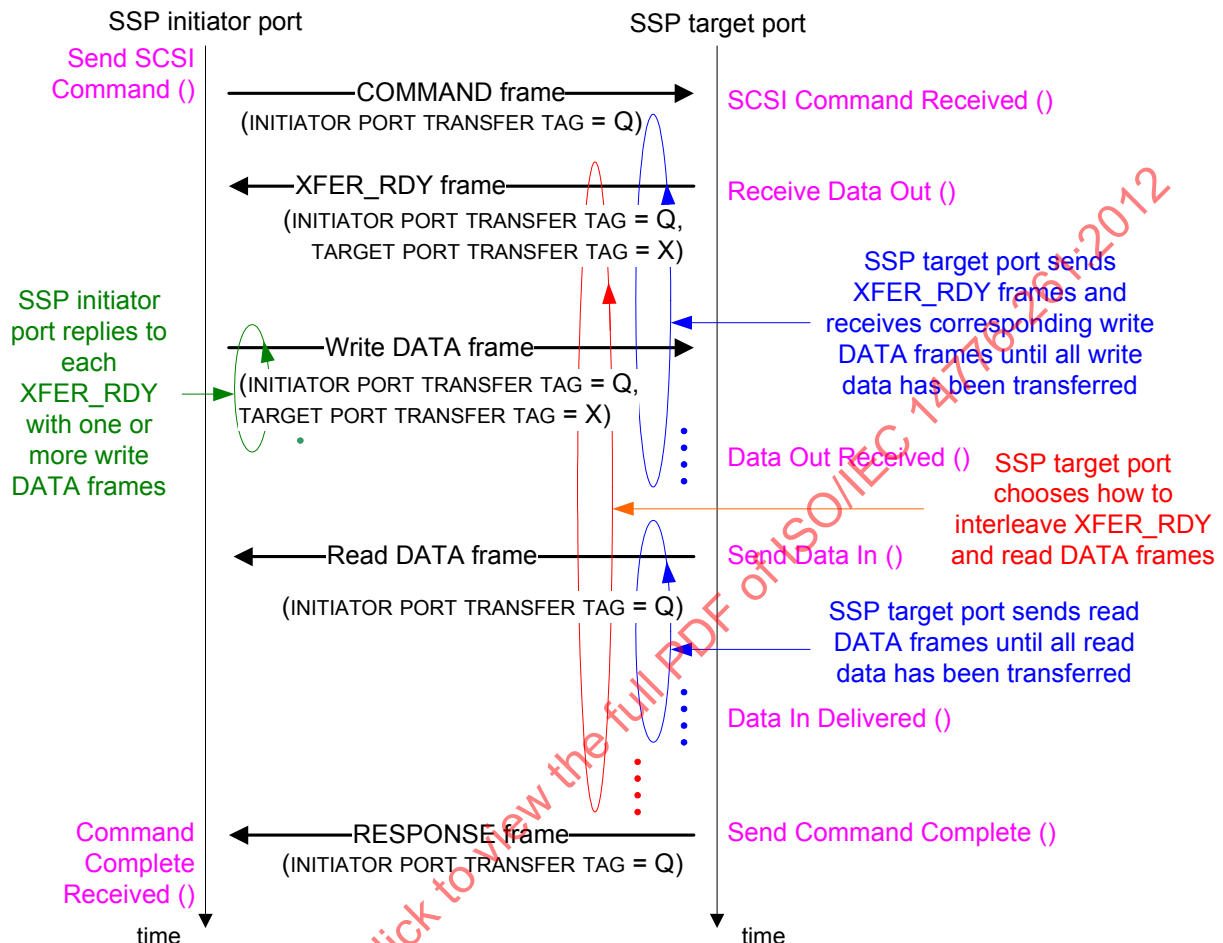


Figure 142 — Bidirectional command sequence of SSP frames

An SSP target port may transmit read DATA frames for a bidirectional command at the same time it is receiving write DATA frames for the same bidirectional command.

8.2.4 SSP transport layer handling of link layer errors

8.2.4.1 SSP transport layer handling of link layer errors overview

The transport layer, sometimes assisted by the SCSI application layer, handles some link layer errors (e.g., NAKs and ACK/NAK timeouts). See 8.2.5 for transport layer handling of transport layer errors (e.g., invalid frame contents).

Link layer errors that occur when transmitting XFER_RDY and DATA frames are handled differently based on the TLR CONTROL field in the COMMAND frame header (see 8.2.1) and the TRANSPORT LAYER RETRIES bit in the Protocol-Specific Logical Unit mode page (see 9.2.7.3) of the logical unit that is the source of the frame.

If transport layer retries are disabled, then the logical unit:

- sets the RETRY DATA FRAMES bit to zero in each XFER_RDY frame;
- may or may not select a different value for the TARGET PORT TRANSFER TAG field in each XFER_RDY frame than that used in the previous XFER_RDY frame for that I_T_L_Q nexus;

- c) processes XFER_RDY frame link layer errors as described in 8.2.4.4.3;
- d) processes read DATA frame link layer errors as described in 8.2.4.5.3; and
- e) processes write DATA frame link layer errors as described in 8.2.4.6.3.

If transport layer retries are enabled, then the logical unit:

- a) sets the RETRY DATA FRAMES bit to one in each XFER_RDY frame;
- b) selects a different value for the TARGET PORT TRANSFER TAG field in each XFER_RDY frame than that used in the previous XFER_RDY frame for that I_T_L_Q nexus;
- c) processes XFER_RDY frame link layer errors as described in 8.2.4.4.2;
- d) processes read DATA frame link layer errors as described in 8.2.4.5.2; and
- e) processes write DATA frame link layer errors as described in 8.2.4.6.2.

8.2.4.2 COMMAND frame - handling of link layer errors

If an SSP initiator port transmits a COMMAND frame and receives a NAK for that frame, then the COMMAND frame was not received. The SSP initiator port should retransmit, in the same or in a new connection, the COMMAND frame at least one time (see 8.2.6.2.3.3). The SSP initiator port may reuse the initiator port transfer tag from the frame for which the NAK was received.

If an SSP initiator port transmits a COMMAND frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5);
- 2) to determine whether the command was received, the application client calls Send Task Management Function Request () (see 9.2.2) with:
 - A) Nexus set to the I_T_L_Q nexus of the COMMAND frame; and
 - B) Function Identifier set to QUERY TASK;
 and
- 3) the SSP initiator port transmits the TASK frame in a new connection to the SSP target port.

If the command is a write command or a bidirectional command and the SSP initiator port receives an XFER_RDY frame for the I_T_L_Q nexus of the command before the RESPONSE frame for the QUERY TASK, then the COMMAND frame was received and is being processed by the SSP target port, and the XFER_RDY frame is valid.

If the command is a read command or a bidirectional command and the SSP initiator port receives a read DATA frame for the I_T_L_Q nexus of the command before the RESPONSE frame for the QUERY TASK, then the COMMAND frame was received and is being processed by the SSP target port, and the read DATA frame is valid.

If the SSP initiator port receives a RESPONSE frame for the I_T_L_Q nexus of the command before the RESPONSE frame for the QUERY TASK, then the COMMAND frame was received by the SSP target port, the RESPONSE frame is valid, and the command processing is complete. The SSP initiator port may reuse the initiator port transfer tag of the COMMAND frame.

If the SSP initiator port receives a RESPONSE frame for the QUERY TASK with a response code of TASK MANAGEMENT FUNCTION SUCCEEDED, then the COMMAND frame was received by the SSP target port (i.e., an ACK was transmitted by the SSP target port for the COMMAND frame) and the command is being processed.

If the SSP initiator port receives a RESPONSE frame for the QUERY TASK with a response code of TASK MANAGEMENT FUNCTION COMPLETE, then the COMMAND frame is not being processed. If neither an XFER_RDY frame, a read DATA frame, nor a RESPONSE frame has been received for the I_T_L_Q nexus of the command, then the COMMAND frame was not received. The SSP initiator port should retransmit the COMMAND frame at least one time. The SSP initiator port may reuse the initiator port transfer tag of the COMMAND frame.

8.2.4.3 TASK frame - handling of link layer errors

If an SSP initiator port transmits a TASK frame and receives a NAK for that frame, then the TASK frame was not received. The SSP initiator port should retransmit, in the same or in a new connection, the TASK frame at least one time with the RETRANSMIT bit set to one (see 8.2.6.2.2.2). The SSP initiator port may reuse the initiator port transfer tag from the frame for which the NAK was received.

If an SSP initiator port transmits a TASK frame and does not receive an ACK or NAK for that frame (e.g., an ACK/NAK timeout occurs, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5);
- 2) the application client calls Send Task Management Request () using the same initiator port transfer tag (see 9.2.2); and
- 3) the SSP initiator port transmits the TASK frame with the RETRANSMIT bit set to one in a new connection to the SSP target port (see 8.2.6.2.2.2).

If the SSP initiator port receives a RESPONSE frame for the TASK frame that arrives before the ACK or NAK for the TASK frame, then the TASK frame was received by the SSP target port (i.e., an ACK was transmitted by the SSP target port for the TASK frame), the RESPONSE frame is valid, and the task management function is complete (see 8.2.6.2.2.3). The initiator port may reuse the initiator port transfer tag of the TASK frame.

8.2.4.4 XFER_RDY frame - handling of link layer errors

8.2.4.4.1 XFER_RDY frame overview

If transport layer retries are enabled, then the SSP target port processes link layer errors that occur while transmitting XFER_RDY frames as described in 8.2.4.4.2.

If transport layer retries are disabled, then the SSP target port processes link layer errors that occur while transmitting XFER_RDY frames as described in 8.2.4.4.3.

8.2.4.4.2 XFER_RDY frame with transport layer retries enabled

If an SSP target port transmits an XFER_RDY frame and receives a NAK for that frame, then the SSP target port retransmits, in the same or a new connection, the XFER_RDY frame at least one time with:

- a) the TARGET PORT TRANSFER TAG field set to a different value than in the original XFER_RDY frame;
- b) the RETRANSMIT bit set to one; and
- c) the other fields set to the same values as in the original XFER_RDY frame (see 8.2.6.3.3.3).

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5); and
- 2) the SSP target port retransmits, in a new connection, the XFER_RDY frame with:
 - A) the TARGET PORT TRANSFER TAG field set to a different value than in the original XFER_RDY frame;
 - B) the RETRANSMIT bit set to one; and
 - C) the other fields set to the same values as in the original XFER_RDY frame (see 8.2.6.3.3.3).

If an SSP initiator port is processing an XFER_RDY frame for an I_T_L_Q nexus (e.g., transmitting write DATA frames for the nexus), and the SSP initiator port receives a new XFER_RDY frame for that I_T_L_Q nexus with the RETRANSMIT bit set to one in the frame, then the ST_ITS state machine stops processing the original XFER_RDY frame (i.e., stops transmitting write DATA frames) and starts servicing the new XFER_RDY frame (see 8.2.6.2.3). The ST_ITS state machine does not transmit any write DATA frames for the original XFER_RDY frame after transmitting a write DATA frame for the new XFER_RDY frame.

An SSP target port may reuse the value in the TARGET PORT TRANSFER TAG field from an XFER_RDY frame for an I_T_L_Q nexus after the SSP target port receives a write DATA frame for a subsequent XFER_RDY frame for that I_T_L_Q nexus.

An SSP target port retransmits each XFER_RDY frame that does not receive an ACK or NAK at least one time.

The number of times an SSP target port retransmits each XFER_RDY frame is vendor specific. When the SSP target port reaches its vendor specific limit, the SSP target port follows the procedure for transport layer retries disabled described in 8.2.4.4.3.

8.2.4.4.3 XFER_RDY frame with transport layer retries disabled

If an SSP target port transmits an XFER_RDY frame and receives a NAK for that frame, then:

- 1) the device server calls Send Command Complete () to return CHECK CONDITION status for that command with the sense key set to ABORTED COMMAND and the additional sense code set to NAK RECEIVED (see 9.2.3); and
- 2) the SSP target port transmits the RESPONSE frame in the same or a new connection.

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5);
- 2) the device server calls Send Command Complete () to return CHECK CONDITION status for that command with the sense key set to ABORTED COMMAND and the additional sense code set to ACK/NAK TIMEOUT or CONNECTION LOST (see 9.2.3); and
- 3) the SSP target port transmits the RESPONSE frame in a new connection.

8.2.4.5 Read DATA frame - handling of link layer errors

8.2.4.5.1 Read DATA frame overview

If an SSP target port transmits a read DATA frame for a command with transport layer retries enabled, then the SSP target port processes link layer errors that occur while transmitting read DATA frames as described in 8.2.4.5.2.

If an SSP target port transmits a read DATA frame for a command with transport layer retries disabled, then the SSP target port processes link layer errors that occur while transmitting read DATA frames as described in 8.2.4.5.3.

8.2.4.5.2 Read DATA frame with transport layer retries enabled

If an SSP target port transmits a read DATA frame and receives a NAK for that frame, then the read DATA frame was not received. The SSP target port retransmits, in the same or in a new connection, all the read DATA frames for that I_T_L_Q nexus since a previous time when ACK/NAK balance occurred at least one time (see 8.2.6.3.3.4).

If an SSP target port transmits a read DATA frame and does not receive an ACK or NAK for that frame (e.g., an ACK/NAK timeout occurs, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5); and
- 2) the ST_TTS state machine retransmits, in a new connection, all the read DATA frames for that I_T_L_Q nexus since a previous time when ACK/NAK balance occurred at least one time (see 8.2.6.3.3.4).

The CHANGING DATA POINTER bit is set to one in the first retransmitted read DATA frame and the CHANGING DATA POINTER bit is set to zero in subsequent read DATA frames.

The ST_TTS state machine retransmits each read DATA frame that does not receive an ACK at least one time (see 8.2.6.3.3).

The number of times an SSP target port retransmits each read DATA frame is vendor specific. When an SSP target port reaches its vendor specific limit for retransmitting read DATA frames, the SSP target port follows the procedure for transport layer retries disabled described in 8.2.4.5.3.

8.2.4.5.3 Read DATA frame with transport layer retries disabled

If an SSP target port transmits a read DATA frame and receives a NAK for that frame, then:

- 1) the device server calls Send Command Complete () to return CHECK CONDITION status for that command with the sense key set to ABORTED COMMAND and the additional sense code set to NAK RECEIVED (see 9.2.3); and
- 2) the SSP target port transmits the RESPONSE frame in the same or a new connection.

If an SSP target port transmits a read DATA frame and does not receive an ACK or NAK for that frame (e.g., an ACK/NAK timeout occurs), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5);
- 2) the device server calls Send Command Complete () to return CHECK CONDITION status for that command with the sense key set to ABORTED COMMAND and the additional sense code set to ACK/NAK TIMEOUT (see 9.2.3); and
- 3) the SSP target port transmits the RESPONSE frame in a new connection.

8.2.4.6 Write DATA frame - handling of link layer errors

8.2.4.6.1 Write DATA frame overview

An SSP initiator port processes link layer errors that occur while transmitting write DATA frames transmitted in response to an XFER_RDY frame that has its RETRY DATA FRAMES bit set to one as described in 8.2.4.6.2.

An SSP initiator port processes link layer errors that occur while transmitting write DATA frames in response to an XFER_RDY frame that has its RETRY DATA FRAMES bit set to zero as described in 8.2.4.6.3.

8.2.4.6.2 Write DATA frame with transport layer retries enabled

If an SSP initiator port transmits a write DATA frame and receives a NAK for that frame, then the write DATA frame was not received. The SSP_ITS state machine retransmits, in the same or in a new connection, all the write DATA frames for the previous XFER_RDY frame (see 8.2.6.2.3.3.3).

If an SSP initiator port transmits a write DATA frame and does not receive an ACK or NAK for that frame (e.g., an ACK/NAK timeout occurs, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5); and
- 2) the ST_ITS state machine retransmits, in a new connection, all the write DATA frames for the previous XFER_RDY frame (see 8.2.6.2.3.3.3).

If that SSP initiator port receives a new XFER_RDY frame or a RESPONSE frame for the command while retransmitting or preparing to retransmit the write DATA frames, then the ST_IFR state machine and ST_ITS state machine process the XFER_RDY frame or RESPONSE frame and stop retransmitting the write DATA frames (see 8.2.6.2.2 and 8.2.6.2.3). The ST_ITS state machine does not transmit a write DATA frame for the previous XFER_RDY frame after transmitting a write DATA frame in response to the new XFER_RDY frame.

The CHANGING DATA POINTER bit is set to one in the first retransmitted write DATA frame and the CHANGING DATA POINTER bit is set to zero in subsequent write DATA frames.

The ST_ITS state machine retransmits each write DATA frame that does not receive an ACK at least one time (see 8.2.6.2.3).

The number of times an SSP initiator port retransmits each write DATA frame is vendor specific. When an SSP initiator port reaches its vendor specific limit for retransmitting write DATA frames, the SSP initiator port follows the procedure for transport layer retries disabled described in 8.2.4.6.3.

8.2.4.6.3 Write DATA frame with transport layer retries disabled

If an SSP initiator port transmits a write DATA frame and does not receive an ACK or NAK for that frame (e.g., an ACK/NAK timeout occurs, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5); and
- 2) the application client aborts the command (see 9.2.2).

If an SSP initiator port transmits a write DATA frame and receives a NAK for that frame, then the application client aborts the command (see 9.2.2).

8.2.4.7 RESPONSE frame - handling of link layer errors

If an SSP target port transmits a RESPONSE frame and receives a NAK for that frame, then the SSP target port retransmits, in the same or a new connection, the RESPONSE frame at least one time with the RETRANSMIT bit set to one and with the other fields set to the same values as in the original RESPONSE frame (see 8.2.6.3.3.3).

If an SSP target port transmits a RESPONSE frame and does not receive an ACK or NAK for that frame (e.g., an ACK/NAK timeout occurs, or the connection is broken), then:

- 1) the SSP_TF state machine closes the connection with DONE (ACK/NAK TIMEOUT) (see 6.17.8.6.5); and
- 2) the SSP target port retransmits, in a new connection, the RESPONSE frame with:
 - A) the RETRANSMIT bit set to one; and
 - B) the other fields set to the same values as in the original RESPONSE frame (see 8.2.6.3.3.3).

The ST_TTS state machine retransmits each RESPONSE frame that does not receive an ACK at least one time (see 8.2.6.3.3). The number of times an SSP target port retransmits each RESPONSE frame is vendor specific.

If an SSP initiator port receives a RESPONSE frame with a RETRANSMIT bit set to one, and the SSP initiator port has previously received a RESPONSE frame for the same I_T_L_Q nexus, then the ST_IFR state machine discards the extra RESPONSE frame (see 8.2.6.2.2.3). If the ST_IFR state machine has not previously received a RESPONSE frame for the I_T_L_Q nexus, then it processes the RESPONSE frame.

8.2.5 SSP transport layer error handling summary

8.2.5.1 SSP transport layer error handling summary introduction

This subclause contains a summary of how SSP ports process transport layer errors. This summary does not include every error case. See 8.2.4 for transport layer handling of link layer errors (e.g., using transport layer retries).

8.2.5.2 SSP initiator port transport layer error handling summary

If an SSP initiator port receives a COMMAND or TASK frame or an unsupported frame type, then the ST_IFR state machine discards the frame (see 8.2.6.2.2.3).

If an SSP initiator port receives an XFER_RDY, read DATA, or RESPONSE frame with an unknown INITIATOR PORT TRANSFER TAG field value, then the ST_IFR state machine discards the frame (see 8.2.6.2.2.3). The application client may then abort the command with that initiator port transfer tag.

If an SSP initiator port receives an XFER_RDY frame with a Transfer Ready information unit that is not 12 bytes long, then the ST_IFR state machine discards the frame (see 8.2.6.2.2.3). The application client may then abort the command.

If an SSP initiator port receives an XFER_RDY frame in response to a command that does not specify write data, then the ST_IFR state machine discards the frame (see 8.2.6.2.2.3), and the application client aborts the command (see 9.2.2).

If an SSP initiator port receives an XFER_RDY frame requesting more write data than expected, then the ST_ITS state machine discards the frame (see 8.2.6.2.3.3), and the application client aborts the command (see 9.2.2).

If an SSP initiator port receives an XFER_RDY frame requesting zero bytes, then the ST_ITS state machine discards the frame (see 8.2.6.2.3.3), and the application client aborts the command (see 9.2.2).

If transport layer retries are disabled and an SSP initiator port receives an XFER_RDY frame with a requested offset that was not expected, then the ST_ITS state machine discards the frame (see 8.2.6.2.3.3), and the application client aborts the command (see 9.2.2).

If an SSP initiator port receives a read DATA frame in response to a command with no read data, then the ST_IFR state machine discards the frame (see 8.2.6.2.2.3), and the application client aborts the command (see 9.2.2).

If an SSP initiator port receives a read DATA frame with more read data than expected, then the ST_ITS state machine discards the frame (see 8.2.6.2.3.3), and the application client aborts the command (see 9.2.2). The SSP initiator port may receive a RESPONSE for the command before being able to abort the command.

If an SSP initiator port receives a read DATA frame with zero bytes, then the ST_ITS state machine discards the frame (see 8.2.6.2.3.3), and the application client aborts the command (see 9.2.2). The SSP initiator port may receive a RESPONSE for the command before being able to abort the command.

If transport layer retries are disabled and an SSP initiator port receives a read DATA frame with a data offset that was not expected, then the ST_ITS state machine discards that frame and any subsequent read DATA frames received for that command (see 8.2.6.2.3.7), and the application client aborts the command (see 9.2.2). The SSP initiator port may receive a RESPONSE for the command before being able to abort the command.

If an SSP initiator port receives a RESPONSE frame that is not the correct length, then the ST_IFR state machine considers the command or task management function completed with an error and discards the frame (see 8.2.6.2.2.3).

8.2.5.3 SSP target port transport layer error handling summary

If an SSP target port receives an XFER_RDY or RESPONSE frame or another unsupported frame type, then the ST_TFR state machine discards the frame (see 8.2.6.3.2.2).

If an SSP target port receives a COMMAND frame and:

- a) the frame is too short to contain a LOGICAL UNIT NUMBER field;
- b) the frame is too short to contain a CDB;
- c) the ADDITIONAL CDB LENGTH field specifies that the frame should be a different length; or
- d) the TLR CONTROL field is set to a non-zero value and non-zero values are not supported,

then the ST_TTS state machine returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID FRAME (see 8.2.6.3.2.2).

If an SSP target port receives a TASK frame that is too short, then the ST_TTS state machine returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID FRAME (see 8.2.6.3.2.2).

If an SSP target port receives a COMMAND frame with an initiator port transfer tag that is already in use for another command, then the device server returns CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to OVERLAPPED COMMANDS ATTEMPTED (see 9.2.4).

If an SSP target port receives:

- a) a COMMAND frame with an initiator port transfer tag that is already in use for a task management function; or
- b) a TASK frame with an initiator port transfer tag that is already in use for a command or another task management function,

then the task router and task manager(s) return a RESPONSE frame with the RESPONSE CODE field set to OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED (see 9.2.4).

If an SSP target port receives a write DATA frame with an unknown initiator port transfer tag, then the ST_TFR state machine discards the frame (see 8.2.6.3.2).

If an SSP target port receives a write DATA frame that does not contain first burst data and for which there is no XFER_RDY frame outstanding (i.e., it has received all requested write data), then the ST_TFR state machine discards the frame (see 8.2.6.3.2.2).

If an SSP target port receives a TASK frame with an unknown logical unit number, then the ST_TFR state machine returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER (see 8.2.6.3.2.2).

If an SSP target port receives a COMMAND frame or TASK frame with a TARGET PORT TRANSFER TAG field set to a value other than FFFFh, then the ST_TFR state machine may return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID FRAME (see 8.2.6.3.2.2).

If an SSP target port is using target port transfer tags and receives a write DATA frame with an unknown target port transfer tag, then the ST_TFR state machine discards the frame (see 8.2.6.3.3).

If transport layer retries are disabled and an SSP target port receives a write DATA frame with a data offset that was not expected, then the ST_TTS state machine discards the frame (see 8.2.6.3.3.6.1), and the device server terminates the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to DATA OFFSET ERROR (see 9.2.3).

If an SSP target port receives a write DATA frame with more write data than expected (i.e., the write DATA frame contains data in excess of that requested by an XFER_RDY frame or, for first burst data, indicated by the FIRST BURST LENGTH field in the Disconnect-Reconnect mode page), then the ST_TTS state machine discards the frame (see 8.2.6.3.3.6.1), and the device server terminates the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to TOO MUCH WRITE DATA (see 9.2.3).

If an SSP target port receives a write DATA frame with zero bytes, then the ST_TTS state machine discards the frame (see 8.2.6.3.3.6.1), and the device server terminates the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to INFORMATION UNIT TOO SHORT (see 9.2.3).

8.2.6 ST (transport layer for SSP ports) state machines

8.2.6.1 ST state machines overview

The ST state machines perform the following functions:

- a) receive and process transport protocol service requests and transport protocol service responses from the SCSI application layer;
- b) receive and process other SAS connection management requests from the SCSI application layer;
- c) send transport protocol service indications and transport protocol service confirmations to the SCSI application layer;
- d) send requests to the port layer to transmit frames and manage SAS connections; and
- e) receive confirmations from the port layer.

The following confirmations between the ST state machines and the port layer:

- a) Transmission Status; and
- b) Frame Received;

include the following as arguments:

- a) initiator port transfer tag;
- b) Destination SAS Address; and
- c) Source SAS Address;

and are used to route the confirmations to the correct ST state machines.

NOTE 71 Although allowed by this standard, the ST state machines do not handle bidirectional commands that result in concurrent write DATA frames and read DATA frames.

8.2.6.2 ST_I (transport layer for SSP initiator ports) state machines

8.2.6.2.1 ST_I state machines overview

The ST_I state machines are as follows:

- a) ST_IFR (initiator frame router) state machine (see 8.2.6.2.2); and
- b) ST_ITS (initiator transport server) state machine (see 8.2.6.2.3).

Each SAS initiator port includes:

- a) one ST_IFR state machine; and
- b) one ST_ITS state machine for each possible command and task management function (i.e., for each initiator port transfer tag).

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Figure 143 shows the ST_I state machines.

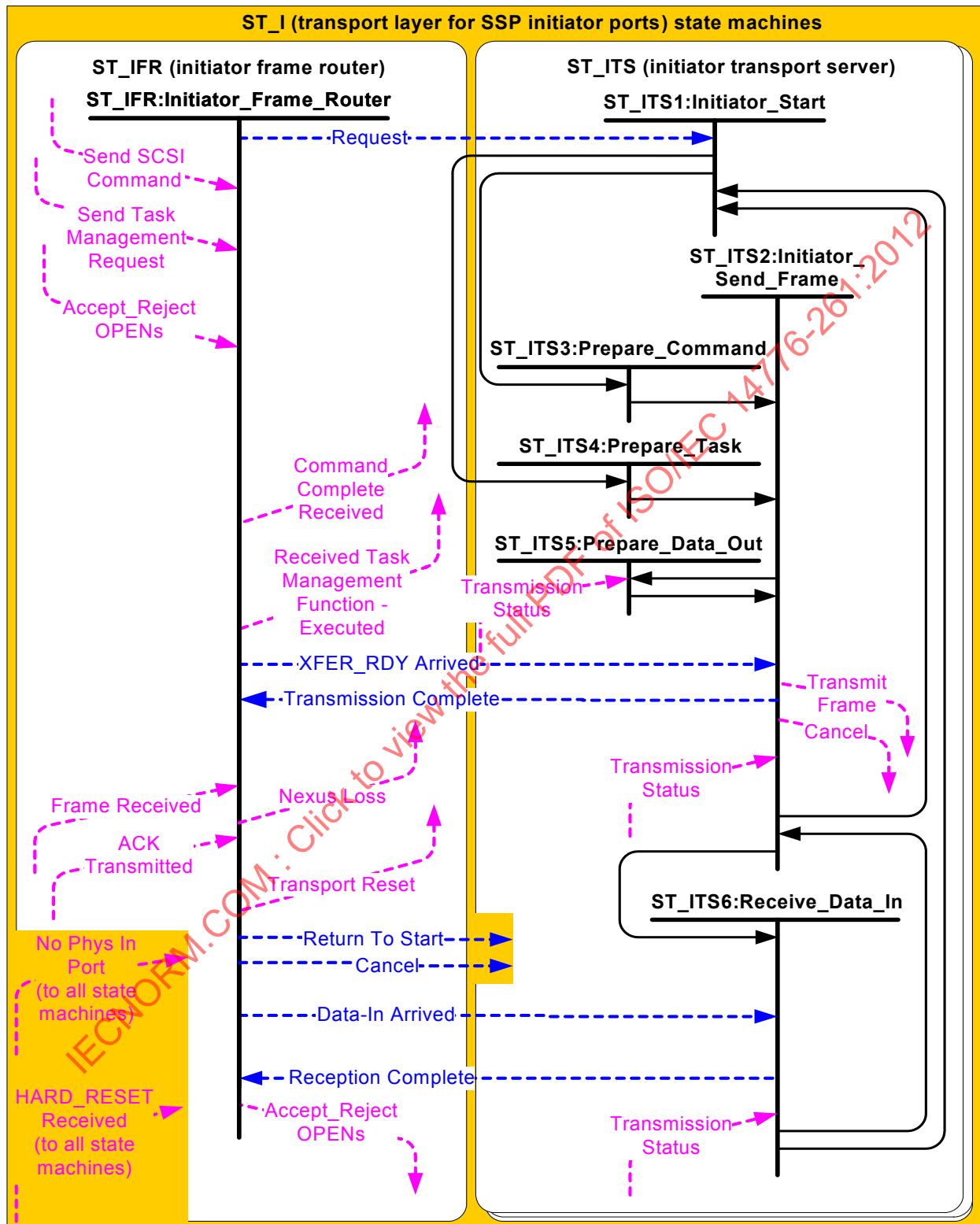


Figure 143 — ST_I (transport layer for SSP initiator ports) state machines

8.2.6.2.2 ST_IFR (initiator frame router) state machine

8.2.6.2.2.1 ST_IFR state machine overview

The ST_IFR state machine performs the following functions:

- a) receives Send SCSI Command and Send Task Management transport protocol service requests from the SCSI application layer;
- b) sends messages to the ST_ITS state machine;
- c) receives messages from the ST_ITS state machine;
- d) receives confirmations from the port layer;
- e) sends transport protocol service confirmations to the SCSI application layer;
- f) receives vendor specific requests from the SCSI application layer;
- g) sends vendor specific confirmations to the SCSI application layer;
- h) receives Accept_Reject OPENs requests from the SCSI application layer;
- i) sends Accept_Reject OPENs requests to the port layer;
- j) sends I_T Nexus Loss event notifications to the SCSI application layer; and
- k) sends Transport Reset event notifications to the SCSI application layer.

This state machine consists of one state.

This state machine shall be started after power on.

8.2.6.2.2.2 Processing transport protocol service requests

If this state machine receives a Send SCSI Command transport protocol service request then this state machine shall send a Request (Send Command) message with Command arguments and Application Client Buffer arguments to the ST_ITS state machine for the specified initiator port transfer tag.

The following is the list of Command arguments:

- a) connection rate;
- b) initiator connection tag;
- c) Destination SAS Address;
- d) Source SAS Address set to the SAS address of the SSP initiator port;
- e) initiator port transfer tag;
- f) logical unit number;
- g) command priority;
- h) task attribute;
- i) additional CDB length;
- j) CDB;
- k) additional CDB bytes, if any;
- l) first burst enabled; and
- m) request fence.

The following is the list of Application Client Buffer arguments:

- a) data-in buffer size;
- b) data-out buffer; and
- c) data-out buffer size.

If the command specifies a write operation and the Send SCSI Command transport service request contains a First Burst Enabled argument, then the Request (Send Command) message shall also include the Enable First Burst argument and the number of bytes for the First Burst Size argument.

If this state machine receives a Send Task Management Request transport protocol service request, then this state machine shall send a Request (Send Task) message with the Task arguments to the ST_ITS state machine for the specified initiator port transfer tag.

The following is the list of Task arguments:

- a) connection rate;
- b) initiator connection tag;

- c) Source SAS Address set to the SAS address of the SSP initiator port;
- d) Destination SAS Address;
- e) retransmit bit;
- f) initiator port transfer tag;
- g) logical unit number;
- h) task management function;
- i) initiator port transfer tag to manage; and
- j) request fence.

If the ST_ITS state machine for the initiator port transfer tag specified in the Send Task Management Request is currently in use, then this state machine shall set the retransmit bit argument to one. If the ST_ITS state machine for the initiator port transfer tag specified in the Send Task Management Request is not currently in use, then this state machine shall set the retransmit bit argument to zero.

8.2.6.2.2.3 Processing Frame Received confirmations

If this state machine receives a Frame Received (ACK/NAK Balanced) confirmation or Frame Received (ACK/NAK Not Balanced) confirmation, then this state machine shall compare the frame type of the frame received with the received confirmation (see table 118 in 8.2.1). If the confirmation was Frame Received (ACK/NAK Balanced) and the frame type is not XFER_RDY, RESPONSE, or DATA, then this state machine shall discard the frame. If the confirmation was Frame Received (ACK/NAK Not Balanced) and the frame type is not DATA, then this state machine shall discard the frame.

If the frame type is correct relative to the Frame Received confirmation, then this state machine may check that the hashed source SAS address matches the SAS address of the SAS port that transmitted the frame and that the hashed destination SAS address matches the SAS address of the SAS port that received the frame based on the connection information. If this state machine checks these SAS addresses, and they do not match, then this state machine:

- a) shall discard the frame; and
- b) may send a vendor specific confirmation to the SCSI application layer to cause the command using that initiator port transfer tag to be aborted.

If the frame type is XFER_RDY, then this state machine shall check the length of the information unit. If the length of the information unit is not correct, then this state machine:

- a) shall discard the frame; and
- b) may send a vendor specific confirmation to the SCSI application layer to cause the command using that initiator port transfer tag to be aborted.

If the frame type is XFER_RDY and the initiator port transfer tag is for a command with no write data, then this state machine shall:

- a) discard the frame;
- b) send a Command Complete Received transport protocol service confirmation with the Delivery Result argument set to Service Delivery or Target Failure - XFER_RDY Not Expected to the SCSI application layer; and
- c) if there is an ST_ITS state machine for the initiator port transfer tag, then send a Return To Start message to that state machine.

If the frame type is DATA and the initiator port transfer tag is for a command with no read data, then this state machine shall:

- a) discard the frame;
- b) send a Command Complete Received transport protocol service confirmation with the Delivery Result argument set to Service Delivery or Target Failure - DATA Not Expected to the SCSI application layer; and
- c) if there is an ST_ITS state machine for the initiator port transfer tag, then send a Return To Start message to that state machine.

If the frame type is RESPONSE, then this state machine shall check the length of the information unit. If the length of the information unit is not correct and the RESPONSE frame was for a command, then this state shall discard the frame and send a Command Complete Received confirmation to the SCSI application layer

with the Service Response argument set to Service Delivery or Target Failure. If the length of the information unit is not correct and the RESPONSE frame was for a task management function, then this state machine shall discard the frame and send a Received Task Management Function – Executed confirmation to the SCSI application layer with the Service Response argument set to Service Delivery or Target Failure.

If the frame type is correct relative to the Frame Received confirmation, then this state machine shall check the initiator port transfer tag. If the initiator port transfer tag does not specify a valid ST_ITS state machine, then this state machine shall discard the frame and may send a vendor specific confirmation to the SCSI application layer to cause the command using that initiator port transfer tag to be aborted.

If the frame type is RESPONSE and this state machine has previously received a RESPONSE frame for the I_T_L_Q nexus, then this state machine shall discard the frame.

If the frame type is RESPONSE, the fields checked in the frame are correct, and this state machine has not previously received a RESPONSE frame for this I_T_L_Q nexus, then this state machine shall send a Return To Start message to the ST_ITS state machine for the specified initiator port transfer tag and:

- a) if the RESPONSE frame was for a command, then this state machine shall send a Command Complete Received protocol service confirmation to the SCSI application layer with the arguments set as specified in table 156 (see 9.2.1.5); or
- b) if the RESPONSE frame was for a task management request, then this state machine shall send a Received Task Management Function Executed protocol service confirmation to the SCSI application layer with the arguments set as specified in table 156 (see 9.2.1.5).

If the frame type is XFER_RDY and the fields checked in the frame are correct, then this state machine shall wait to receive an ACK Transmitted confirmation.

If this state machine receives an ACK Transmitted confirmation for an XFER_RDY frame, then it shall send an XFER_RDY Arrived message to the ST_ITS state machine specified by the initiator port transfer tag. The message shall include the following Xfer_Rdy arguments:

- a) retry data frames;
- b) retransmit bit;
- c) target port transfer tag;
- d) requested offset; and
- e) write data length.

If the frame type is DATA and the fields checked in the frame are correct, then this state machine shall send a Data-In Arrived message to the ST_ITS state machine specified by the initiator port transfer tag. The message shall include the following Read Data arguments:

- a) changing data pointer;
- b) number of fill bytes;
- c) data offset; and
- d) data.

8.2.6.2.2.4 Processing Transmission Complete and Reception Complete messages

If this state machine receives a Transmission Complete (I_T Nexus Loss) message, then this state machine shall send a Nexus Loss event notification to the SCSI application layer.

Table 131 defines the transport protocol service confirmation and Delivery Result argument generated as a result of receiving a Transmission Complete message or a Reception Complete message indicating that an error occurred during the transmission or reception of a frame.

Table 131 — Confirmations sent to the SCSI application layer if a frame transmission error or reception error occurs

Message received from ST_ITS state machine	Protocol service confirmation and Delivery Result argument sent to the SCSI application layer
Transmission Complete (Command Failed, ACK/NAK Timeout)	Command Complete Received (Service Delivery or Target Failure - ACK/NAK Timeout)
Transmission Complete (Command Failed, NAK Received)	Command Complete Received (Service Delivery or Target Failure - NAK Received)
Transmission Complete (Command Failed, Connection Failed)	Command Complete Received (Service Delivery or Target Failure - Connection Failed)
Transmission Complete (Task Failed, ACK/NAK Timeout)	Received Task Management Function - Executed (Service Delivery or Target Failure - ACK/NAK Timeout)
Transmission Complete (Task Failed, NAK Received)	Received Task Management Function - Executed (Service Delivery or Target Failure - NAK Received)
Transmission Complete (Task Failed, Connection Failed)	Received Task Management Function - Executed (Service Delivery or Target Failure - Connection Failed)
Transmission Complete (XFER_RDY Incorrect Write Data Length)	Command Complete Received (Service Delivery or Target Failure - XFER_RDY Incorrect Write Data Length)
Transmission Complete (XFER_RDY Requested Offset Error)	Command Complete Received (Service Delivery or Target Failure - XFER_RDY Requested Offset Error)
Transmission Complete (Cancel Acknowledged)	Command Complete Received (Service Delivery or Target Failure - Cancel Acknowledged)
Reception Complete (Data Offset Error)	Command Complete Received (Service Delivery or Target Failure - DATA Data Offset Error)
Reception Complete (Too Much Read Data)	Command Complete Received (Service Delivery or Target Failure - DATA Too Much Read Data)
Reception Complete (Incorrect Data Length)	Command Complete Received (Service Delivery or Target Failure - DATA Incorrect Data Length)
Reception Complete (Command Failed, ACK/NAK Timeout)	Command Complete Received (Service Delivery or Target Failure - ACK/NAK Timeout)
Reception Complete (Cancel Acknowledged)	Command Complete Received (Service Delivery or Target Failure - Cancel Acknowledged)

The protocol service confirmation shall include the initiator port transfer tag as an argument.

8.2.6.2.2.5 Processing miscellaneous requests

If this state machine receives an Accept_Reject OPENs (Accept SSP) request or an Accept_Reject OPENs (Reject SSP) request, then this state machine shall send an Accept_Reject OPENs request with the same arguments to the port layer.

If this state machine receives a HARD_RESET Received confirmation, then this state machine shall send a Transport Reset event notification to the SCSI application layer.

If this state machine receives a No Phys In Port confirmation, then this state machine shall send a Command Complete Received (Service Delivery or Target Failure - Connection Failed) or Received Task Management

Function Executed (Service Delivery or Target Failure - Connection Failed) confirmation to the SCSI application layer for each ST_ITS state machine that is not in the ST_ITS1:Initiator_Start state.

This state machine may receive vendor specific requests from the SCSI application layer that cause this state machine to send a Cancel message to an ST_ITS state machine.

8.2.6.2.3 ST_ITS (initiator transport server) state machine

8.2.6.2.3.1 ST_ITS state machine overview

The ST_ITS state machine performs the following functions:

- a) receives and processes messages from the ST_IFR state machine;
- b) sends messages to the ST_IFR state machine;
- c) sends request to the port layer regarding frame transmission;
- d) receives confirmations from the port layer regarding frame transmission; and
- e) receives HARD_RESET Received confirmations and No Phys In Port confirmations from the port layer.

This state machine consists of the following states:

- a) ST_ITS1:Initiator_Start state (see 8.2.6.2.3.2) (initial state);
- b) ST_ITS2:Initiator_Send_Frame state (see 8.2.6.2.3.3);
- c) ST_ITS3:Prepare_Command state (see 8.2.6.2.3.4);
- d) ST_ITS4:Prepare_Task state (see 8.2.6.2.3.5);
- e) ST_ITS5:Prepare_Data_Out state (see 8.2.6.2.3.6); and
- f) ST_ITS6:Receive_Data_In state (see 8.2.6.2.3.7).

This state machine shall start in the ST_ITS1:Initiator_Start state after power on.

If this state machine receives a HARD_RESET Received confirmation or a No Phys In Port confirmation, then this state machine shall transition to the ST_ITS1:Initiator_Start state.

This state machine shall maintain the state machine variables defined in table 132.

Table 132 — ST_ITS state machine variables

State machine variable	Description
Data-In Buffer Offset	Current offset in the application client's data-in buffer (i.e., the application client buffer for read data)
Data-Out Buffer Offset	Current offset in the application client's data-out buffer (i.e., the application client buffer for write data)
Previous Requested Offset	Offset in the application client's data-out buffer (i.e., the application client buffer for write data) from the last XFER_RDY frame received
Previous Write Data Length	Write data length from the last XFER_RDY frame received

This state machine shall maintain the state machine arguments defined in table 133.

Table 133 — ST_ITS state machine arguments

State machine argument	Description
Command	Consists of the Command arguments received in the Request (Send Command) message
Task	Consists of the arguments received in the Request (Send Task) message
Xfer_Rdy	Consists of the arguments received in the XFER_RDY Arrived message
Data-Out Buffer	The location of the application client's data-out buffer (i.e., the application client buffer for write data)
Data-Out Buffer Size	The size in bytes of the application client's data-out buffer (i.e., the application client buffer for write data)
Data-In Buffer Size	The size in bytes of the application client's data-in buffer (i.e., the application client buffer for read data)

8.2.6.2.3.2 ST_ITS1:Initiator_Start state

8.2.6.2.3.2.1 State description

This state is the initial state of the ST_ITS state machine.

Upon entry into this state, this state shall set the Data-In Buffer Offset state machine variable to zero.

Upon entry into this state, this state shall set the Data-Out Buffer Offset state machine variable to zero.

8.2.6.2.3.2.2 Transition ST_ITS1:Initiator_Start to ST_ITS3:Prepare_Command

This transition shall occur after receiving a Request (Send Command) message.

8.2.6.2.3.2.3 Transition ST_ITS1:Initiator_Start to ST_ITS4:Prepare_Task

This transition shall occur after receiving a Request (Send Task) message.

8.2.6.2.3.3 ST_ITS2:Initiator_Send_Frame state

8.2.6.2.3.3.1 General

If this state is entered from the ST_ITS3:Prepare_Command state for transmission of a COMMAND frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_ITS6:Receive_Data_In state, and the vendor specific number of retries has not been reached for the COMMAND frame requesting a read operation, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_ITS4:Prepare_Task state for transmission of an TASK frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_ITS5:Prepare_Data_Out state for transmission of a write DATA frame, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer after this state has received an XFER_RDY Arrived message.

If this state is entered from the ST_ITS5:Prepare_Data_Out state for transmission of a write DATA frame and first burst is enabled, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer

after this state has received a Transmission Status (Frame Transmitted) confirmation and a Transmission Status (ACK Received) confirmation for the COMMAND frame.

A Transmit Frame request shall include the COMMAND frame from the ST_ITS3:Prepare_Command state or from the ST_ITS6:Receive_Data_In state, the TASK frame from the ST_ITS4:Prepare_Task state, or the write DATA frame from the ST_ITS5:Prepare_Data_Out state and the following arguments to be used for any OPEN address frame:

- a) Initiator Port Bit set to one;
- b) Protocol set to SSP;
- c) Connection Rate;
- d) Initiator Connection Tag;
- e) Destination SAS Address; and
- f) Source SAS Address.

After sending a Transmit Frame request this state shall wait to receive a Transmission Status confirmation.

If the confirmation is Transmission Status (I_T Nexus Loss), then this state shall send a Transmission Complete (I_T Nexus Loss) message to the ST_IFR state machine. This Transmission Complete message shall include the initiator port transfer tag as an argument.

If the confirmation is not Transmission Status (Frame Transmitted) or Transmission Status (I_T Nexus Loss) (see table 114), and the Transmit Frame request was for a COMMAND frame or a DATA frame, then this state shall send a Transmission Complete (Command Failed, Connection Failed) message to the ST_IFR state machine. The message shall include the initiator port transfer tag.

If the confirmation is not Transmission Status (Frame Transmitted) or Transmission Status (I_T Nexus Loss) (see table 114), and the Transmit Frame request was for a TASK frame, then this state shall send a Transmission Complete (Task Failed, Connection Failed) message to the ST_IFR state machine. The message shall include the initiator port transfer tag.

If the confirmation is Transmission Status (Frame Transmitted), and the Transmit Frame request was for a COMMAND frame not requesting a read operation, a COMMAND frame not requesting a write operation, a TASK frame, or a write DATA frame where the number of data bytes that have been transmitted equal the Data-Out Buffer Size state machine argument, then this state shall wait to receive one of the following confirmations:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

If the confirmation is Transmission Status (Frame Transmitted), and the Transmit Frame request was for a COMMAND frame requesting a write operation, or a write DATA frame where the number of data bytes that have been transmitted is less than the Data-Out Buffer Size state machine argument and the write data length from the previous XFER_RDY frame, then this state shall wait to receive one of the following:

- a) a Transmission Status (ACK Received) confirmation;
- b) a Transmission Status (NAK Received) confirmation;
- c) a Transmission Status (ACK/NAK Timeout) confirmation;
- d) a Transmission Status (Connection Lost Without ACK/NAK) confirmation; or
- e) an XFER_RDY Arrived message.

If a XFER_RDY Arrived message is received, then the ST_ITS shall respond to the XFER_RDY frame as if a Transmission Status (ACK Received) confirmation was received.

NOTE 72 If the number of data bytes requested to be transmitted for the Send SCSI Command protocol service request are fewer than the number of bytes in the service request, then this state may send additional Transmit Frame requests for write DATA frames for the protocol service request before receiving a Transmission Status (ACK Received), Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK) confirmation for Transmit Frame requests for previous write DATA frames sent for the I_T_L_Q nexus.

After a Transmission Status (Frame Transmitted) confirmation is received, if a Transmission Status (NAK Received) confirmation is received, the Transmit Frame request was for a COMMAND frame, and the vendor specific number of retries has not been reached, then this state shall send a Transmit Frame (Interlocked) request to the port layer (i.e., the last COMMAND frame is retransmitted).

After a Transmission Status (Frame Transmitted) confirmation is received, if a Transmission Status (NAK Received) confirmation is received, the Transmit Frame request was for a TASK frame, and the vendor specific number of retries has not been reached, then this state shall send a Transmit Frame (Interlocked) request to the port layer (i.e., the last TASK frame is retransmitted).

Table 134 defines the messages that this state shall send to the ST_IFR state machine upon receipt of the listed confirmations, based on the conditions under which each confirmation was received.

Table 134 — Messages sent to the ST_IFR state machine

Confirmation received from the port layer	Conditions under which confirmation was received	Message sent to ST_IFR state machine
Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK)	The Transmit Frame request was for a COMMAND frame.	Transmission Complete (Command Failed, ACK/NAK Timeout)
	The Transmit Frame request was for a TASK frame.	Transmission Complete (Task Failed, ACK/NAK Timeout)
Transmission Status (NAK Received)	The Transmit Frame request was for a COMMAND frame and the vendor specific number of retries has been reached.	Transmission Complete (Command Failed, NAK Received)
	The Transmit Frame request was for a TASK frame and the vendor specific number of retries has been reached.	Transmission Complete (Task Failed, NAK Received)
Transmission Status (NAK Received)	The Transmit Frame request was for a write DATA frame and: a) the RETRY DATA FRAMES bit was set to zero in the XFER_RDY frame requesting the data; or b) the RETRY DATA FRAMES bit was set to one in the XFER_RDY frame requesting the data, and the vendor specific number of retries has been reached.	Transmission Complete (Data-Out Failed, NAK Received)
Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK)		Transmission Complete (Data-Out Failed, ACK/NAK Timeout)

After this state sends a Transmission Complete (Command Failed, ACK/NAK Timeout) message this state shall continue processing messages and confirmations.

NOTE 73 The application client may determine the command was received and is being processed by the device server and allow the command to complete. The application client may accomplish this by the use of the QUERY TASK task management request.

If this state receives a Return to Start message or a Return to Start argument, and this state has not received confirmations for all Transmit Frame requests sent to the port layer, then this state shall send a Cancel request to the port layer. This state may also send a Cancel request to the port layer to cancel a previous Transmit Frame request.

If this state receives a Cancel message or a Cancel argument, and this state has received confirmations for all Transmit Frame requests sent to the port layer, then this state shall send a Transmission Complete (Cancel Acknowledged) message to the ST_IFR state machine.

If this state receives a Cancel message or a Cancel argument, and this state has not received confirmations for all Transmit Frame requests sent to the port layer, then this state shall send a Cancel request to the port layer. This state may also send a Cancel request to the port layer to cancel a previous Transmit Frame request. The Cancel request shall include the following arguments:

- a) Destination SAS Address; and
- b) Initiator Port Transfer Tag.

NOTE 74 The Cancel message results from a vendor specific request from the SCSI application layer after the SCSI application layer has used a task management function to determine that the SSP target port did not receive the COMMAND frame.

If this state receives a Transmission Status (Cancel Acknowledged) confirmation, then this state shall send a Transmission Complete (Cancel Acknowledged) message to the ST_IFR state machine.

If this state receives an XFER_RDY Arrived message, then this state shall verify the Xfer_Rdy state machine argument as specified in table 135. If the verification fails, then this state shall send the Transmission Complete message specified in table 135 to the ST_IFR state machine.

Table 135 — Transmission Complete messages for XFER_RDY frame verification failures

Message sent to ST_IFR ^a	Condition
Transmission Complete (XFER_RDY Incorrect Write Data Length)	The Write Data Length Xfer_Rdy state machine argument is zero.
	The Requested Offset Xfer_Rdy state machine argument plus the Write Data Length Xfer_Rdy state machine argument is greater than the Data-Out Buffer Size state machine argument.
Transmission Complete (XFER_RDY Requested Offset Error)	First burst is disabled, this is the first XFER_RDY frame for a command, and the value in the Requested Offset Xfer_Rdy state machine argument is not set to zero.
	First burst is enabled, this is the first XFER_RDY frame for a command, and the value in the Requested Offset Xfer_Rdy state machine argument is not equal to the value indicated by the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 9.2.7.2.5).
	Transport layer retries are disabled and the Requested Offset Xfer_Rdy state machine argument is not equal to the Previous Requested Offset state machine variable plus the Previous Write Data Length Field state machine variable.
	Transport layer retries are enabled, the Retransmit Bit Xfer_Rdy state machine argument is set to zero, and the Requested Offset Xfer_Rdy state machine argument is not equal to the Previous Requested Offset state machine variable plus the Previous Write Data Length state machine variable.
	Transport layer retries are enabled, this is not the first XFER_RDY frame for the command, the Retransmit Bit Xfer_Rdy state machine argument is set to one, and the Requested Offset Xfer_Rdy state machine argument is not equal to the Previous Requested Offset state machine variable.
^a If more than one condition is true, then this state shall send the Transmission Complete (XFER_RDY Incorrect Write Data Length) message to the ST_IFR state machine.	

After this state verifies an XFER_RDY frame, it shall:

- a) set the Data-Out Buffer Offset state machine variable to the Requested Offset Xfer_Rdy state machine argument;

- b) set the Previous Requested Offset state machine variable to the Requested Offset Xfer_Rdy state machine argument; and
- c) set the Previous Write Data Length state machine variable to the Write Data Length Xfer_Rdy state machine argument.

8.2.6.2.3.3.2 Transition ST_ITS2:Initiator_Send_Frame to ST_ITS1:Initiator_Start

This transition shall occur after:

- a) this state has sent one of the following to the ST_IFR state machine:
 - A) a Transmission Complete (Command Failed, NAK Received) message;
 - B) a Transmission Complete (Task Failed, ACK/NAK Timeout) message;
 - C) a Transmission Complete (Task Failed, NAK Received) message;
 - D) a Transmission Complete (Command Failed, ACK/NAK Timeout) message and the command was for a non-data operation;
 - E) a Transmission Complete (Data-Out Failed, NAK Received) message;
 - F) a Transmission Complete (Data-Out Failed, ACK/NAK Timeout) message;
 - G) a Transmission Complete (XFER_RDY Incorrect Write Data Length) message;
 - H) a Transmission Complete (XFER_RDY Requested Offset Error) message;
 - I) a Transmission Complete (Cancel Acknowledged) message;
 - J) a Transmission Complete (Command Failed, Connection Failed) message; or
 - K) a Transmission Complete (Task Failed, Connection Failed) message;
- or
- b) this state has received a Return To Start message or Return To Start argument, and has received:
 - A) confirmations for all Transmit Frame requests sent to the port layer; or
 - B) a Transmission Status (Cancel Acknowledged) confirmation.

8.2.6.2.3.3.3 Transition ST_ITS2:Initiator_Send_Frame to ST_ITS5:Prepare_Data_Out

If first burst is enabled, then this transition shall occur and include the First Burst argument after receiving:

- a) a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) confirmation for a COMMAND frame requesting a write operation; or
- b) a Transmission Status (Frame Transmitted) confirmation for a Transmit Frame (Non-interlocked) request if the Data-Out Buffer Offset state machine variable is less than the first burst size.

This transition shall occur after receiving:

- a) an XFER_RDY Arrived message; or
- b) a Transmission Status (Frame Transmitted) confirmation for a Transmit Frame (Non-interlocked) request if the Data-Out Buffer Offset state machine variable is less than the Requested Offset Xfer_Rdy state machine argument plus the Write Data Length Xfer_Rdy state machine argument.

NOTE 75 This transition occurs even if this state has not received a Transmission Status (ACK Received) confirmation for the write DATA frame.

This transition shall include a Retry argument and occur after:

- a) this state receives one of the following confirmations or arguments for a write DATA frame:
 - A) Transmission Status (NAK Received);
 - B) Transmission Status (ACK/NAK Timeout); or
 - C) Transmission Status (Connection Lost without ACK/NAK);
- b) the RETRY DATA FRAMES bit is set to one in the XFER_RDY frame for the write operation;
- c) the Data-Out Buffer Offset state machine variable is set to the Requested Offset Xfer_Rdy state machine argument;
- d) all write DATA frames that have received a Transmission Status (Frame Transmitted) confirmation have received a Transmission Status confirmation; and
- e) the vendor specific number of retries, if any, for the write DATA frame has not been reached.

8.2.6.2.3.3.4 Transition ST_ITS2:Initiator_Send_Frame to ST_ITS6:Process_Data_In

This transition shall occur after receiving a Transmission Status (Frame Transmitted) confirmation for a COMMAND frame for a command requesting a read operation.

NOTE 76 This transition occurs even if this state has not received a Transmission Status (ACK Received) for the COMMAND frame.

8.2.6.2.3.4 ST_ITS3:Prepare_Command state

8.2.6.2.3.4.1 State description

This state shall construct a COMMAND frame using the Command arguments:

- a) FRAME TYPE field set to 06h (i.e., COMMAND frame);
- b) HASHED DESTINATION SAS ADDRESS field set to the hashed value of the Destination SAS Address Commands argument;
- c) HASHED SOURCE SAS ADDRESS field set to the hashed value of the SSP initiator port's SAS address;
- d) RETRY DATA FRAMES bit set to zero;
- e) RETRANSMIT bit set to zero;
- f) CHANGING DATA POINTER bit set to zero;
- g) NUMBER OF FILL BYTES field set to 00b;
- h) INITIATOR PORT TRANSFER TAG field set to the Initiator Port Transfer Tag Command argument;
- i) TARGET PORT TRANSFER TAG field set to FFFFh;
- j) DATA OFFSET field set to 00000000h;
- k) in the information unit, LOGICAL UNIT NUMBER field set to the Logical Unit Number Command argument;
- l) in the information unit, ENABLE FIRST BURST bit set to the Enable First Burst Command argument;
- m) in the information unit, COMMAND PRIORITY field set to the Command Priority Command argument;
- n) in the information unit, TASK ATTRIBUTE field set to the Task Attribute Command argument;
- o) in the information unit, ADDITIONAL CDB LENGTH field set to the Additional CDB Length Command argument;
- p) in the information unit, CDB field set to the CDB Command argument;
- q) in the information unit, ADDITIONAL CDB BYTES field set to the Additional CDB Bytes Command argument, if any; and
- r) no fill bytes.

8.2.6.2.3.4.2 Transition ST_ITS3:Prepare_Command to ST_ITS2:Initiator_Send_Frame

This transition shall occur after this state:

- a) constructs a COMMAND frame;
- b) receives a Cancel message; or
- c) receives a Return To Start message.

This transition shall include:

- a) if neither a Cancel message nor a Return to Start message was received, then the COMMAND frame as an argument;
- b) if a Cancel message was received, then a Cancel argument; or
- c) if a Return To Start message was received, then a Return To Start argument.

8.2.6.2.3.5 ST_ITS4:Prepare_Task state

8.2.6.2.3.5.1 State description

This state shall construct a TASK frame using the Task arguments:

- a) FRAME TYPE field set to 16h (i.e., TASK frame);
- b) HASHED DESTINATION SAS ADDRESS field set to the hashed value of the Destination SAS Address Task argument;

- c) HASHED SOURCE SAS ADDRESS field set to the hashed value of the SSP initiator port's SAS address;
- d) RETRY DATA FRAMES bit set to zero;
- e) RETRANSMIT bit set to the Retransmit Bit Task argument;
- f) CHANGING DATA POINTER bit set to zero;
- g) NUMBER OF FILL BYTES field set to 00b;
- h) INITIATOR PORT TRANSFER TAG field set to the Initiator Port Transfer Tag Task argument;
- i) TARGET PORT TRANSFER TAG field set to FFFFh;
- j) DATA OFFSET field set to 00000000h;
- k) in the information unit, LOGICAL UNIT NUMBER field set to the Logical Unit Number Task argument;
- l) in the information unit, TASK MANAGEMENT FUNCTION field set to the Task Management Function Task argument;
- m) in the information unit, INITIATOR PORT TRANSFER TAG TO MANAGE field set to the Initiator Port Transfer Tag Task argument of the command to be managed; and
- n) no fill bytes.

8.2.6.2.3.5.2 Transition ST_ITS4:Prepare_Task to ST_ITS2:Initiator_Send_Frame

This transition shall occur after this state:

- a) constructs a TASK frame;
- b) receives a Cancel message; or
- c) receives a Return To Start message.

This transition shall include:

- a) if neither a Cancel message nor a Return to Start message was received, then the TASK frame as an argument;
- b) if a Cancel message was received, then a Cancel argument; or
- c) if a Return To Start message was received, then a Return To Start argument.

8.2.6.2.3.6 ST_ITS5:Prepare_Data_Out state

8.2.6.2.3.6.1 State description

This state shall construct a write DATA frame using the following Xfer_Rdy state machine arguments and Command state machine arguments:

- a) FRAME TYPE field set to 01h (i.e., DATA frame);
- b) HASHED DESTINATION SAS ADDRESS field set to the hashed value of the Destination SAS Address Command argument;
- c) HASHED SOURCE SAS ADDRESS field set to the hashed value of the SSP initiator port's SAS address;
- d) RETRY DATA FRAMES bit set to zero;
- e) RETRANSMIT bit set to zero;
- f) CHANGING DATA POINTER bit set as specified in this subclause;
- g) NUMBER OF FILL BYTES field set to the number of fill bytes, based on the length of the specified data;
- h) INITIATOR PORT TRANSFER TAG field set to the Initiator Port Transfer Tag Command argument;
- i) TARGET PORT TRANSFER TAG field set to FFFFh if this state received a First Burst argument or the Target Port Transfer Tag Xfer_Rdy argument if this state did not receive a First Burst argument;
- j) DATA OFFSET field set to the Data-Out Buffer Offset state machine variable;
- k) in the information unit, DATA field set to the information that starts at the location in the Data-Out Buffer state machine argument pointed to by the Data-Out Buffer Offset state machine variable. If the number of bytes remaining to be transferred as defined by the following calculation:

bytes remaining to be transferred = Write Data Length Xfer_Rdy state machine argument - (Data-Out Buffer Offset state machine argument - Requested Offset Xfer_Rdy state machine argument)

is equal to the maximum size of the write Data information unit, then the amount of data shall be the maximum size of the write Data information unit. Otherwise, the amount of data shall be the lesser of:

- A) the bytes remaining to be transferred; and
- B) the maximum size of the Write information unit;

and

- l) fill bytes, if any.

If this state is entered without a Retry argument, then this state shall set the CHANGING DATA POINTER bit to zero.

If this state is entered with a Retry argument, then this state shall set the CHANGING DATA POINTER bit to one.

After constructing the write DATA frame, this state shall set the Data-Out Buffer Offset state machine variable to the value of the DATA OFFSET field plus the number of bytes in the DATA field in the write Data information unit.

8.2.6.2.3.6.2 Transition ST_ITS5:Prepare_Data_Out to ST_ITS2:Initiator_Send_Frame

This transition shall occur after this state:

- a) constructs a write DATA frame;
- b) receives a Cancel message; or
- c) receives a Return To Start message.

This transition shall include the received Transmission Status, if any, as an argument and:

- a) if neither a Cancel message nor a Return to Start message was received, then the write DATA frame as an argument;
- b) if a Cancel message was received, then a Cancel argument; or
- c) if a Return To Start message was received, then a Return To Start argument.

8.2.6.2.3.7 ST_ITS6:Receive_Data_In state

8.2.6.2.3.7.1 State description

If this state receives a Data-In Arrived message, then this state shall verify the values in the read DATA frame received with the message as defined in table 136.

If the verification fails, then this state sends the Reception Complete message specified in table 136 to the ST_IFR state machine.

Table 136 — Reception Complete messages for read DATA frame verification failures

Message sent to ST_IFR ^a	Condition
Reception Complete (Data Offset Error)	Transport layer retries are disabled, and the DATA OFFSET field in the read DATA frame is not equal to the Data-In Buffer Offset state machine variable.
	The DATA OFFSET field in the read DATA frame is greater than the Data-In Buffer Size state machine argument.
Reception Complete (Too Much Read Data)	The number of bytes in the DATA field in the read Data information unit plus the Data-In Buffer Offset state machine variable is greater than the Data-In Buffer Size state machine argument.
Reception Complete (Information Unit Too Short)	Either: a) the number of bytes in the DATA field in the read Data information unit is zero; or b) this is not the last read DATA frame for the command and the NUMBER OF FILL BYTES field is not set to 00b.
^a If more than one condition is true, then this state shall select which message to send to the ST_IFR state machine using the following order: 1) Reception Complete (Data Offset Error); 2) Reception Complete (Too Much Read Data); or 3) Reception Complete (Incorrect Data Length).	

If:

- a) transport layer retries are enabled;
- b) the CHANGING DATA POINTER bit is set to zero;
- c) the DATA OFFSET field is not set to the Data-In Buffer Offset state machine variable;
- d) the DATA OFFSET field is less than the Data-In Buffer Size state machine argument; and
- e) the DATA OFFSET field plus the number of bytes in the DATA field in the read Data information unit is less than or equal to the Data-In Buffer Size state machine argument,

then this state should discard all Data-In Arrived messages until a read DATA frame is received in which the CHANGING DATA POINTER bit is set to one. This state shall resume processing additional Data-In Arrived messages when it receives a Data-In Arrived message with the CHANGING DATA POINTER bit set to one.

If the read DATA frame verification is successful or after this state resumes processing Data-In Arrived messages, then this state shall process the data received in the read DATA frame and set the Data-In Buffer Offset state machine variable to the DATA OFFSET field plus the number of bytes in the DATA field in the read Data information unit.

If data received in the read DATA frame overlaps data previously received and verified as having no errors, then this state may either discard the overlapping data, or replace the previously received data with the new data.

If this state receives a Transmission Status (ACK/NAK Timeout) confirmation or a Transmission Status (Connection Lost Without ACK/NAK) confirmation, then this state shall send a Reception Complete (Command Failed, ACK/NAK Timeout) message to the ST_IFR state machine.

After this state sends a Reception Complete (Command Failed, ACK/NAK Timeout) message, this state shall continue processing messages and confirmations.

NOTE 77 The application client may determine the command was received and is being processed by the device server and allow the command to complete.

If this state receives a Cancel message, then this state shall send a Reception Complete (Cancel Acknowledged) message to the ST_IFR state machine. The Reception Complete message shall include the initiator port transfer tag as an argument.

NOTE 78 The Cancel message results from a vendor specific request from the SCSI application layer after the SCSI application layer has used a task management function to determine that the SSP target port did not receive the COMMAND frame.

8.2.6.2.3.7.2 Transition ST_ITS6:Receive_Data_In to ST_ITS1:Initiator_Start

This transition shall occur after this state:

- a) sends one of the following to the ST_IFR state machine:
 - A) a Reception Complete (Data Offset Error) message;
 - B) a Reception Complete (Too Much Read Data) message;
 - C) a Reception Complete (Incorrect Data Length) message; or
 - D) a Reception Complete (Cancel Acknowledged) message;
 or
- b) receives a Return To Start message.

8.2.6.2.3.7.3 Transition ST_ITS6:Receve_Data_In to ST_ITS2:Initiator_Send_Frame

This transition shall occur after this state receives a Transmission Status (NAK Received) confirmation for a COMMAND frame for a command requesting a read operation.

8.2.6.3 ST_T (transport layer for SSP target ports) state machines

8.2.6.3.1 ST_T state machines overview

The ST_T state machines are as follows:

- a) ST_TFR (target frame router) state machine (see 8.2.6.3.2); and
- b) ST_TTS (target transport server) state machine (see 8.2.6.3.3).

The SAS target port includes:

- a) one ST_TFR state machine; and
- b) one ST_TTS state machine for each possible command and task management function (i.e., for each initiator port transfer tag).

This state machine may maintain the timers listed in table 137.

Table 137 — ST_T state machine timers

Timer	Initial value
Initiator Response Timeout	The value in the INITIATOR RESPONSE TIMEOUT field in the Protocol-Specific Port mode page (see 9.2.7.4).

Figure 144 shows the ST_T state machines.

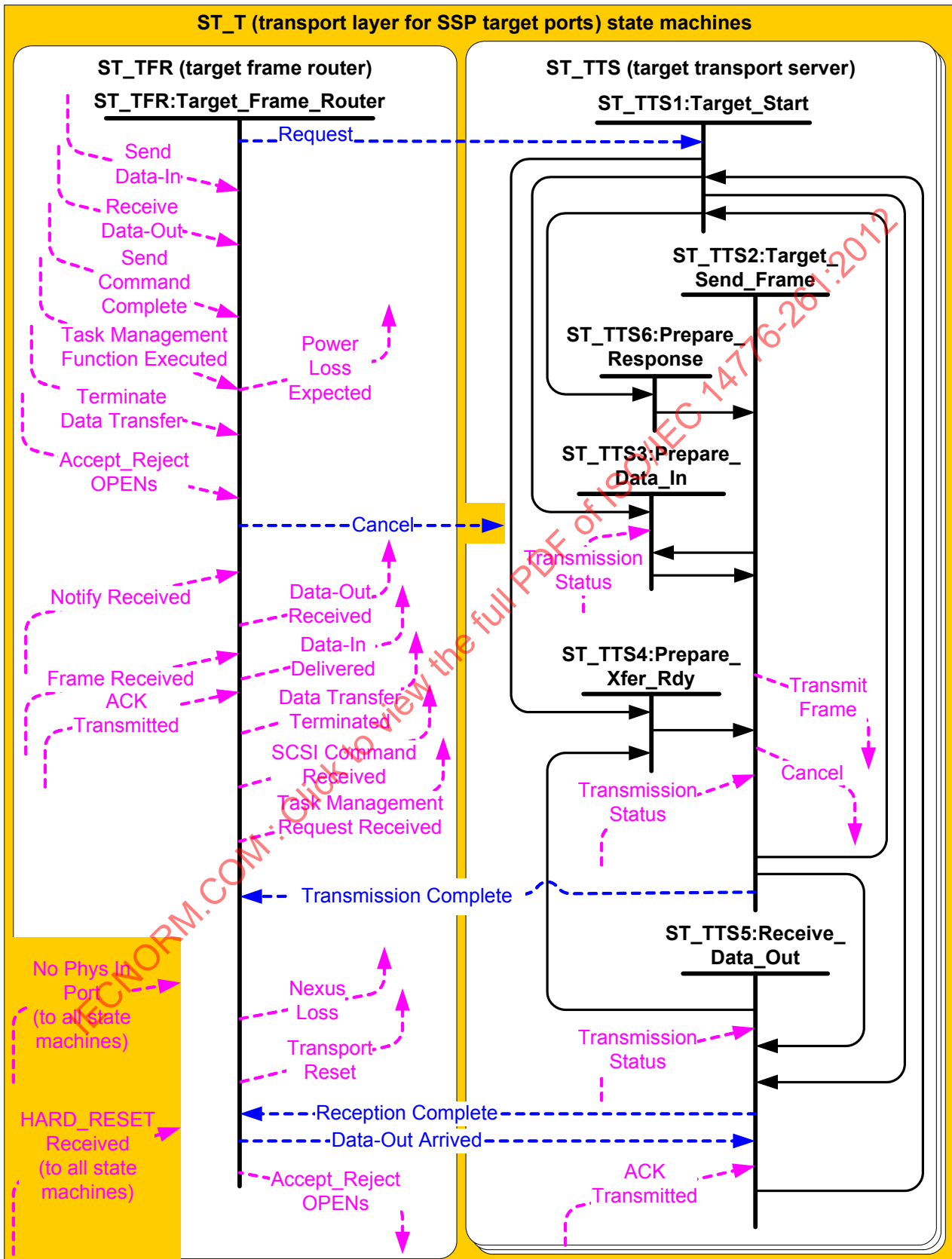


Figure 144 — ST_T (transport layer for SSP target ports) state machines

8.2.6.3.2 ST_TFR (target frame router) state machine

8.2.6.3.2.1 ST_TFR state machine overview

The ST_TFR state machine performs the following functions:

- a) receives confirmations from the port layer;
- b) receives transport protocol service requests from the SCSI application layer;
- c) sends transport protocol service indications to the SCSI application layer;
- d) sends messages to the ST_TTS state machine;
- e) receives messages from the ST_TTS state machine;
- f) receives Accept_Reject OPENs requests from the SCSI application layer;
- g) sends Accept_Reject OPENs requests to the port layer;
- h) sends Nexus Loss event notifications to the SCSI application layer;
- i) sends Transport Reset event notifications to the SCSI application layer; and
- j) sends Power Loss Expected event notifications to the SCSI application layer.

This state machine consists of one state.

This state machine shall be started after power on.

If this state receives a Notify Received (Power Loss Expected) confirmation, then this state shall:

- a) send a Cancel message to all the ST_TTS state machines; and
- b) send a Power Loss Expected confirmation to the SCSI application layer.

8.2.6.3.2.2 Processing Frame Received confirmations

If this state machine receives a Frame Received (ACK/NAK Balanced) or Frame Received (ACK/NAK Not Balanced) confirmation, then this state machine shall check the frame type in the received frame (see table 118 in 8.2.1). If the frame type is not COMMAND, TASK, or DATA, then this state machine shall discard the frame. If the confirmation was Frame Received (ACK/NAK Not Balanced) and the frame type is not DATA, then this state machine shall discard the frame.

This state machine may check that reserved fields in the received frame are zero. If non-zero values are not supported in the TLR CONTROL field in a COMMAND frame, then the TLR CONTROL field shall be treated as a reserved field. If any reserved fields are checked and they are not set to zero, then this state machine shall send the following to an ST_TTS state machine that does not have an active command or task management function and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the Destination SAS Address argument set to the SAS address from which the invalid frame was received; and
- c) the service response argument set to Invalid Frame.

The check of reserved fields within the frame shall not apply to the reserved fields within the CDB in a COMMAND frame. Checking of reserved fields in a CDB is described in SPC-4.

The following is the list of Transport Response arguments:

- a) connection rate;
- b) initiator connection tag;
- c) Destination SAS Address (i.e., the SAS address to which the RESPONSE frame is to be transmitted);
- d) Source SAS Address set to the SAS address of the SAS port containing the state machine;
- e) initiator port transfer tag; and
- f) service response.

The response fence argument is not included in the Transport Response arguments.

If the frame type is correct relative to the Frame Received confirmation, then this state machine may check that the hashed source SAS address matches the SAS address of the SAS port that transmitted the frame and that the hashed destination SAS address matches the SAS address of the SAS port that received the frame based on the connection information. If this state machine checks these SAS addresses, and they do not match, then this state machine shall discard the frame.

If the frame type is COMMAND or TASK then this state machine shall check the length of the information unit. If the length of the information unit is not correct, then this state machine shall send the following to an ST_TTS state machine that does not have an active command or task management function and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the Destination SAS Address argument set to the SAS address from which the invalid frame was received; and
- c) the service response argument set to Invalid Frame.

If the frame type is TASK, this state machine checks initiator port transfer tags, the RETRANSMIT bit in the new TASK frame is set to one, and the initiator port transfer tag for the new TASK frame is the same as the initiator port transfer tag for a previous TASK frame where the task management function for the previous TASK frame is not complete, then this state machine shall discard the new TASK frame and not send a Task Management Request Received confirmation to the SCSI application layer.

If the frame type is TASK and this state machine does not check initiator port transfer tags, then this state machine shall ignore the RETRANSMIT bit.

If the frame type is COMMAND or TASK, then this state machine may check the target port transfer tag. If this state checks the target port transfer tag and it is set to a value other than FFFFh, then this state machine shall send the following to an ST_TTS state machine that does not have an active command or task management function and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the Destination SAS Address argument set to the SAS address from which the invalid frame was received; and
- c) the service response argument set to Invalid Frame.

If the frame type is TASK, then this state machine shall check the logical unit number. If the logical unit number is unknown, then this state machine shall send the following to an ST_TTS state machine that does not have an active command or task management function and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the Destination SAS Address argument set to the SAS address from which the invalid frame was received; and
- c) the service response argument set to Incorrect Logical Unit Number.

If:

- a) the frame type is DATA and this frame is for first burst data; or
- b) this state machine did not assign a target port transfer tag for the data transfer,

then this state machine may check the target port transfer tag. If the target port transfer tag is set to a value other than FFFFh, then this state machine shall send the following to an ST_TTS state machine that does not have an active command or task management function and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the Destination SAS Address argument set to the SAS address from which the invalid frame was received; and
- c) the service response argument set to Invalid Frame.

If the frame type is COMMAND or TASK and the fields checked in the frame are correct, then this state machine shall wait to receive an ACK Transmitted confirmation.

If the frame type is COMMAND, the fields checked in the frame are correct, and this state machine receives an ACK Transmitted confirmation, then this state machine shall send a SCSI Command Received transport protocol service indication with the following arguments to the SCSI application layer:

- a) Source SAS Address (i.e., the SAS address that transmitted the COMMAND frame);
- b) Initiator Port Transfer Tag;
- c) Logical Unit Number;
- d) Task Attribute;
- e) Command Priority;
- f) CDB; and

- g) Additional CDB Bytes, if any.

If the frame type is TASK, the fields checked in the frame are correct, and this state machine receives an ACK Transmitted confirmation, then this state machine shall send a Task Management Request Received transport protocol service indication with the following arguments to the SCSI application layer:

- a) Source SAS Address (i.e., the SAS address that transmitted the TASK frame);
- b) Initiator Port Transfer Tag;
- c) Logical Unit Number;
- d) Task Management Function; and
- e) Initiator Port Transfer Tag To Manage.

If the frame type is DATA, and the initiator port transfer tag does not match an initiator port transfer tag for an outstanding command performing write operations, then this state machine shall discard the frame.

If the frame type is DATA, and the initiator port transfer tag matches an initiator port transfer tag for an outstanding command performing write operations when first burst is disabled or for which no Transmission Complete (Xfer_Rdy Delivered) message has been received from an ST_TTS state machine, then this state machine shall discard the frame.

If the frame type is DATA and a target port transfer tag was received in a Transmission Complete (Xfer_Rdy Delivered) message, then this state machine shall check the target port transfer tag. If the target port transfer tag received in the DATA frame does not match the Target Port Transfer Tag argument in the Transmission Complete (Xfer_Rdy Delivered) message, then this state machine shall discard the frame.

If the frame type is DATA and the fields checked in the frame are correct, and first burst is enabled or this state machine has received a Transmission Complete (Xfer_Rdy Delivered) from the ST_TTS state machine for the request, then this state machine shall send a Data-Out Arrived message to the ST_TTS state machine specified by the initiator port transfer tag in the frame. The message shall include the content of the write DATA frame.

8.2.6.3.2.3 Processing transport protocol service requests and responses

If this state machine receives a Send Data-In transport protocol service request from the SCSI application layer, then this state machine shall send a Request (Send Data-In) message to an ST_TTS state machine that does not have an active command or task management function. The message shall include the following Data-In arguments:

- a) connection rate;
- b) initiator connection tag;
- c) Destination SAS Address (i.e., the SAS address to which the read DATA frame is to be transmitted);
- d) Source SAS Address set to the SAS address of the SSP target port;
- e) initiator port transfer tag;
- f) device server buffer;
- g) request byte count; and
- h) application client buffer offset.

If this state machine receives a Receive Data-Out transport protocol service request from the SCSI application layer, then this state machine shall send a Request (Receive Data-Out) message to an ST_TTS state machine that does not have an active command or task management function. The message shall include the following Data-Out state machine arguments:

- a) connection rate;
- b) initiator connection tag;
- c) Destination SAS Address (i.e., the SAS address to which the XFER_RDY frame is to be transmitted);
- d) Source SAS Address set to the SAS address of the SSP target port;
- e) initiator port transfer tag;
- f) device server buffer;
- g) request byte count;
- h) application client buffer offset; and
- i) target port transfer tag.

If first burst is enabled, then the Request (Receive Data_Out) message shall also include the Enable First Burst argument and First Burst Size argument. The First Burst Size argument shall be set to the first burst size from the Disconnect-Reconnect mode page (see 9.2.7.2.5).

If this state machine receives a Send Command Complete transport protocol service response from the SCSI application layer with the Service Response argument set to TASK COMPLETE, then this state machine shall send a Request (Send Application Response) message to the ST_TTS state machine specified by the initiator port transfer tag. The message shall include the following Application Response arguments:

- a) connection rate;
- b) initiator connection tag;
- c) Destination SAS Address (i.e., the SAS address to which the RESPONSE frame is to be transmitted);
- d) Source SAS Address set to the SAS address of the SSP target port;
- e) initiator port transfer tag;
- f) status;
- g) status qualifier, if any;
- h) sense data, if any; and
- i) response fence.

If this state machine receives a Task Management Function Executed transport protocol service response from the SCSI application layer, then this state machine shall send the following to the ST_TTS state machine specified by the initiator port transfer tag:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the service response argument set as specified in table 138; and
- c) the response fence argument set to the Task Management Function Executed protocol service response Response Fence argument.

Table 138 specifies which argument to send with the Request (Send Transport Response) message based on the Service Response argument that was received.

Table 138 — Task Management Function Executed Service Response argument mapping to Request (Send Transport Response) service response argument

Task Management Function Executed protocol service response Service Response argument received	Request (Send Transport Response) message service response argument
FUNCTION COMPLETE	Task Management Function Complete
FUNCTION SUCCEEDED	Task Management Function Succeeded
FUNCTION REJECTED	Task Management Function Not Supported
INCORRECT LOGICAL UNIT NUMBER	Incorrect Logical Unit Number
SERVICE DELIVERY OR TARGET FAILURE - Overlapped Initiator Port Transfer Tag Attempted	Overlapped Initiator Port Transfer Tag Attempted

If this state machine receives a Terminate Data Transfer protocol service request from the SCSI application layer and this state machine has not sent a Request message to a ST_TTS state machine for the Send Data-In or Receive Data-Out protocol service request to which the Terminate Data Transfer request applies, then this state machine shall:

- 1) discard the Terminate Data Transfer request and any corresponding Send Data-In or Receive Data-Out request; and
- 2) send a Data Transfer Terminated protocol service confirmation to the SCSI application layer.

If this state machine receives a Terminate Data Transfer protocol service request from the SCSI application layer and this state machine has sent a Request message to a ST_TTS state machine for the Send Data-In protocol service request to which the Terminate Data Transfer request applies, then this state machine shall

send a Cancel message to the ST_TTS state machine specified by the initiator port transfer tag and the Send Data-In protocol service request.

If this state machine receives a Terminate Data Transfer protocol service request from the SCSI application layer and this state machine has sent a Request message to a ST_TTS state machine for the Receive Data-Out protocol service request to which the Terminate Data Transfer request applies, then this state machine shall send a Cancel message to the ST_TTS state machine specified by the initiator port transfer tag and the Receive Data-Out protocol service request.

This state machine receives Transmission Complete and Reception Complete messages that may result in this state machine sending a Nexus Loss event notification or a protocol service confirmation to the SCSI application layer. If this state machine receives a Transmission Complete (I_T Nexus Loss) message, then this state machine shall send a Nexus Loss event notification to the SCSI application layer. Table 139 defines messages received from ST_TTS state machines and the corresponding service confirmations, if any, that shall be sent upon receipt of the message.

Table 139 — Confirmations sent to the SCSI application layer (part 1 of 2)

Message received from ST_TTS state machine	Protocol service confirmation sent to SCSI application layer
Transmission Complete (Xfer_Rdy Delivered)	None
Transmission Complete (Response Delivered)	None
Transmission Complete (Response Failed) ^a	None
Transmission Complete (Data Transfer Terminated)	Data Transfer Terminated
Transmission Complete (Data-In Delivered)	Data-In Delivered with the Delivery Result argument set to DELIVERY SUCCESSFUL
Transmission Complete (Xfer_Rdy Failed, NAK Received)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - NAK RECEIVED
Transmission Complete (Xfer_Rdy Failed, Connection Failed)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - CONNECTION FAILED
Transmission Complete (Data-In Failed, NAK Received)	Data-In Delivered with the Delivery Result argument set to DELIVERY FAILURE - NAK RECEIVED
Transmission Complete (Data-In Failed, ACK/NAK Timeout)	Data-In Delivered with the Delivery Result argument set to DELIVERY FAILURE - ACK/NAK TIMEOUT
Transmission Complete (Data-In Failed, Connection Lost without ACK/NAK)	Data-In Delivered with the Delivery Result argument set to DELIVERY FAILURE - CONNECTION FAILED
Reception Complete (Data-Out Received)	Data-Out Received with the Delivery Result argument set to DELIVERY SUCCESSFUL
Reception Complete (Data Offset Error)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - DATA OFFSET ERROR
^a SAM-5 does not define a mechanism for the device server to determine the result of its Send Command Complete and Task Management Function Executed transport protocol service response calls.	

Table 139 — Confirmations sent to the SCSI application layer (part 2 of 2)

Message received from ST_TTS state machine	Protocol service confirmation sent to SCSI application layer
Reception Complete (Too Much Write Data)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - TOO MUCH WRITE DATA
Reception Complete (Information Unit Too Short)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - INFORMATION UNIT TOO SHORT
Reception Complete (Initiator Response Timeout)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT
Reception Complete (Data Transfer Terminated)	Data Transfer Terminated
^a SAM-5 does not define a mechanism for the device server to determine the result of its Send Command Complete and Task Management Function Executed transport protocol service response calls.	

Each protocol service confirmation shall include the initiator port transfer tag as an argument.

8.2.6.3.2.4 Processing miscellaneous requests and confirmations

If this state machine receives an Accept_Reject OPENs (Accept SSP) request or an Accept_Reject OPENs (Reject SSP) request, then this state machine shall send an Accept_Reject OPENs request with the same arguments to the port layer.

If this state machine receives a HARD_RESET Received confirmation, then this state shall send a Transport Reset event notification to the SCSI application layer.

If this state machine receives a No Phys In Port confirmation, then this state shall send a Nexus Loss event notification to the SCSI application layer.

8.2.6.3.3 ST_TTS (target transport server) state machine

8.2.6.3.3.1 ST_TTS state machine overview

The ST_TTS state machine performs the following functions:

- receives and processes messages from the ST_TFR state machine;
- sends messages to the ST_TFR state machine;
- communicates with the port layer using requests and confirmations regarding frame transmission; and
- receives HARD_RESET Received confirmations and No Phys In Port confirmations from the port layer.

This state machine consists of the following states:

- ST_TTS1:Target_Start (see 8.2.6.3.3.2) (initial state);
- ST_TTS2:Target_Send_Frame (see 8.2.6.3.3.3);
- ST_TTS3:Prepare_Data_In (see 8.2.6.3.3.4);
- ST_TTS4:Prepare_Xfer_Rdy (see 8.2.6.3.3.5);
- ST_TTS5:Receive_Data_Out (see 8.2.6.3.3.6); and
- ST_TTS6:Prepare_Response (see 8.2.6.3.3.7).

This state machine shall start in the ST_TTS1:Target_Start state after power on.

If this state machine receives a HARD_RESET Received confirmation or a No Phys In Port confirmation, then this state machine shall transition to the ST_TTS1:Target_Start state.

The state machine shall maintain the state machine variables defined in table 140.

Table 140 — ST_TTS state machine variables

State machine variable	Description
Read Data Offset	Offset into the application client's data-in buffer (i.e., the application client buffer for read data)
Balance Point Read Data Offset	Offset into the application client's data-in buffer (i.e., the application client buffer for read data) of last point at which the number of Transmission Status (ACK Received) confirmations or arguments was equal to the number of transmitted read DATA frames
Read Data Frames Transmitted	The number of Transmission Status (Frame Transmitted) confirmations received for read DATA frames
Read Data Frames ACKed	The number of Transmission Status (ACK Received) confirmations received for read DATA frames
Read Data Buffer End	One greater than the offset into the application client's data-in buffer (i.e., the application client buffer for read data) of the last location into which read data is to be placed.
Requested Write Data Offset	Device server requested offset in the application client buffer for write data
Requested Write Data Length	Amount of write data requested by the device server from the application client buffer
Write Data Offset	Offset into the application client's data-out buffer (i.e., the application client buffer containing write data)

This state machine shall maintain the state machine arguments defined in table 141.

Table 141 — ST_TTS state machine arguments

State machine argument	Description
Data-In	The Data-In arguments received in the Request (Send Data-In) message (see 8.2.6.3.2.3)
Data-Out	The Data-Out arguments received in the Request (Receive Data-Out) message (see 8.2.6.3.2.3)

8.2.6.3.3.2 ST_TTS1:Target_Start state

8.2.6.3.3.2.1 State description

This state is the initial state of the ST_TTS state machine.

Upon entry into this state, this state shall:

- set the Read Data Offset state machine variable to the Application Client Buffer Offset Data-In state machine argument;
- set the Balance Point Read Data Offset state machine variable to the Application Client Buffer Offset Data-In state machine argument;
- set the Read Data Frames Transmitted state machine variable to zero;
- set the Read Data Frames ACKed state machine variable to zero;

- e) set the Read Data Buffer End state machine variable to the Application Client Buffer Offset Data-In state machine argument plus the Request Byte Count Data-In state machine argument; and
- f) set the Requested Write Data Offset state machine variable to the Application Client Buffer Offset Data-Out state machine argument.

If this state was entered without an Enable First Burst Data-Out state machine argument, then the Requested Write Data Length state machine variable shall be set to the Request Byte Count Data-Out state machine argument.

If this state was entered with an Enable First Burst Data-Out state machine argument, then the Requested Write Data Length state machine variable shall be set to the First Burst Size Data-Out state machine argument.

8.2.6.3.3.2.2 Transition ST_TTS1:Target_Start to ST_TTS3:Prepare_Data_In

This transition shall occur after receiving a Request (Send Data-In) message.

8.2.6.3.3.2.3 Transition ST_TTS1:Target_Start to ST_TTS4:Prepare_Xfer_Rdy

If this state was entered without an Enable First Burst Data-Out state machine argument, then this transition shall occur after a Request (Receive Data-Out) message is received.

8.2.6.3.3.2.4 Transition ST_TTS1:Target_Start to ST_TTS5:Receive_Data_Out

If this state was entered with an Enable First Burst Data-Out state machine argument, then this transition shall occur after a Request (Receive Data-Out) message is received.

8.2.6.3.3.2.5 Transition ST_TTS1:Target_Start to ST_TTS7:Prepare_Response

This transition shall occur after receiving a Request (Send Transport Response) message.

The transition shall include the Transport Response arguments.

8.2.6.3.3.3 ST_TTS2:Target_Send_Frame state

8.2.6.3.3.3.1 State description

If this state is entered from the ST_TTS3:Prepare_Data_In state for transmission of a read DATA frame, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer.

If this state is entered from the ST_TTS4:Prepare_Xfer_Rdy state for transmission of an XFER_RDY frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_TTS6:Prepare_Response state for transmission of a RESPONSE frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

All Transmit Frame requests from this state shall include the read DATA frame from the ST_TTS3:Prepare_Data_In state, the XFER_RDY frame from the ST_TTS4:Prepare_Xfer_Rdy state, or the RESPONSE frame from the ST_TTS6:Prepare_Response state and the following arguments to be used for any OPEN address frame:

- a) Initiator Port Bit set to zero;
- b) Protocol set to SSP;
- c) Connection Rate;
- d) Initiator Connection Tag;
- e) Destination SAS Address; and
- f) Source SAS Address.

After sending a Transmit Frame request this state shall wait to receive a Transmission Status confirmation.

If the confirmation or argument is Transmission Status (I_T Nexus Loss), then this state shall send a Transmission Complete (I_T Nexus Loss) message to the ST_TFR state machine. The Transmission Complete message shall include the initiator port transfer tag as an argument.

If the confirmation or argument is not Transmission Status (Frame Transmitted) or Transmission Status (I_T Nexus Loss), then this state shall send the Transmission Complete message defined in table 142 to the ST_TFR state machine. The message shall include the following arguments:

- a) Initiator Port Transfer Tag; and
- b) arguments received with the Transmission Status confirmation.

If the confirmation is Transmission Status (Frame Transmitted) and the Transmit Frame request was for:

- a) an XFER_RDY frame; or
- b) a RESPONSE frame,

then this state shall wait to receive one of the following confirmations:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

If the confirmation is Transmission Status (Frame Transmitted) and the Transmit Frame request was for a read DATA frame, then this state shall:

- a) increment the Read Data Frames Transmitted state machine variable by one; and
- b) set the Read Data Offset state machine variable to the current Read Data Offset state machine variable plus the number of read data bytes transmitted in the DATA frame associated with the Transmission Status (Frame Transmitted) confirmation.

If the confirmation is Transmission Status (ACK Received) and the Transmit Frame request was for a read DATA frame, then this state shall increment the Read Data Frames ACKed state machine variable by one.

If the confirmation is Transmission Status (Frame Transmitted), the Transmit Frame request was for a read DATA frame, and the Read Data Offset state machine variable is equal to the Read Data Buffer End state machine variable, then this state shall wait to receive:

- a) Transmission Status (ACK Received) confirmations or arguments for each outstanding read DATA frame (i.e., Read Data Frames Transmitted state machine variable equals the Read Data Frames ACKed state machine variable); or
- b) one of the following confirmations:
 - A) Transmission Status (NAK Received);
 - B) Transmission Status (ACK/NAK Timeout); or
 - C) Transmission Status (Connection Lost Without ACK/NAK).

NOTE 79 If the number of data bytes that have been transmitted for a Request (Send Data-In) message are fewer than the Request Byte Count Data-In state machine argument, then this state transitions to the ST_TTS3:Prepare_Data_In state to construct the additional read DATA frames for the request before receiving a Transmission Status (ACK Received), Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK) confirmation.

When the Read Data Frames Transmitted state machine variable equals the Read Data Frames ACKed state machine variable and the Transmit Frame request was for a read DATA frame, this state shall:

- a) not modify the Balance Point Read Data Offset state machine variable (i.e., the balance point remains at the last point at which balance occurred); or
- b) set the Balance Point Read Data Offset state machine variable to the current Read Data Offset state machine variable.

If the Transmit Frame request was for a RESPONSE frame, the vendor specific number of retries has not been reached, and this state receives one of the following confirmations:

- a) Transmission Status (NAK Received);
- b) Transmission Status (ACK/NAK Timeout); or

- c) Transmission Status (Connection Lost Without ACK/NAK),

then this state shall:

- a) set the RETRANSMIT bit to one;
- b) set the other fields to the same values as contained in the failed RESPONSE frame; and
- c) resend a Transmit Frame (Interlocked) request to the port layer for the failed RESPONSE frame.

If transport layer retries are enabled, the Transmit Frame request was for a XFER_RDY frame, the vendor specific number of retries has not been reached, and this state receives one of the following confirmations:

- a) Transmission Status (NAK Received);
- b) Transmission Status (ACK/NAK Timeout); or
- c) Transmission Status (Connection Lost Without ACK/NAK),

then this state shall:

- a) set the RETRANSMIT bit to one;
- b) set the TARGET PORT TRANSFER TAG field to a value that is different than the target port transfer tag in the previous XFER_RDY frame associated with the Data-Out state machine arguments and is different than any other target port transfer tag currently in use. If write data is received for a subsequent XFER_RDY frame for a command, then all target port transfer tags used for previous XFER_RDY frames for the command are no longer in use;
- c) set the other fields to the same values contained in the failed XFER_RDY frame; and
- d) resend a Transmit Frame (Interlocked) request to the port layer for the failed XFER_RDY frame.

Table 142 defines messages that this state shall send to the ST_TFR state machine upon receipt of the listed confirmations and arguments, based on the conditions under which each confirmation or argument was received.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

Table 142 — Messages sent to the ST_TFR state machine

Confirmation received from the port layer or argument received from ST_TTS3:Prepare_Data_In	Conditions under which confirmation was received	Message sent to the ST_TFR state machine
Transmission Status (ACK Received)	The Transmit Frame request was for an XFER_RDY frame.	Transmission Complete (Xfer_Rdy Delivered) with a Target Port Transfer Tag argument
	The Transmit Frame request was for a RESPONSE frame.	Transmission Complete (Response Delivered)
	The Transmit Frame request was for a read DATA frame and: a) the Read Data Offset state machine variable is equal to the Read Data Buffer End state machine variable; and b) the Read Data Offset state machine variable is equal to the Balance Point Read Data Offset state machine variable.	Transmission Complete (Data-In Delivered)
Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK)	The Transmit Frame request was for a RESPONSE frame and the vendor specific number of retries has been reached.	Transmission Complete (Response Failed)
Transmission Status (NAK Received)	The Transmit Frame request was for an XFER_RDY frame and: a) if transport layer retries are disabled; or b) if transport layer retries are enabled and the vendor specific number of retries has been reached.	Transmission Complete (Xfer_Rdy Failed, NAK Received)
Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK)		Transmission Complete (Xfer_Rdy Failed, Connection Failed)
Transmission Status (NAK Received)	The Transmit Frame request was for a read DATA frame and: a) if transport layer retries are disabled; or b) if transport layer retries are enabled and the vendor specific number of retries has been reached.	Transmission Complete (Data-In Failed, NAK Received)
Transmission Status (ACK/NAK Timeout)		Transmission Complete (Data-In Failed, ACK/NAK Timeout)
Transmission Status (Connection Lost Without ACK/NAK)		Transmission Complete (Data-In Failed, Connection Lost Without ACK/NAK)

Table 143 defines messages that this state shall send to the ST_TFR state machine upon receipt of the listed confirmations and arguments.

Table 143 — Additional messages sent to the ST_TFR state machine

Confirmation received from the port layer or argument received from ST_TTS3:Prepare_Data_In	Message sent to the ST_TFR state machine
Transmission Status (Break Received)	Transmission Complete (Data Transfer Terminated)

If this state receives a Cancel message or a Cancel argument and this state has received confirmations for all Transmit Frame requests sent to the port layer, then this state shall send a Transmission Complete (Data Transfer Terminated) message to the ST_TFR state machine.

If this state receives a Cancel message or a Cancel argument and this state has not received confirmations for all Transmit Frame requests sent to the port layer, then this state shall send a Cancel request to the port layer to cancel previous Transmit Frame requests. The Cancel request shall include the following arguments:

- a) Destination SAS Address; and
- b) Initiator Port Transfer Tag.

Upon receipt of a Transmission Status (Cancel Acknowledged) confirmation or argument this state shall send a Transmission Complete (Data Transfer Terminated) message to the ST_TFR state machine.

A Transmission Complete message to the ST_TFR state machine shall include the following arguments:

- a) Destination SAS Address; and
- b) Initiator Port Transfer Tag.

8.2.6.3.3.3.2 Transition ST_TTS2:Target_Send_Frame to ST_TTS1:Target_Start

This transition shall occur after sending a Transmission Complete message other than the Transmission Complete (Xfer_Rdy Delivered) message to the ST_TFR state machine.

8.2.6.3.3.3.3 Transition ST_TTS2:Target_Send_Frame to ST_TTS3:Prepare_Data_In

This transition shall occur after receiving a Transmission Status (Frame Transmitted) confirmation for a read DATA frame if the Read Data Offset state machine variable is less than the Read Data Buffer End state machine variable (i.e., there is more read data to transfer).

If transport layer retries are enabled and the vendor specific number of retries, if any, for the read DATA frame has not been reached, then this transition shall occur and include a Retry argument after receiving one of the following confirmations for a read DATA frame:

- a) Transmission Status (NAK Received);
- b) Transmission Status (ACK/NAK Timeout); or
- c) Transmission Status (Connection Lost Without ACK/NAK).

8.2.6.3.3.3.4 Transition ST_TTS2:Target_Send_Frame to ST_TTS5:Receive_Data_Out

This transition shall occur after sending a Transmission Complete (Xfer_Rdy Delivered) message to the ST_TFR state machine.

8.2.6.3.3.4 ST_TTS3:Prepare_Data_In state

8.2.6.3.3.4.1 State description

This state retrieves the data from the Device Server Buffer Data-In state machine argument and constructs a read DATA frame.

This state shall construct a read DATA frame using the Data-In state machine arguments as follows:

- a) FRAME TYPE field set to 01h (i.e., DATA frame);
- b) HASHED DESTINATION SAS ADDRESS field set to the hashed value of the Destination SAS Address Data-In state machine argument;
- c) HASHED SOURCE SAS ADDRESS field set to the hashed value of the SSP target port's SAS address;
- d) RETRY DATA FRAMES bit set to zero;
- e) RETRANSMIT bit set to zero;
- f) CHANGING DATA POINTER set as specified in this subclause;
- g) NUMBER OF FILL BYTES field set to the number of fill bytes needed for the specified read data;
- h) INITIATOR PORT TRANSFER TAG field set to the Initiator Port Transfer Tag Data-In state machine argument;
- i) TARGET PORT TRANSFER TAG field set to a vendor specific value;
- j) DATA OFFSET field set as specified in this subclause;
- k) in the information unit, DATA field set as specified in this subclause; and
- l) fill bytes, if required.

If this state is entered without a Retry argument then this state shall:

- a) set the CHANGING DATA POINTER bit set to zero;
- b) set the DATA OFFSET field to the Read Data Offset state machine variable; and
- c) in the information unit, set the DATA field to the information in the Device Server Buffer argument that corresponds to the read data to be transferred. If the Read Data Buffer End state machine variable minus the Read Data Offset state machine variable is equal to the maximum size of the read Data information unit, then the amount of data shall be the maximum size of the read Data information unit. Otherwise, the amount of data shall be the lesser of:
 - A) the Read Data Buffer End state machine variable minus the Read Data Offset state machine variable; and
 - B) the maximum size of the read Data information unit for this Send Data-In request.

If this state is entered with a Retry argument then this state shall either:

- a) set the CHANGING DATA POINTER bit in the frame to one;
- b) set the DATA OFFSET field to the Balance Point Read Data Offset state machine variable;
- c) set the Read Data Offset state machine variable to the Balance Point Read Data Offset state machine variable;
- d) set the Read Data Frames Transmitted state machine variable to zero;
- e) set the Read Data Frames ACKed state machine variable to zero; and
- f) in the information unit, set the DATA field to the information in the Device Server Buffer argument that corresponds to the read data to be transferred. If the Read Data Buffer End state machine variable minus the Read Data Offset state machine variable is equal to the maximum size of the read Data information unit, then the amount of data shall be the maximum size of the read Data information unit. Otherwise, the amount of data shall be the lesser of:
 - A) the Read Data Buffer End state machine variable minus the Balance Point Read Data Offset state machine variable; and
 - B) the maximum size of the read Data information unit for this Send Data-In request;

or:

- a) set the CHANGING DATA POINTER bit in the frame to one;
- b) set the DATA OFFSET field to the Application Client Buffer Offset Data-In state machine argument;
- c) set the Read Data Offset state machine variable to the Application Client Buffer Offset Data-In state machine argument;
- d) set the Read Data Frames Transmitted state machine variable to zero;
- e) set the Read Data Frames ACKed state machine variable to zero; and
- f) in the information unit, set the DATA field to the information in the Device Server Buffer argument that corresponds to the read data to be transferred. If the Request Byte Count Data-In state machine argument is equal to the maximum size of the read Data information unit, then the amount of data shall be the maximum size of the read Data information unit. Otherwise, the amount of data shall be the lesser of:
 - A) the Request Byte Count Data-In state machine argument; and
 - B) the maximum size of the read Data information unit for this Send Data-In request.

8.2.6.3.3.4.2 Transition ST_TTS3:Prepare_Data_In to ST_TTS2:Target_Send_Frame

This transition shall occur after this state:

- a) constructs a read DATA frame; or
- b) receives a Cancel message.

This transition shall include the received Transmission Status, if any, as an argument and:

- a) if a Cancel message was not received, then the read DATA frame as an argument; or
- b) if a Cancel message was received, then a Cancel argument.

8.2.6.3.3.5 ST_TTS4:Prepare_Xfer_Rdy state

8.2.6.3.3.5.1 State description

This state shall construct an XFER_RDY frame using the Data-Out state machine arguments:

- a) FRAME TYPE field set to 05h (i.e., XFER_RDY frame);
- b) HASHED DESTINATION SAS ADDRESS field set to the hashed value of the Destination SAS Address Data-Out state machine argument;
- c) HASHED SOURCE SAS ADDRESS field set to the hashed value of the SSP target port's SAS address;
- d) RETRY DATA FRAMES bit set to one if transport layer retries are enabled and zero if transport layer retries are disabled;
- e) RETRANSMIT bit set to zero;
- f) CHANGING DATA POINTER bit set to zero;
- g) NUMBER OF FILL BYTES field set to 00b;
- h) INITIATOR PORT TRANSFER TAG field set to the Initiator Port Transfer Tag Data-Out state machine argument;
- i) if transport layer retries are disabled, TARGET PORT TRANSFER TAG field set to a vendor specific value;
- j) if transport layer retries are enabled, TARGET PORT TRANSFER TAG field set to a vendor specific value that is different from:
 - A) the target port transfer tag in the previous XFER_RDY frame associated with the Data-Out state machine arguments; and
 - B) any other target port transfer tag currently in use.

If write data is received for a subsequent XFER_RDY frame for a command, then all target port transfer tags used for previous XFER_RDY frames for the command are no longer in use;

- k) DATA OFFSET field set to 00000000h;
- l) in the information unit, REQUESTED OFFSET field set to the Requested Write Data Offset state machine variable;
- m) in the information unit, WRITE DATA LENGTH field set as specified in this subclause; and
- n) no fill bytes.

If the SSP target port has the resources available to receive all of the write data as indicated by the Requested Write Data Length state machine variable, then this state shall set the WRITE DATA LENGTH field in the XFER_RDY information unit to the Requested Write Data Length state machine variable.

If the SSP target port does not have the resources available to receive all of the write data as indicated by the Requested Write Data Length state machine variable (e.g., the SSP target port has a vendor specific limit as to how much write data may be received during one operation), then this state shall set the WRITE DATA LENGTH field in the XFER_RDY information unit and the Requested Write Data Length state machine variable to a value representing the amount of write data for which the SSP target port has available resources to receive.

8.2.6.3.3.5.2 Transition ST_TTS4:Prepare_Xfer_Rdy to ST_TTS2:Target_Send_Frame

This transition shall occur after this state:

- a) constructs an XFER_RDY frame; or
- b) receives a Cancel message.

This transition shall include:

- a) if a Cancel message was not received, then the XFER_RDY frame as an argument; or
- b) if a Cancel message was received, then a Cancel argument.

8.2.6.3.3.6 ST_TTS5:Receive_Data_Out state

8.2.6.3.3.6.1 State description

Upon entry into this state, the Write Data Offset state machine variable is set to the Requested Write Data Offset state machine variable.

If this state receives a Data-Out Arrived message, then this state shall verify the write DATA frame received with the Data-Out Arrived values as specified in table 144. If the verification test fails, then this state sends the message specified in table 144 to the ST_TFR state machine.

Table 144 — Reception Complete message for write DATA frame verification failures

Message sent to ST_TFR ^a	Condition
Reception Complete (Data Offset Error)	Transport layer retries are disabled, and the DATA OFFSET field is not equal to the Write Data Offset state machine variable.
	The DATA OFFSET field is: a) less than the Requested Write Data Offset state machine variable; or b) greater than or equal to the Requested Write Data Offset state machine variable plus the Requested Write Data Length state machine variable
Reception Complete (Too Much Write Data)	The number of bytes in the DATA field in the write Data information unit plus the Write Data Offset state machine variable is greater than the Request Byte Count Data-Out state machine argument.
Reception Complete (Information Unit Too Short)	Either: a) the number of bytes in the DATA field in the write Data information unit is zero; or b) this is not the last write DATA frame for the command and the NUMBER OF FILL BYTES field for the frame is not set to 00b.
^a If more than one condition is true, then this state shall select which message to send to the ST_TFR state machine using the following order: 1) Reception Complete (Data Offset Error); 2) Reception Complete (Too Much Write Data); or 3) Reception Complete (Information Unit Too Short).	

If:

- a) transport layer retries are enabled;
- b) the CHANGING DATA POINTER bit is set to zero; and
- c) the value in the DATA OFFSET field is not equal to the Write Data Offset state machine variable,

then this state should discard all Data-Out Arrived messages until the CHANGING DATA POINTER bit is set to one. This state shall resume processing additional Data-Out Arrived messages when it receives a Data-Out Arrived message with the CHANGING DATA POINTER bit set to one.

If the WRITE data frame verification is successful and the Data-Out Arrived message is not discarded, then this state shall:

- a) process the write data as indicated in the Data-Out state machine arguments using the Device Server Buffer (e.g., logical block address) to which the write data is to be transferred; and
- b) set the Write Data Offset state machine variable to the current Write Data Offset state machine variable plus the number of bytes received in the DATA field of the write Data information unit.

If the WRITE data frame verification is successful and the CHANGING DATA POINTER bit set to one, then this state shall:

- a) set the Write Data Offset state machine variable to the Requested Write Data Offset state machine variable plus the number of bytes received in the DATA field of the write Data information unit; and
- b) process the write data as indicated in the Data-Out state machine arguments using the Device Server Buffer (e.g., logical block address) to which the write data is to be transferred.

If data received in the write DATA frame overlaps data previously received and verified to have no errors, then this state may either discard the overlapping data or replace the previously received data with the new data.

If the Initiator Response Timeout timer is implemented, then this state shall initialize and start the Initiator Response Timeout timer:

- a) upon entry into this state; and
- b) when this state receives and verifies the write DATA frame received with the Data-Out Arrived values (i.e., Data-Out data was received and processed).

If the Initiator Response Timeout timer is running, then this state shall stop the timer before transitioning from this state.

If the Initiator Response Timeout timer expires, then this state shall send a Reception Complete (Initiator Response Timeout) message to the ST_TFR state machine.

If the Write Data Offset state machine variable equals the Request Byte Count Data-Out state machine argument plus the Application Client Buffer Offset Data-Out state machine argument, then this state shall send a Reception Complete (Data-Out Received) message to the ST_TFR state machine after an ACK Transmitted confirmation is received for each write DATA frame previously received.

If this state receives a Cancel message, then this state shall send a Reception Complete (Data Transfer Terminated) message to the ST_TFR state machine.

If this state receives Transmission Status (Break Received) confirmation, then this state shall send a Reception Complete (Data Transfer Terminated) message to the ST_TFR state machine.

The Reception Complete message, if any, shall include the initiator port transfer tag as an argument.

8.2.6.3.3.6.2 Transition ST_TTS5:Receive_Data_Out to ST_TTS1:Target_Start

This transition shall occur after sending a Reception Complete message to the ST_TFR state machine.

8.2.6.3.3.6.3 Transition ST_TTS5:Receive_Data_Out to ST_TTS4:Prepare_Xfer_Rdy

This transition shall occur:

- 1) if the Write Data Offset state machine variable is less than Request Byte Count Data-Out state machine argument plus the Application Client Buffer Offset Data-Out state machine argument and equal to the Requested Write Data Offset state machine variable plus the Requested Write Data Length state machine variable;
- 2) after an ACK Transmitted confirmation is received for each write DATA frame previously received;
- 3) after determining the amount of write data already transferred by subtracting the Application Client Buffer Offset Data-Out state machine argument from the Write Data Offset state machine variable;
- 4) after setting the Requested Write Data Length state machine variable to the Request Byte Count Data-Out state machine argument minus the amount of write data already transferred; and
- 5) after setting the Requested Write Data Offset state machine variable to the Write Data Offset state machine variable.

8.2.6.3.3.7 ST_TTS6:Prepare_Response state

8.2.6.3.3.7.1 State description

This state shall construct a RESPONSE frame using the received Application Response arguments or the received Transport Response arguments as follows:

- a) FRAME TYPE field set to 07h (i.e., RESPONSE frame);
- b) HASHED DESTINATION SAS ADDRESS field set to the hashed value of the Application Response or Transport Response Destination SAS Address argument;
- c) HASHED SOURCE SAS ADDRESS field set to the hashed value of the SSP target port's SAS address;
- d) RETRY DATA FRAMES bit set to zero;
- e) RETRANSMIT bit set to zero;
- f) CHANGING DATA POINTER bit set to zero;
- g) INITIATOR PORT TRANSFER TAG field set to the Initiator Port Transfer Tag Application Response argument or the Initiator Port Transfer Tag Transport Response argument;
- h) TARGET PORT TRANSFER TAG field set to a vendor specific value;
- i) DATA OFFSET field set to 00000000h;
- j) information unit set as specified in this subclause; and
- k) fill bytes, if needed as specified in this subclause.

If this state was entered with the Transport Response arguments, then this state shall set the fields as follows:

- a) NUMBER OF FILL BYTES field set to the number of fill bytes, based on the length of the response data, if any;
- b) in the information unit, set the DATAPRES field to RESPONSE DATA;
- c) in the information unit, set the STATUS field to zero;
- d) in the information unit, set the STATUS QUALIFIER field to 0000h;
- e) in the information unit, set the SENSE DATA LENGTH field to 00000000h;
- f) in the information unit, set the RESPONSE DATA LENGTH field to 00000004h;
- g) in the information unit, set the RESPONSE DATA field as specified in table 145; and
- h) in the information unit, do not include the SENSE DATA field.

Table 145 defines how the RESPONSE DATA field shall be set based on the arguments received with the Request (Send Transport Response) message.

Table 145 — Request argument to RESPONSE frame response data field mapping

Request argument	RESPONSE frame response data field
Invalid Frame	INVALID FRAME
Function Complete	TASK MANAGEMENT FUNCTION COMPLETE
Function Succeeded	TASK MANAGEMENT FUNCTION SUCCEEDED
Function Not Supported	TASK MANAGEMENT FUNCTION NOT SUPPORTED
Function Failed	TASK MANAGEMENT FUNCTION FAILED
Incorrect Logical Unit Number	INCORRECT LOGICAL UNIT NUMBER
Overlapped Initiator Port Transfer Tag Attempted	OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED

If this state was entered with the Application Response arguments, then this state shall set the fields as follows:

- a) in the information unit, set the DATAPRES field to SENSE_DATA if sense data is to be included in the information unit or NO_DATA if sense data is not to be included in the information unit;
- b) in the information unit, set the STATUS field to the status;
- c) in the information unit, set the STATUS QUALIFIER field to the status qualifier, if any;

- d) in the information unit, set the SENSE DATA LENGTH field to the length of the sense data, if any;
- e) in the information unit, set the RESPONSE DATA LENGTH field to 00000000h;
- f) in the information unit, do not include the RESPONSE DATA field;
- g) in the information unit, set the SENSE DATA field to the sense data, if any; and
- h) NUMBER OF FILL BYTES field set to the number of fill bytes, based on the length of the sense data, if needed.

8.2.6.3.3.7.2 Transition ST_TTS6:Prepare_Response to ST_TTS2:Target_Send_Frame

This transition shall occur after this state constructs a RESPONSE frame.

This transition shall include:

- a) if a Cancel message was not received, then the RESPONSE frame as an argument; or
- b) if a Cancel message was received, then a Cancel argument.

8.3 STP transport layer

8.3.1 Initial FIS

A SATA device phy transmits a Register - Device to Host FIS after completing the link reset sequence, except for the case described in F.5. The expander device shall update a set of shadow registers with the contents of this FIS and shall not deliver this FIS to any STP initiator port. SMP initiator ports may read the shadow register contents using the SMP REPORT PHY SATA function (see 9.4.3.12). The expander device originates a Broadcast (Change) after receiving the Register - Device to Host FIS (see 6.12).

8.3.2 BIST Activate FIS

STP initiator ports and STP target ports shall not generate BIST Activate FISes and shall process any BIST Activate FISes received as frames having invalid FIS types (i.e., have the link layer generate SATA_R_ERR in response).

8.3.3 TT (transport layer for STP ports) state machines

The STP transport layer uses the transport layer state machines defined in SATA, modified to communicate with the port layer rather than directly with the link layer. These modifications are not described in this standard.

8.4 SMP transport layer

8.4.1 SMP transport layer overview

Table 146 defines the SMP frame format.

Table 146 — SMP frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE							
1	Frame-type dependent bytes							
n - 4								
n - 3	(MSB)	CRC						
n								(LSB)

Table 147 defines the SMP FRAME TYPE field, which defines the format of the frame-type dependent bytes.

Table 147 — SMP FRAME TYPE field

Code	Name	Frame type	Originator	Reference
40h	SMP_REQUEST	SMP function request	SMP initiator port	8.4.2
41h	SMP_RESPONSE	SMP function response	SMP target port	8.4.3
All others	Reserved.			

The number of frame-type dependent bytes shall be either:

- a) three bytes; or
- b) three bytes plus an integer multiple of four bytes,

so the CRC field is aligned on a four byte boundary.

The CRC field contains a CRC value (see 6.5) that is computed over the entire SMP frame prior to the CRC field, and shall begin on a four-byte boundary. The CRC field is checked by the SMP link layer (see 6.19).

The SMP_REQUEST frame is sent by an SMP initiator port to request an SMP function be performed by a management device server. Table 148 defines the SMP_REQUEST frame format.

Table 148 — SMP_REQUEST frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	Request bytes							
n - 4								
n - 3								
n	CRC						(LSB)	

The SMP FRAME TYPE field shall be set to the value defined in table 148.

The format and length of the request bytes is defined by the SMP function description (see 9.4.3.2).

The number of request bytes are either:

- a) three bytes; or
- b) three bytes plus an integer multiple of four bytes,

so the CRC field is aligned on a four byte boundary.

The maximum number of request bytes is 1 023, making the maximum size of the frame 1 028 bytes (i.e., 1 byte header + 1 023 request bytes + 4 bytes of CRC).

NOTE 80 Management application clients compliant with SAS-1.1 may send a vendor specific SMP request frame containing 1 027 request bytes. The SMP_TP state machine discards SMP request frames that exceed 1 023 request bytes (see 6.19.5.4.2.2). SMP request frames defined in those versions of this standard did not have more than 39 request bytes.

The CRC field is defined in 8.4.1.

8.4.3 SMP_RESPONSE frame

The SMP_RESPONSE frame is sent by an SMP target port in response to an SMP_REQUEST frame. Table 149 defines the SMP_RESPONSE frame format.

Table 149 — SMP_RESPONSE frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	Response bytes							
n - 4								
n - 3								
n	CRC						(LSB)	

The SMP FRAME TYPE field shall be set to the value defined in table 149.

The format and length of the request bytes is defined by the SMP function description (see 9.4.3.3).

The number of response bytes are either:

- a) three bytes; or
- b) three bytes plus an integer multiple of four bytes,

so the CRC field is aligned on a four byte boundary.

The maximum number of response bytes is 1 023, making the maximum size of the frame 1 028 bytes (i.e., 1 byte header + 1 023 request bytes + 4 bytes of CRC).

NOTE 81 Management device servers compliant with SAS-1.1 may send a vendor specific SMP response frame containing 1 027 response bytes. The SMP_IP state machine discards SMP response frames that exceed 1 023 request bytes (see 6.19.5.3.4). SMP response frames defined in those versions of this standard did not have more than 59 request bytes.

The CRC field is defined in 8.4.1.

8.4.4 Sequence of SMP frames

Inside an SMP connection, the SMP initiator port transmits a single SMP_REQUEST frame and the SMP target port replies with a single SMP_RESPONSE frame.

Figure 145 shows the sequence of SMP frames.

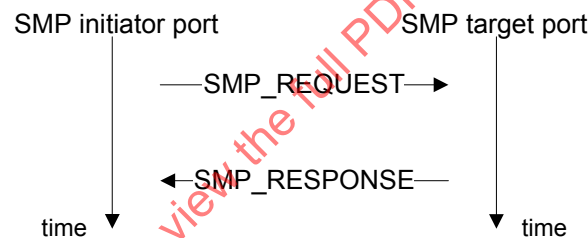


Figure 145 — Sequence of SMP frames

8.4.5 MT (transport layer for SMP ports) state machines

8.4.5.1 SMP transport layer state machines overview

The SMP transport layer contains state machines that process requests from the management application layer and return confirmations to the management application layer. The SMP transport state machines are as follows:

- a) MT_IP (transport layer for SMP initiator ports) state machine (see 8.4.5.2); and
- b) MT_TP (transport layer for SMP target ports) state machine (see 8.4.5.3).

8.4.5.2 MT_IP (transport layer for SMP initiator ports) state machine

8.4.5.2.1 MT_IP state machine overview

The MT_IP state machine processes requests from the management application layer. These management requests are sent to the port layer, and the resulting SMP frame or error condition is sent to the management application layer as a confirmation.

This state machine consists of the following states:

- a) MT_IP1:Idle (see 8.4.5.2.2) (initial state);
- b) MT_IP2:Send (see 8.4.5.2.3); and

- c) MT_IP3:Receive (see 8.4.5.2.4).

This state machine shall start in the MT_IP1:Idle state.

Figure 146 describes the MT_IP state machine.

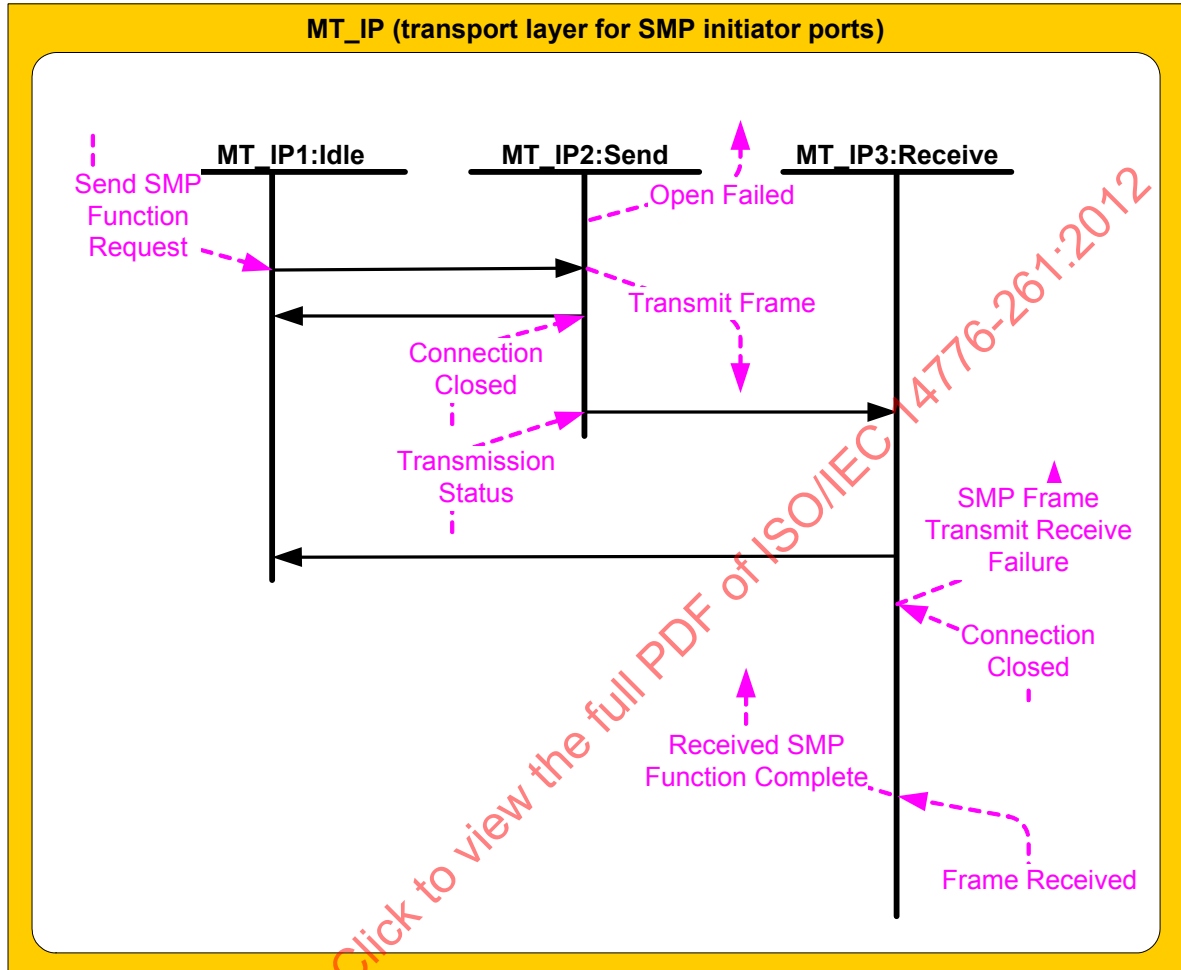


Figure 146 — MT_IP (transport layer for SMP initiator ports) state machine

8.4.5.2.2 MT_IP1:Idle state

8.4.5.2.2.1 State description

This state is the initial state of the MT_IP state machine.

This state waits for a Send SMP Function Request request, which includes the following arguments:

- Connection Rate;
- Destination SAS Address; and
- Request Bytes.

8.4.5.2.2.2 Transition MT_IP1:Idle to MT_IP2:Send

This transition shall occur after a Send SMP Function Request request is received. This transition shall include the following arguments:

- Connection Rate;
- Destination SAS Address; and

- c) Request Bytes.

8.4.5.2.3 MT_IP2:Send state

8.4.5.2.3.1 State description

This state constructs an SMP_REQUEST frame using the following arguments received in the transition into this state:

- a) Request Bytes;

and sends a Transmit Frame request to the port layer with the following arguments:

- a) Initiator Port bit set to one;
- b) Protocol set to SMP;
- c) Connection Rate;
- d) Initiator Connection Tag set to FFFFh;
- e) Destination SAS Address;
- f) Source SAS Address set to the SAS address of the SMP initiator port; and
- g) Request Bytes.

8.4.5.2.3.2 Transition MT_IP2:Send to MT_IP1:Idle

This transition shall occur after receiving either a Connection Closed confirmation or a Transmission Status confirmation other than a Transmission Status (Frame Transmitted) confirmation, and after sending an Open Failed confirmation to the management application layer.

8.4.5.2.3.3 Transition MT_IP2:Send to MT_IP3:Receive

This transition shall occur after receiving a Transmission Status (Frame Transmitted) confirmation.

8.4.5.2.4 MT_IP3:Receive state

8.4.5.2.4.1 State description

This state waits for a confirmation from the port layer that either an SMP frame has been received or a failure occurred.

If a Frame Received (SMP Successful) confirmation is received and the SMP frame type is equal to 41h, then this state shall send a Received SMP Function Complete confirmation to the management application layer.

If a Frame Received (SMP Successful) confirmation is received and the SMP frame type is not equal to 41h, then this state shall send an SMP Frame Transmit Receive Failure confirmation to the management application layer.

If a Connection Closed or Frame Received (SMP Unsuccessful) confirmation is received, then this state shall send an SMP Frame Transmit Receive Failure confirmation to the management application layer.

8.4.5.2.4.2 Transition MT_IP3:Receive to MT_IP1:Idle

This transition shall occur after one of the following:

- a) sending a Received SMP Function Complete confirmation; or
- b) sending an SMP Frame Transmit Receive Failure confirmation.

8.4.5.3 MT_TP (transport layer for SMP target ports) state machine

8.4.5.3.1 MT_TP state machine overview

The MT_TP state machine informs the management application layer of the receipt of an SMP frame, and sends the resulting SMP frame to the port layer.

This state machine consists of the following states:

- a) MT_TP1:Idle (see 8.4.5.3.2) (initial state); and
- b) MT_TP2:Respond (see 8.4.5.3.3).

This state machine shall start in the MT_TP1:Idle state.

Figure 147 describes the MT_TP state machine.

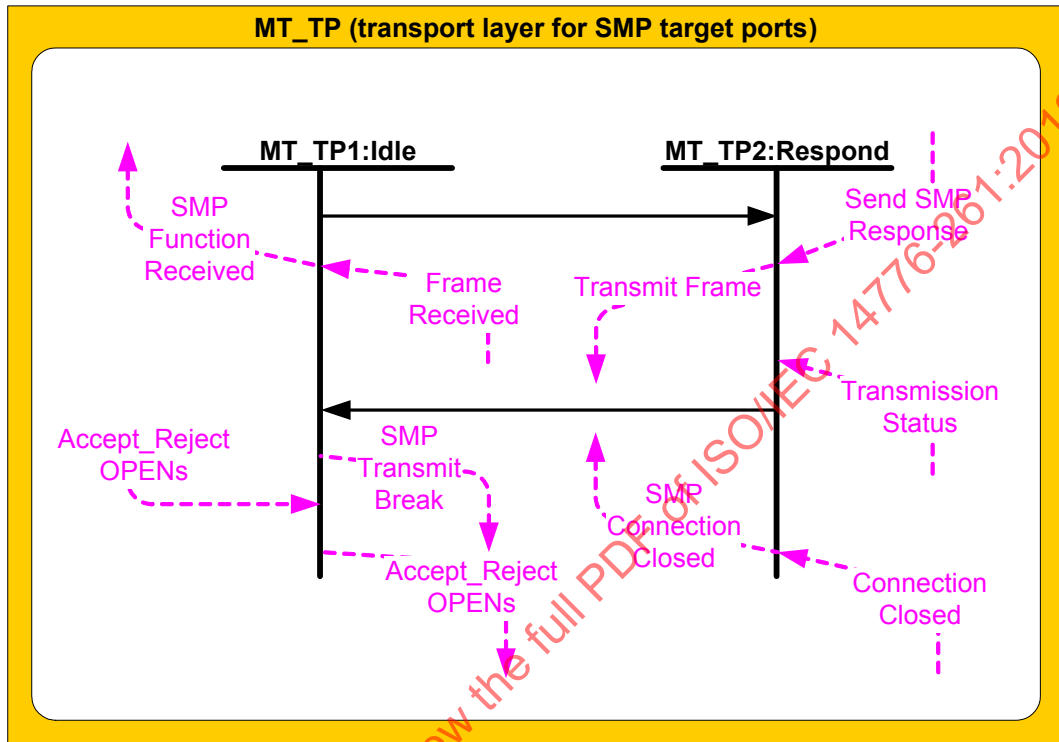


Figure 147 — MT_TP (transport layer for SMP target ports) state machine

The MT_TP state machine shall comply with the time limits listed in table 150.

Table 150 — MT_TP time limits

Time limit	Value	Description
SMP Response time limit	1 900 µs	Maximum time from receiving an SMP_REQUEST frame to transmitting an SMP_RESPONSE frame

8.4.5.3.2 MT_TP1:Idle state

8.4.5.3.2.1 State description

This state is the initial state of the MT_TP state machine.

This state waits for a Frame Received (SMP Successful) confirmation. If the SMP frame type is not equal to 40h, then this state shall discard the frame and send an SMP Transmit Break request to the port layer. Otherwise, this state shall send an SMP Function Received confirmation to the management application layer.

If an Accept_Reject OPENs (Accept SMP) request or an Accept_Reject OPENs (Reject SMP) request is received, then this state shall send an Accept_Reject OPENs request with the same arguments to the port layer.

8.4.5.3.2.2 Transition MT_TP1:Idle to MT_TP2:Respond

This transition shall occur after sending an SMP Function Received confirmation.

8.4.5.3.3 MT_TP2:Respond state**8.4.5.3.3.1 State description**

This state waits for a Send SMP Response request, which includes the following argument:

- a) Response Bytes.

After receiving a Send SMP Response request, this state shall construct an SMP_RESPONSE frame using the arguments from the Send SMP Response request and send a Transmit Frame request to the port layer within the SMP Response time limit specified in table 150 (see 8.4.5.3.1).

If this state receives a Connection Closed confirmation, then this state shall send an SMP Connection Closed confirmation to the management application layer.

8.4.5.3.3.2 Transition MT_TP2:Respond to MT_TP1:Idle

This transition shall occur after one of the following:

- a) receiving a Transmission Status (Frame Transmitted) confirmation; or
- b) sending an SMP Connection Closed confirmation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 14776-261:2012

9 Application layer

9.1 Application layer overview

The application layer defines SCSI, ATA, and management specific features.

9.2 SCSI application layer

9.2.1 SCSI transport protocol services

9.2.1.1 SCSI transport protocol services overview

An application client requests the processing of a SCSI command by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following procedure call (see SAM-5):

Service response = Execute Command (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Priority]),
OUT ([Data-In Buffer], [Sense Data], [Sense Data Length],
Status, [Status Qualifier]))

This standard defines the transport protocol services required by SAM-5 in support of these procedure calls. Table 151 describes the mapping of the Execute Command procedure call to transport protocol services and the SSP implementation of each transport protocol service.

Table 151 — Execute Command procedure call transport protocol services

Transport protocol service	I/T ^a	SSP implementation	Reference
Request/Confirmation			
Send SCSI Command request	I	COMMAND frame	9.2.1.2
SCSI Command Received indication	T	Receipt of the COMMAND frame	9.2.1.3
Send Command Complete response	T	RESPONSE frame	9.2.1.4
Command Complete Received confirmation	I	Receipt of the RESPONSE frame or problem transmitting the COMMAND frame	9.2.1.5
Data-In Transfer ^b			
Send Data-In request	T	Read DATA frames	9.2.1.6
Data-In Delivered confirmation	T	Receipt of ACKs for the read DATA frames	9.2.1.7
Data-Out Transfer ^b			
Receive Data-Out request	T	XFER_RDY frame	9.2.1.8
Data-Out Received confirmation	T	Receipt of write DATA frames	9.2.1.9
Terminate Data Transfer ^b			
Terminate Data Transfer request	T		9.2.1.10
Data Transfer Terminated confirmation	T		9.2.1.11
^a I/T indicates whether the SSP initiator port (I) or the SSP target port (T) implements the transport protocol service.			
^b Data transfer transport protocol services for SCSI initiator ports are not specified by SAM-5.			

An application client requests the processing of a SCSI task management function by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following procedure calls (see SAM-5):

- a) Service Response = ABORT TASK (IN (Nexus));
- b) Service Response = ABORT TASK SET (IN (Nexus));
- c) Service Response = CLEAR ACA (IN (Nexus));
- d) Service Response = CLEAR TASK SET (IN (Nexus));
- e) Service Response = I_T NEXUS RESET (IN (Nexus));
- f) Service Response = LOGICAL UNIT RESET (IN (Nexus));
- g) Service Response = QUERY TASK (IN (Nexus));
- h) Service Response = QUERY TASK SET (IN (Nexus), OUT ([Additional Response Information])); and
- i) Service Response = QUERY ASYNCHRONOUS EVENT (IN (Nexus), OUT ([Additional Response Information])).

This standard defines the transport protocol services required by SAM-5 in support of these procedure calls. Table 152 describes the mapping of these procedure calls to transport protocol services and the SSP implementation of each transport protocol service.

Table 152 — Task management function procedure call transport protocol services

Transport protocol service	I/T ^a	SSP implementation	Reference
Request/Confirmation			
Send Task Management Request request	I	TASK frame	9.2.1.12
Task Management Request Received indication	T	Receipt of the TASK frame	9.2.1.13
Task Management Function Executed response	T	RESPONSE frame	9.2.1.14
Received Task Management Function Executed confirmation	I	Receipt of the RESPONSE frame or problem transmitting the TASK frame	9.2.1.15
^a I/T indicates whether the SSP initiator port (I) or the SSP target port (T) implements the transport protocol service.			

Transport protocol services are used as the requests and confirmations to the SSP transport layer state machines (see 8.2.6) from the SCSI application layer.

9.2.1.2 Send SCSI Command transport protocol service

An application client uses the Send SCSI Command transport protocol service request to request that an SSP initiator port transmit a COMMAND frame.

Send SCSI Command (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Priority], [CRN], [First Burst Enabled], [Request Fence]))

Table 153 shows how the arguments to the Send SCSI Command transport protocol service are used.

Table 153 — Send SCSI Command transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to send the COMMAND frame; b) T specifies the target port to which the COMMAND frame is to be sent; c) L specifies the LOGICAL UNIT NUMBER field in the COMMAND frame header; and d) Q specifies the INITIATOR PORT TRANSFER TAG field in the COMMAND frame header.
CDB	Specifies the CDB field in the COMMAND frame.
Task Attribute	Specifies the TASK ATTRIBUTE field in the COMMAND frame.
[Data-In Buffer Size]	Maximum of 2^{32} bytes. ^a
[Data-Out Buffer]	Internal to the SSP initiator port.
[Data-Out Buffer Size]	Maximum of 2^{32} bytes. ^a
[CRN]	Ignored
[Command Priority]	Specifies the COMMAND PRIORITY field in the COMMAND frame.
[First Burst Enabled]	Specifies the ENABLE FIRST BURST bit in the COMMAND frame and causes the SSP initiator port to transmit the number of bytes indicated by the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 9.2.7.2.5) for the SCSI target port without waiting for an XFER_RDY frame.
[Request Fence]	If included, specifies an I_T nexus, I_T_L nexus, or I_T_L_Q nexus for which the COMMAND frame is fenced.
^a See the restrictions on the REQUESTED OFFSET field and the WRITE DATA LENGTH field in the SSP XFER_RDY frame (see 8.2.2.3) and the DATA OFFSET field in the SSP DATA frame (see 8.2.1).	

An application client shall set the Request Fence argument to the nexus containing any commands or task management functions that the command affects (e.g., for a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action, the application client sets the Response Fence argument to the I_T_L nexus) or upon which the command depends (e.g., when the Task Attribute argument is set to Ordered, the application client sets the Response Fence argument to the I_T_L_Q nexus of the previous command). If the application client is not able to determine the nexus affected by the command or upon which the command depends, then it should set the Request Fence argument to the I_T nexus.

9.2.1.3 SCSI Command Received transport protocol service

An SSP target port uses the SCSI Command Received transport protocol service indication to notify a task manager that it has received a COMMAND frame.

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

Table 154 shows how the arguments to the SCSI Command Received transport protocol service are determined.

Table 154 — SCSI Command Received transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port that sent the COMMAND frame; b) T indicates the target port that received the COMMAND frame; c) L indicates the value of the LOGICAL UNIT NUMBER field in the COMMAND frame header; and d) Q indicates the value of the INITIATOR PORT TRANSFER TAG field in the COMMAND frame header.
CDB	Indicates the value of the CDB field in the COMMAND frame.
Task Attribute	Indicates the value of the TASK ATTRIBUTE field in the COMMAND frame.
[CRN]	Ignored
[Command Priority]	Indicates the value of the COMMAND PRIORITY field in the COMMAND frame.
[First Burst Enabled]	Indicates that first burst data is being delivered based on the ENABLE FIRST BURST bit in the COMMAND frame and the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 9.2.7.2.5).

9.2.1.4 Send Command Complete transport protocol service

A device server uses the Send Command Complete transport protocol service response to request that an SSP target port transmit a RESPONSE frame.

Send Command Complete (IN (I_T_L_Q Nexus, [Sense Data], [Sense Data Length], Status, [Status Qualifier], Service Response, [Response Fence]))

A device server shall only call Send Command Complete () after receiving SCSI Command Received ().

A device server shall not call Send Command Complete () for a given I_T_L_Q nexus until:

- all its outstanding Receive Data-Out () calls for that I_T_L_Q nexus have been responded to with Data-Out Received (); and
- all its outstanding Send Data-In () calls for that I_T_L_Q nexus have been responded to with Data-In Delivered ().

Table 155 shows how the arguments to the Send Command Complete transport protocol service are used.

Table 155 — Send Command Complete transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to which the RESPONSE frame is to be sent; b) T specifies the target port to send the RESPONSE frame; c) L specifies the LOGICAL UNIT NUMBER field in the RESPONSE frame header; and d) Q specifies the INITIATOR PORT TRANSFER TAG field in the RESPONSE frame header.
[Sense Data]	Specifies the SENSE DATA field in the RESPONSE frame.
[Sense Data Length]	Specifies the SENSE DATA LENGTH field in the RESPONSE frame.
Status	Specifies the STATUS field in the RESPONSE frame.
[Status Qualifier]	Specifies the STATUS QUALIFIER field in the RESPONSE frame.
Service Response	Specifies the DATAPRES field and STATUS field in the RESPONSE frame: a) COMMAND COMPLETE: The DATAPRES field is set to NO_DATA or SENSE_DATA; or b) SERVICE DELIVERY OR TARGET FAILURE: The DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to INVALID FRAME or OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED.
[Response Fence]	If included, specifies an I_T nexus, I_T_L nexus, or I_T_L_Q nexus for which the RESPONSE frame is fenced.

A device server shall set the Response Fence argument to the nexus containing any commands or task management functions that the command affects (e.g., for a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action, the device server sets the Response Fence argument to the I_T_L nexus) or upon which the command completion depends (e.g., when returning a unit attention condition with an additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR, the device server sets the Response Fence argument to the I_T_L nexus). If the device server is not able to determine the nexus affected by the command or upon which the command depends, then it should set the Response Fence argument to the I_T nexus.

9.2.1.5 Command Complete Received transport protocol service

An SSP initiator port uses the Command Complete Received transport protocol service confirmation to notify an application client that it has received a response for its COMMAND frame (e.g., a RESPONSE frame or a NAK) or terminated a command because of an error.

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

Table 156 shows how the arguments to the Command Complete Received transport protocol service are determined.

Table 156 — Command Complete Received transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port that received the RESPONSE frame; b) T indicates the target port that sent the RESPONSE frame; c) L indicates the value of the LOGICAL UNIT NUMBER field in the RESPONSE frame header or COMMAND frame header; and d) Q indicates the value of the INITIATOR PORT TRANSFER TAG field in the RESPONSE frame header or COMMAND frame header.
[Data-In Buffer]	Internal to the SSP initiator port.
[Sense Data]	Indicates the value of the SENSE DATA field in the RESPONSE frame.
[Sense Data Length]	The smaller of the value of the SENSE DATA LENGTH field in the RESPONSE frame and the actual number of sense data bytes received by the SSP initiator port.
Status	Indicates the value of the STATUS field in the RESPONSE frame.
[Status Qualifier]	Indicates the value of the STATUS QUALIFIER field in the RESPONSE frame.
Service Response	Either: a) COMMAND COMPLETE: The RESPONSE frame contains a DATAPRES field set to NO_DATA or SENSE_DATA; or b) SERVICE DELIVERY OR TARGET FAILURE: Either: A) the RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to INVALID FRAME or OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED; or B) the ST_IFR state machine detects an error as described in 8.2.6.2.2.3 and 8.2.6.2.2.4 (e.g., a NAK was received for the COMMAND frame or the length of the RESPONSE frame is incorrect).

9.2.1.6 Send Data-In transport protocol service

A device server uses the Send Data-In transport protocol service request to request that an SSP target port transmit a read DATA frame.

Send Data-In (IN (I_T_L_Q Nexus, Device Server Buffer, Application Client Buffer Offset, Request Byte Count))

A device server shall only call Send Data-In () during a read command or bidirectional command.

A device server shall not call Send Data-In () for a given I_T_L_Q nexus after the device server has called Send Command Complete () for that I_T_L_Q nexus (e.g., a RESPONSE frame for that I_T_L_Q nexus has been transmitted) or called Task Management Function Executed for a task management function that terminates that task (e.g., an ABORT TASK).

Table 157 shows how the arguments to the Send Data-In transport protocol service are used.

Table 157 — Send Data-In transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to which the read DATA frame is to be sent; b) T specifies the target port to send the read DATA frame; c) L specifies the LOGICAL UNIT NUMBER field in the read DATA frame header; and d) Q specifies the INITIATOR PORT TRANSFER TAG field in the read DATA frame header.
Device Server Buffer	Internal to the device server.
Application Client Buffer Offset	Specifies the DATA OFFSET field in the read DATA frame.
Request Byte Count	Specifies the size of the read DATA frame.

9.2.1.7 Data-In Delivered transport protocol service

An SSP target port uses the Data-In Delivered transport protocol service indication to notify a device server of the results of transmitting a read DATA frame.

Data-In Delivered (IN (I_T_L_Q Nexus, Delivery Result))

Table 158 shows how the arguments to the Data-In Delivered transport protocol service are determined.

Table 158 — Data-In Delivered transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port that sent the read DATA frame; b) T indicates the target port that received the read DATA frame; c) L indicates the value of the LOGICAL UNIT NUMBER field in the read DATA frame header; and d) Q indicates the value of the INITIATOR PORT TRANSFER TAG field in the read DATA frame header.
Delivery Result	From the response to the outgoing read DATA frame: a) DELIVERY SUCCESSFUL: The read DATA frame received an ACK; or b) DELIVERY FAILURE: The read DATA frame received a NAK or no response.

9.2.1.8 Receive Data-Out transport protocol service

A device server uses the Receive Data-Out transport protocol service request to request that an SSP target port transmit an XFER_RDY frame.

Receive Data-Out (IN (I_T_L_Q Nexus, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))

A device server shall only call Receive Data-Out () during a write command or bidirectional command.

A device server shall not call Receive Data-Out () for a given I_T_L_Q nexus until the Data-Out Received () has completed without error for the previous Receive Data-Out () call for that I_T_L_Q nexus (i.e., no XFER_RDY frame shall be transmitted until all write DATA frames for the previous XFER_RDY frame, if any, have been received, and the link layer has provided acknowledgement for all of the previous write DATA frames for that I_T_L_Q nexus).

A device server shall not call Receive Data-Out () for a given I_T_L_Q nexus after a Send Command Complete () has been called for that I_T_L_Q nexus or after a Task Management Function Executed () has been called for a task management function that terminates that command (e.g., an ABORT TASK).

Table 159 shows how the arguments to the Receive Data-Out transport protocol service are used.

Table 159 — Receive Data-Out transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to which the XFER_RDY frame is to be sent; b) T specifies the target port to send the XFER_RDY frame; c) L specifies the LOGICAL UNIT NUMBER field in the XFER_RDY frame header; and d) Q specifies the INITIATOR PORT TRANSFER TAG field in the XFER_RDY frame header.
Application Client Buffer Offset	Specifies the REQUESTED OFFSET field in the XFER_RDY frame.
Request Byte Count	Specifies WRITE DATA LENGTH field in the XFER_RDY frame.
Device Server Buffer	Internal to the device server.

9.2.1.9 Data-Out Received transport protocol service

An SSP target port uses the Data-Out Received transport protocol service indication to notify a device server of the result of transmitting an XFER_RDY frame (e.g., receiving write DATA frames in response).

Data-Out Received (IN (I_T_L_Q Nexus, Delivery Result))

Table 160 shows how the arguments to the Data-Out Received transport protocol service are determined.

Table 160 — Data-Out Received transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port to which the XFER_RDY frame was sent; b) T indicates the target port that sent the XFER_RDY frame; c) L indicates the value of the LOGICAL UNIT NUMBER field in the XFER_RDY frame header; and d) Q indicates the value of the INITIATOR PORT TRANSFER TAG field in the XFER_RDY frame header.
Delivery Result	From the response to the XFER_RDY: a) DELIVERY SUCCESSFUL: The XFER_RDY frame was transmitted without error and all the write DATA frames for the requested write data were received; or b) DELIVERY FAILURE: The XFER_RDY frame received a NAK or no response.

9.2.1.10 Terminate Data Transfer transport protocol service

A device server uses the Terminate Data Transfer transport protocol service request to request that an SSP target port terminate any Send Data-In () or Receive Data-Out () transport protocol services, if any, being processed using the specified nexus.

Terminate Data Transfer (IN (Nexus))

Table 161 shows how the arguments to the Terminate Data Transfer transport protocol service are used.

Table 161 — Terminate Data Transfer transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus, specifying the scope of the data transfer(s) to terminate.

9.2.1.11 Data Transfer Terminated transport protocol service

An SSP target port uses the Data Transfer Terminated transport protocol service indication to notify a device server that all data transfers for the indicated nexus have been terminated.

Data Transfer Terminated (IN (Nexus))

Table 162 shows how the arguments to the Data Transfer Terminated transport protocol service are determined.

Table 162 — Data Transfer Terminated transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus indicated by the preceding Terminate Data Transfer () call.

9.2.1.12 Send Task Management Request transport protocol service

An application client uses the Send Task Management Request transport protocol service request to request that an SSP initiator port transmit a TASK frame.

Send Task Management Request (IN (Nexus, Function Identifier, Task Management Tag, [Request Fence]))

Table 163 shows how the arguments to the Send Task Management Request transport protocol service are used.

Table 163 — Send Task Management Request transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier), where: a) I specifies the initiator port to send the TASK frame; b) T specifies the target port to which the TASK frame is sent; c) L (for an I_T_L nexus or an I_T_L_Q nexus) specifies the LOGICAL UNIT NUMBER field in the TASK frame header; and d) Q (for an I_T_L_Q nexus) specifies the INITIATOR PORT TRANSFER TAG TO MANAGE field in the TASK frame header.
Function Identifier	Specifies the TASK MANAGEMENT FUNCTION field in the TASK frame. Only these task management functions are supported: a) ABORT TASK (Nexus argument specifies an I_T_L_Q Nexus); b) ABORT TASK SET (Nexus argument specifies an I_T_L Nexus); c) CLEAR ACA (Nexus argument specifies an I_T_L Nexus); d) CLEAR TASK SET (Nexus argument specifies an I_T_L Nexus); e) I_T NEXUS RESET (Nexus argument specifies an I_T Nexus); f) LOGICAL UNIT RESET (Nexus argument specifies an I_T_L Nexus); g) QUERY TASK (Nexus argument specifies an I_T_L_Q Nexus); h) QUERY TASK SET (Nexus argument specifies an I_T_L Nexus); and i) QUERY ASYNCHRONOUS EVENT (Nexus argument specifies an I_T_L Nexus).
Task Management Tag	Specifies the INITIATOR PORT TRANSFER TAG field in the TASK frame header.
[Request Fence]	If included, specifies an I_T nexus, I_T_L nexus, or I_T_L_Q nexus for which the TASK frame is fenced.

An application client shall set the Request Fence argument to the Nexus argument.

9.2.1.13 Task Management Request Received transport protocol service

An SSP target port uses the Task Management Request Received transport protocol service indication to notify a task manager that it has received a TASK frame.

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

Table 164 shows how the arguments to the Task Management Request Received transport protocol service are determined.

Table 164 — Task Management Request Received transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier), where: a) I indicates the initiator port that sent the TASK frame; b) T indicates the target port that received the TASK frame; c) L (for an I_T_L nexus or an I_T_L_Q nexus) indicates the LOGICAL UNIT NUMBER field in the TASK frame header; and d) Q (for an I_T_L_Q nexus) indicates the INITIATOR PORT TRANSFER TAG TO MANAGE field in the TASK frame header.
Function Identifier	Indicates the TASK MANAGEMENT FUNCTION field in the TASK frame. Only these task management functions are supported: a) ABORT TASK (Nexus argument indicates an I_T_L_Q Nexus); b) ABORT TASK SET (Nexus argument indicates an I_T_L Nexus); c) CLEAR ACA (Nexus argument indicates an I_T_L Nexus); d) CLEAR TASK SET (Nexus argument indicates an I_T_L Nexus); e) I_T NEXUS RESET (Nexus argument indicates an I_T Nexus); f) LOGICAL UNIT RESET (Nexus argument indicates an I_T_L Nexus); g) QUERY TASK (Nexus argument indicates an I_T_L_Q Nexus); h) QUERY TASK SET (Nexus argument indicates an I_T_L Nexus); and i) QUERY ASYNCHRONOUS EVENT (Nexus argument indicates an I_T_L Nexus).
Task Management Tag	Indicates the INITIATOR PORT TRANSFER TAG field in the TASK frame header.

9.2.1.14 Task Management Function Executed transport protocol service

A task manager uses the Task Management Function Executed transport protocol service response to request that an SSP target port transmit a RESPONSE frame.

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

A task manager shall only call Task Management Function Executed () after receiving Task Management Request Received ().

Table 165 shows how the arguments to the Task Management Function Executed transport protocol service are used.

Table 165 — Task Management Function Executed transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T_L nexus or I_T_L_Q nexus (depending on the function), where: a) I specifies the initiator port to which the RESPONSE frame is sent; b) T specifies the target port to send the RESPONSE frame; c) L (for an I_T_L nexus or an I_T_L_Q nexus) specifies the logical unit that is sending the response frame; and d) Q (for an I_T_L_Q nexus) specifies the command that was managed by the task management function.
Service Response	Specifies the DATAPRES field and RESPONSE CODE field in the RESPONSE frame: a) FUNCTION COMPLETE: The RESPONSE frame DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION COMPLETE; b) FUNCTION SUCCEEDED: The RESPONSE frame DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION SUCCEEDED; c) FUNCTION REJECTED: The DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED; d) INCORRECT LOGICAL UNIT NUMBER: The DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to INCORRECT LOGICAL UNIT NUMBER; or e) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE frame DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to: A) INVALID FRAME; B) TASK MANAGEMENT FUNCTION FAILED; or C) OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED.
[Additional Response Information]	Specifies the ADDITIONAL RESPONSE INFORMATION field in the RESPONSE frame.
Task Management Tag	Specifies the INITIATOR PORT TRANSFER TAG field in the RESPONSE frame header.
[Response Fence]	If included, specifies an I_T nexus, I_T_L nexus, or I_T_L_Q nexus for which the RESPONSE frame is fenced.

A device server shall set the Response Fence argument to the Nexus argument.

9.2.1.15 Received Task Management Function Executed transport protocol service

An SSP initiator port uses the Received Task Management Function Executed transport protocol service confirmation to notify an application client that it has received a response to a TASK frame (e.g., received a RESPONSE frame or a NAK).

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

Table 166 shows how the arguments to the Received Task Management Function Executed transport protocol service are determined.

Table 166 — Received Task Management Function Executed transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the function), where: a) I indicates the initiator port that received the RESPONSE frame; b) T indicates the target port that sent the RESPONSE frame; c) L (for an I_T_L nexus or an I_T_L_Q nexus) indicates the logical unit that sent the response frame, and is indicated by the LOGICAL UNIT NUMBER field of the TASK frame with an INITIATOR PORT TRANSFER TAG field equal to the INITIATOR PORT TRANSFER TAG field in the RESPONSE frame header; and d) Q (for an I_T_L_Q nexus) indicates the command that was managed by the task management function, and is indicated by the INITIATOR PORT TRANSFER TAG field of the COMMAND frame with an INITIATOR PORT TRANSFER TAG field equal to the INITIATOR PORT TRANSFER TAG TO MANAGE field in the TASK frame with an INITIATOR PORT TRANSFER TAG field equal to the INITIATOR PORT TRANSFER TAG field in the RESPONSE frame header.
Service Response	Indicates the response to the TASK frame: a) FUNCTION COMPLETE: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION COMPLETE; b) FUNCTION SUCCEEDED: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION SUCCEEDED; c) FUNCTION REJECTED: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION NOT SUPPORTED; d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER; or e) SERVICE DELIVERY OR TARGET FAILURE: the ST_IFR state machine detects an error as described in 8.2.6.2.2.3 and 8.2.6.2.2.4 (e.g., a NAK was received for the COMMAND frame or the length of the RESPONSE frame is incorrect), or the RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to: A) INVALID FRAME; B) TASK MANAGEMENT FUNCTION FAILED; or C) OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED.
[Additional Response Information]	Indicates the ADDITIONAL RESPONSE INFORMATION field in the RESPONSE frame.
Task Management Tag	Indicates the INITIATOR PORT TRANSFER TAG field in the RESPONSE frame header or the TASK frame header.

9.2.2 Application client error handling

If an SSP initiator port calls Command Complete Received () with a Service Response of:

- a) Service Delivery or Target Failure - XFER_RDY Incorrect Write Data Length;
- b) Service Delivery or Target Failure - XFER_RDY Requested Offset Error;
- c) Service Delivery or Target Failure - XFER_RDY Not Expected;
- d) Service Delivery or Target Failure - DATA Incorrect Data Length;
- e) Service Delivery or Target Failure - DATA Too Much Read Data;
- f) Service Delivery or Target Failure - DATA Data Offset Error;
- g) Service Delivery or Target Failure - DATA Not Expected;

- h) Service Delivery or Target Failure - NAK Received; or
- i) Service Delivery or Target Failure - Connection Failed,

then the application client shall abort the command (e.g., by sending an ABORT TASK task management function).

After an application client calls Send SCSI Command (), if Command Complete Received () returns a Service Response of Service Delivery or Target Failure - ACK/NAK Timeout, then the application client shall send a QUERY TASK task management function with Send Task Management Request () to determine whether the command was received with no error. If Received Task Management Function Executed () returns a Service Response of FUNCTION SUCCEEDED, then the application client shall assume the command was delivered with no error. If Received Task Management Function Executed () returns a Service Response of FUNCTION COMPLETE, and Command Complete Received () has not yet been invoked a second time for the command in question (e.g., indicating a RESPONSE frame arrived for the command before the QUERY TASK was processed), then the application client shall assume the command was not delivered and may reuse the initiator port transfer tag. The application client should call Send SCSI Command () again with identical arguments.

After a Received Task Management Function Executed () call with a Service Response of Service Delivery or Target Failure - ACK/NAK Timeout, an application client should call Send Task Management Request () with identical arguments, including the same initiator port transfer tag.

After a Command Complete Received () or Received Task Management Function Executed () call returns a Service Response other than Service Delivery or Target Failure - ACK/NAK Timeout, an application client shall not reuse the initiator port transfer tag until it determines the initiator port transfer tag is no longer in use by the logical unit (e.g., the ACK for the RESPONSE frame was seen by the SSP target port). Examples of ways the application client may determine that an initiator port transfer tag may be used are:

- a) receiving another frame in the same connection;
- b) receiving a DONE (NORMAL) or DONE (CREDIT TIMEOUT) in the same connection; or
- c) receiving a DONE (ACK/NAK TIMEOUT) in the same connection, then running a QUERY TASK task management function to confirm that the initiator port transfer tag is no longer active in the logical unit.

9.2.3 Device server error handling

If an SSP target port calls Data-Out Received () with a Delivery Result set to a value in table 167, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set as indicated in table 167.

Table 167 — Delivery Result to additional sense code mapping

Delivery Result	Additional sense code
DELIVERY FAILURE - DATA OFFSET ERROR	DATA OFFSET ERROR
DELIVERY FAILURE - TOO MUCH WRITE DATA	TOO MUCH WRITE DATA
DELIVERY FAILURE - INFORMATION UNIT TOO SHORT	INFORMATION UNIT TOO SHORT
DELIVERY FAILURE - ACK/NAK TIMEOUT	ACK/NAK TIMEOUT
DELIVERY FAILURE - CONNECTION FAILED	Should be CONNECTION LOST May be ACK/NAK TIMEOUT
DELIVERY FAILURE - NAK RECEIVED	NAK RECEIVED
DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT	INITIATOR RESPONSE TIMEOUT

9.2.4 Task router and task manager error handling

If a SCSI target device performs initiator port transfer tag checking and an SSP target port calls SCSI Command Received () with an initiator port transfer tag already in use by another command (i.e., an overlapped command (see SAM-5)) in any logical unit, then the task router and task manager(s) shall:

- a) abort all task management functions received on that I_T nexus; and
- b) respond to the overlapped command as defined in SAM-5.

If a SCSI target device performs initiator port transfer tag checking and:

- a) an SSP target port calls SCSI Command Received () with an initiator port transfer tag already in use by a task management function in any logical unit; or
- b) an SSP target port calls Task Management Request Received () with an initiator port transfer tag already in use by a command or task management function in any logical unit,

then the task router and task manager(s) shall:

- a) abort all commands received on that I_T nexus;
- b) abort all task management functions received on that I_T nexus; and
- c) call Task Management Function Executed () with the Service Response set to FUNCTION REJECTED - Overlapped Initiator Port Transfer Tag Attempted (i.e., requesting that the target port set the DATAPRES field to RESPONSE_DATA and the RESPONSE_CODE field set to OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED).

9.2.5 SCSI transport protocol event notifications

Table 168 lists the SCSI transport protocol event notifications supported by this standard.

Table 168 — SCSI transport protocol events

Event notification	SAS SSP implementation
Transport Reset	Receipt of a hard reset sequence (see 4.4.2)
Nexus Loss	Receipt of specific OPEN_REJECTs for a specific time period (see 4.5).
Power Loss Expected	Receipt of a NOTIFY (POWER LOSS EXPECTED) (see 6.2.5.3.3)

9.2.6 SCSI commands

9.2.6.1 INQUIRY command

SAS-specific vital product data accessed with the INQUIRY command (see SPC-4) is described in 9.2.11.

9.2.6.2 MODE SELECT and MODE SENSE commands

SAS-specific mode pages accessed with the MODE SELECT command and MODE SENSE command (see SPC-4) are described in 9.2.7.

9.2.6.3 LOG SELECT and LOG SENSE commands

SAS-specific log pages accessed with the LOG SELECT command and LOG SENSE command (see SPC-4) are described in 9.2.8.

9.2.6.4 SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands

SAS-specific diagnostic pages accessed with the SEND DIAGNOSTIC command and RECEIVE DIAGNOSTIC RESULTS command (see SPC-4) are described in 9.2.9.

Zoning (see 4.9) is applied to SES-3 diagnostic pages as described in 9.2.9.

9.2.6.5 START STOP UNIT command

The power condition states controlled by the START STOP UNIT command (see SBC-3) for a SAS device are described in 9.2.10.

9.2.7 SCSI mode parameters

9.2.7.1 SCSI mode parameters overview

Table 169 defines mode pages supported by logical units in SCSI target devices in SAS domains (i.e., with SSP target ports) that support the MODE SELECT command or MODE SENSE command.

Table 169 — SSP target port mode pages

Mode page code	Subpage code	Description	Reference
02h	00h	Disconnect-Reconnect mode page	9.2.7.2
18h	00h	Protocol-Specific Logical Unit mode page	9.2.7.3
	01h to DFh	Reserved	
	E0h to FEh	Vendor specific	
	FFh	Return all subpages for this mode page code	SPC-4
19h	00h	Protocol-Specific Port mode page	9.2.7.4
	01h	Phy Control And Discover mode page	9.2.7.5
	02h	Shared Port Control mode page	9.2.7.6
	03h	Enhanced Phy Control mode page	9.2.7.7
	04h to DFh	Reserved	
	E0h to FEh	Vendor specific	
	FFh	Return all subpages for this mode page code	SPC-4

If any field in an implemented mode page is not implemented, then the value of the field shall be assumed to be zero (i.e., as if the field is set to zero) (see SPC-4).

If a mode page defined by this standard is not implemented, then the value of each field in that mode page that is:

- a) allowed by this standard to be changeable (e.g., not defined as a read only field); and
- b) not used solely to define the mode page structure (e.g., the NUMBER OF PHYS field in the Phy Control And Discover mode page) or coordinate access to the mode page (e.g., the GENERATION CODE field in the Phy Control And Discover mode page),

shall be assumed to be zero (i.e., as if the mode page is implemented and the field is set to zero).

9.2.7.2 Disconnect-Reconnect mode page

9.2.7.2.1 Disconnect-Reconnect mode page overview

The Disconnect-Reconnect mode page (see SPC-4) provides the application client the means to tune the performance of a service delivery subsystem. Table 170 defines the parameters which are applicable to SSP.

The application client sends the values in the fields to be used by the device server to control the SSP connections by means of a MODE SELECT command. The device server shall then communicate the field values to the SSP target port. The field values are communicated from the device server to the SSP target port in a vendor specific manner.

SAS devices shall only use the parameter fields defined in table 170. If any other fields within the Disconnect-Reconnect mode page of the MODE SELECT command contain a non-zero value, then the device server shall terminate the MODE SELECT command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 170 — Disconnect-Reconnect mode page for SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (02h)					
1	PAGE LENGTH (0Eh)							
2	Reserved							
3	Reserved							
4	(MSB)	BUS INACTIVITY TIME LIMIT						
5								(LSB)
6	Reserved							
7								
8	(MSB)	MAXIMUM CONNECT TIME LIMIT						
9								(LSB)
10	(MSB)	MAXIMUM BURST SIZE						
11								(LSB)
12	Reserved							
13								
14	(MSB)	FIRST BURST SIZE						
15								(LSB)

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 170.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 170.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 170.

The BUS INACTIVITY TIME LIMIT field is defined in SPC-4 and 9.2.7.2.2.

The MAXIMUM CONNECT TIME LIMIT field is defined in SPC-4 and 9.2.7.2.3.

The MAXIMUM BURST SIZE field is defined in SPC-4 and 9.2.7.2.4.

The FIRST BURST SIZE field is defined in SPC-4 and 9.2.7.2.5.

9.2.7.2.2 BUS INACTIVITY TIME LIMIT field

The value in the BUS INACTIVITY TIME LIMIT field contains the maximum time in 100 µs increments that an SSP target port is permitted to maintain a connection (see 4.1.12) without transferring a frame to the SSP initiator port. When this time is exceeded, the SSP target port shall prepare to close the connection (i.e., by requesting to have the link layer transmit DONE). This value may be rounded as defined in SPC-4. A value of 0000h in

this field specifies that there is no bus inactivity time limit. The bus inactivity time limit is enforced by the port layer (see 7.2.3).

9.2.7.2.3 MAXIMUM CONNECT TIME LIMIT field

The value in the MAXIMUM CONNECT TIME LIMIT field contains the maximum duration of a connection (see 4.1.12) in 100 μ s increments (e.g., a value of 0001h in this field means that the time is less than or equal to 100 μ s and a value of 0002h in this field means that the time is less than or equal to 200 μ s). When this time is exceeded, the SSP target port shall prepare to close the connection. If an SSP target port is transferring a frame when the maximum connection time limit is exceeded, then the SSP target port shall complete transfer of the frame before preparing to close the connection. This value may be rounded as defined in SPC-4. A value of 0000h in this field specifies that there is no maximum connection time limit. The maximum connection time limit is enforced by the port layer (see 7.2.3).

9.2.7.2.4 MAXIMUM BURST SIZE field

For read data, the value in the MAXIMUM BURST SIZE field contains the maximum amount of data in 512-byte increments that is transferred during a connection by an SSP target port per I_T_L_Q nexus without transferring at least one frame for a different I_T_L_Q nexus.

If the SSP target port:

- a) has read data to transfer for only one I_T_L_Q nexus; and
- b) has no requests to transfer write data for any I_T_L_Q nexus;

then the SSP target port shall prepare to close the connection after the amount of data specified by the MAXIMUM BURST SIZE field is transferred to the SSP initiator port.

For write data, the value shall specify the maximum amount of data that an SSP target port requests via a single XFER_RDY frame (see 8.2.2.3).

This value is specified in 512-byte increments (e.g., a value of 0001h in this field means that the number of bytes transferred to the SSP initiator port for the nexus is less than or equal to 512, and a value of 0002h in this field means that the number of bytes transferred to the SSP initiator port for the nexus is less than or equal to 1 024). A value of 0000h in this field specifies that there is no maximum burst size.

In terms of the SCSI transport protocol services (see 9.2.1), the device server shall limit the Request Byte Count argument to the Receive Data-Out () protocol service and the Send Data-In () protocol service to the amount specified in this field.

9.2.7.2.5 FIRST BURST SIZE field

If the ENABLE FIRST BURST bit in the COMMAND frame is set to zero, then the FIRST BURST SIZE field is ignored.

If the ENABLE FIRST BURST bit in the COMMAND frame is set to one, then the value in the FIRST BURST SIZE field contains the maximum amount of write data in 512-byte increments that may be sent by the SSP initiator port to the SSP target port without having to receive an XFER_RDY frame (see 8.2.2.3) from the SSP target port (e.g., a value of 0001h in this field means that the number of bytes transferred by the SSP initiator port is less than or equal to 512 and a value of 0002h in this field means that the number of bytes transferred by the SSP initiator port is less than or equal to 1 024).

Specifying a non-zero value in the FIRST BURST SIZE field is equivalent to an implicit XFER_RDY frame for each command requiring write data where the WRITE DATA LENGTH field of the XFER_RDY frame is set to 512 times the value of the FIRST BURST SIZE field.

The rules for data transferred using the value in the FIRST BURST SIZE field are the same as those used for data transferred for an XFER_RDY frame (i.e., the number of bytes transferred using the value in the FIRST BURST SIZE field is as if that number of bytes was requested by an XFER_RDY frame).

If the amount of data to be transferred for the command is less than the amount of data specified by the FIRST BURST SIZE field, then the SSP target port shall not transmit an XFER_RDY frame for the command. If the amount of data to be transferred for the command is greater than the amount of data specified by the FIRST BURST SIZE field, then the SSP target port shall transmit an XFER_RDY frame after it has received all of the

data specified by the FIRST BURST SIZE field from the SSP initiator port. All data for the command is not required to be transferred during the same connection in which the command is transferred.

A value of 0000h in this field specifies that there is no first burst size (i.e., an SSP initiator port transmits no write DATA frames to the SSP target port before receiving an XFER_RDY frame).

The first burst size is handled by the SCSI transport protocol services (see 9.2.1) and the SSP transport layer (see 8.2.6).

9.2.7.3 Protocol-Specific Logical Unit mode page

The Protocol-Specific Logical Unit mode page (see SPC-4) contains parameters that affect SSP target port operation on behalf of the logical unit.

The mode page policy (see SPC-4) for this mode page shall be either shared or per target port. If the SAS target device has multiple SSP target ports, then the mode page policy should be per target port.

Parameters in this mode page:

- a) shall affect all phys in the SSP target port if the mode page policy is per target port; or
- b) shall affect all SSP target ports in the SAS target device if the mode page policy is shared.

Table 171 defines the format of the page for SAS SSP.

Table 171 — Protocol-Specific Logical Unit mode page for SAS SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (18h)					
1	PAGE LENGTH (06h)							
2	Reserved			TRANSPORT LAYER RETRIES	PROTOCOL IDENTIFIER (6h)			
3	Reserved							
7								

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 171.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 171.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 171.

A TRANSPORT LAYER RETRIES bit set to one specifies that, for commands received in COMMAND frames with the TLR CONTROL field set to 00b or 11b (see 8.2.1), the target port shall support transport layer retries for XFER_RDY and DATA frames for the logical unit as described in 8.2.4 (i.e., transport layer retries are enabled). A TRANSPORT LAYER RETRIES bit set to zero specifies that, for commands received in COMMAND frames with the TLR CONTROL field set to 00b or 11b (see 8.2.1), transport layer retries shall not be used (i.e., transport layer retries are disabled).

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 171 indicating that this is a SAS SSP specific mode page.

9.2.7.4 Protocol-Specific Port mode page

The Protocol-Specific Port mode page (see SPC-4) contains parameters that affect SSP target port operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT or MODE SENSE commands.

The mode page policy (see SPC-4) for this mode page shall be either shared or per target port. If a SAS target device has multiple SSP target ports, then the mode page policy should be per target port.

Parameters in this mode page:

- a) shall affect all phys in the SSP target port if the mode page policy is per target port; or
- b) shall affect all phys in all SSP target ports in the SAS target device if the mode page policy is shared.

Table 172 defines the format of the page for SAS SSP.

Table 172 — Protocol-Specific Port mode page for SAS SSP

Byte\Bit	7	6	5	4	3	2	1	0						
0	PS	SPF (0b)	PAGE CODE (19h)											
1	PAGE LENGTH (0Eh)													
2	Reserved	CONTINUE AWT	BROADCAST ASYNCHRONOUS EVENT	READY LED MEANING	PROTOCOL IDENTIFIER (6h)									
3	Reserved													
4	(MSB)	I_T NEXUS LOSS TIME												
5								(LSB)						
6	(MSB)	INITIATOR RESPONSE TIMEOUT												
7								(LSB)						
8	(MSB)	REJECT TO OPEN LIMIT												
9								(LSB)						
10	Reserved													
15														

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 172.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 172.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 172.

A CONTINUE AWT bit set to one specifies that the SAS port shall not stop the Arbitration Wait Time timer and shall not set the Arbitration Wait Time timer to zero when the SAS port receives an OPEN_REJECT (RETRY). A CONTINUE AWT bit set to zero specifies that the SAS port shall stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when it receives an OPEN_REJECT (RETRY).

A BROADCAST ASYNCHRONOUS EVENT bit set to one specifies that the device server shall enable origination of Broadcast (Asynchronous Event) (see 4.1.13). A BROADCAST ASYNCHRONOUS EVENT bit set to zero specifies that the device server shall disable origination of Broadcast (Asynchronous Event).

The READY LED MEANING bit specifies the READY LED signal behavior (see 9.4.1). Regardless of the mode page policy (see SPC-4) for this mode page, the shared mode page policy shall be applied to the READY LED MEANING bit.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 172 indicating that this is a SAS SSP specific mode page.

The I_T NEXUS LOSS TIME field contains the minimum time that the SSP target port shall retry connection requests to an SSP initiator port that are rejected with certain responses indicating that the SSP initiator port

may no longer be present (see 7.2.2) before recognizing an I_T nexus loss (see 4.5). Table 173 defines the values of the I_T NEXUS LOSS TIME field. This value is enforced by the port layer (see 7.2.2).

Table 173 — I_T NEXUS LOSS TIME field

Code	Description
0000h	Vendor specific amount of time.
0001h to FFFEh	Time in milliseconds.
FFFFh	The SSP target port shall never recognize an I_T nexus loss (i.e., it shall retry the connection requests forever).

NOTE 82 If this mode page is implemented, then the default value of the I_T NEXUS LOSS TIME field should be non-zero. It is recommended that this value be 2 000 ms.

NOTE 83 An SSP initiator port should retry connection requests for at least the time indicated by the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page for the SSP target port to which it is trying to establish a connection (see 4.5).

The INITIATOR RESPONSE TIMEOUT field contains the minimum time in milliseconds that the SSP target port shall wait for the receipt of a frame (e.g., a write DATA frame) before aborting the command associated with that frame. An INITIATOR RESPONSE TIMEOUT field set to 0000h indicates that the SSP target port shall disable the initiator response timeout timer. This value is enforced by the transport layer (see 8.2.6.3).

The REJECT TO OPEN LIMIT field contains the minimum time in 10 μ s increments that the target port shall wait to establish a connection request with an initiator port on an I_T nexus after receiving an OPEN_REJECT (RETRY), OPEN_REJECT (RESERVED CONTINUE 0), or OPEN_REJECT (RESERVED CONTINUE 1). This value may be rounded as defined in SPC-4. A REJECT TO OPEN LIMIT field set to 0000h indicates that the minimum time is vendor specific. This minimum time is enforced by the port layer (see 7.2.3).

9.2.7.5 Phy Control And Discover mode page

The Phy Control And Discover mode page contains parameters that affect SSP target phy operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT command or MODE SENSE command.

The mode page policy (see SPC-4) for this mode page shall be shared. Parameters in this mode page shall affect only the referenced phy.

Table 174 defines the format of this mode page.

Table 174 — Phy Control And Discover mode page

Byte\Bit	7	6	5	4	3	2	1	0	
0	PS	SPF (1b)	PAGE CODE (19h)						
1	SUBPAGE CODE (01h)								
2	(MSB)	PAGE LENGTH (n - 3)						(LSB)	
3									
4	Reserved								
5	Reserved				PROTOCOL IDENTIFIER (6h)				
6	GENERATION CODE								
7	NUMBER OF PHYS								
SAS phy mode descriptor list									
8	SAS phy mode descriptor (first) (see table 175)								
55									
...									
n - 47	SAS phy mode descriptor (last) (see table 175)								
n									

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 174.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 174.

The SUBPAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 174.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 174 (i.e., $4 + ((\text{the value of the NUMBER OF PHYS field}) \times (\text{the length in bytes of the SAS phy mode descriptor}))$).

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 174 indicating that this is a SAS SSP specific mode page.

The GENERATION CODE field is a one-byte counter that shall be incremented by one by the device server every time the values in this mode page or the Enhanced Phy Control mode page (see 9.2.7.7) are changed. A GENERATION CODE field set to 00h indicates the generation code is unknown. The device server shall wrap this field to 01h as the next increment after reaching its maximum value (i.e., FFh). The GENERATION CODE field is also contained in the Enhanced Phy Control mode page (see clause 9.2.7.7) and the Protocol-Specific Port log page (see clause 9.2.8.1) and may be used to correlate phy settings across mode page and log page accesses.

NOTE 84 Device servers compliant with SAS-1.1 set the GENERATION CODE field to 00h.

The NUMBER OF PHYS field contains the number of phys in the SAS target device and indicates the number of SAS phy mode descriptors in the SAS phy mode descriptor list. This field shall not be changeable with the MODE SELECT command.

The SAS phy mode descriptor list contains a SAS phy mode descriptor for each phy in the SAS target device, not just the SAS target port, starting with the lowest numbered phy and ending with the highest numbered phy as determined by the value in the PHY IDENTIFIER field in the SAS phy mode descriptor.

Table 175 defines the SAS phy mode descriptor.

Table 175 — SAS phy mode descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	Reserved							
3								
4	Reserved	ATTACHED DEVICE TYPE			ATTACHED REASON			
5	REASON				NEGOTIATED LOGICAL LINK RATE			
6	Reserved				ATTACHED SSP INITIATOR PORT	ATTACHED STP INITIATOR PORT	ATTACHED SMP INITIATOR PORT	Reserved
7	Reserved				ATTACHED SSP TARGET PORT	ATTACHED STP TARGET PORT	ATTACHED SMP TARGET PORT	Reserved
8	SAS ADDRESS							
15								
16	ATTACHED SAS ADDRESS							
23								
24	ATTACHED PHY IDENTIFIER							
25	Reserved							
31								
32	PROGRAMMED MINIMUM PHYSICAL LINK RATE				HARDWARE MINIMUM PHYSICAL LINK RATE			
33	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				HARDWARE MAXIMUM PHYSICAL LINK RATE			
34	Reserved							
41								
42	Vendor specific							
43								
44	Reserved							
47								

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field are defined in the SMP PHY CONTROL function (see 9.4.3.28) for accesses with MODE SELECT commands and in the SMP DISCOVER function (see 9.4.3.10) for accesses with MODE SENSE commands.

The fields in the SAS phy mode descriptor not defined in this subclause are defined in the SMP DISCOVER response (see 9.4.3.10). These fields shall not be changeable with the MODE SELECT command.

9.2.7.6 Shared Port Control mode page

The Shared Port Control mode page contains parameters that affect SSP target port operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT command or MODE SENSE command.

The mode page policy (see SPC-4) for this mode page shall be shared.

Table 176 defines the format of this mode page.

Table 176 — Shared Port Control mode page

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (19h)					
1	SUBPAGE CODE (02h)							
2	(MSB)	PAGE LENGTH (000Ch)						
3								
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	(MSB)	POWER LOSS TIMEOUT						
7								
8		Reserved						
15								

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 176.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 176.

The SUBPAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 176.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 176.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 176 indicating that this is a SAS SSP specific mode page.

The POWER LOSS TIMEOUT field contains the maximum time, in one millisecond increments, that a target port shall respond to connection requests with OPEN_REJECT (RETRY) after receiving NOTIFY (POWER LOSS EXPECTED) (see 6.2.5.3.3). A POWER LOSS TIMEOUT field set to 0000h specifies that the maximum time is vendor specific. The power loss timeout shall be restarted on each NOTIFY (POWER LOSS EXPECTED) that is received.

9.2.7.7 Enhanced Phy Control mode page

The Enhanced Phy Control mode page contains parameters that affect SSP target phy operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT command or MODE SENSE command.

The mode page policy (see SPC-4) for this mode page shall be shared.

Table 177 defines the format of this mode page.

Table 177 — Enhanced Phy Control mode page

Byte\Bit	7	6	5	4	3	2	1	0	
0	PS	SPF (1b)	PAGE CODE (19h)						
1	SUBPAGE CODE (03h)								
2	(MSB)	PAGE LENGTH (n - 3)						(LSB)	
3									
4	Reserved								
5	Reserved				PROTOCOL IDENTIFIER (6h)				
6	GENERATION CODE								
7	NUMBER OF PHYS								
Enhanced phy control mode descriptor list									
8	Enhanced phy control mode descriptor (first) (see table 178)								
27									
...									
n - 19	Enhanced phy control mode descriptor (last) (see table 178)								
n									

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 177.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 177.

The SUBPAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 177.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 177 (i.e., 4 + (the value of the NUMBER OF PHYS field) × (the length in bytes of the SAS phy mode descriptor)).

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 177 indicating that this is a SAS SSP specific mode page.

The GENERATION CODE field is defined in the Phy Control and Discover mode page (see 9.2.7.5).

The NUMBER OF PHYS field contains the number of phys in the SAS target device and indicates the number of enhanced phy control mode descriptors in the enhanced phy control mode descriptor list. This field shall not be changeable with the MODE SELECT command.

The enhanced phy control mode descriptor list contains an enhanced phy control mode descriptor for each phy in the SAS target device, not just the SAS target port, starting with the lowest numbered phy and ending with the highest numbered phy as determined by the value in the PHY IDENTIFIER field in the enhanced phy control mode descriptor.

Table 178 defines the enhanced phy control mode descriptor.

Table 178 — Enhanced phy control mode descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	(MSB)							
3	DESCRIPTOR LENGTH (0010h)							
4	(LSB)							
7	PROGRAMMED PHY CAPABILITIES							
8	CURRENT PHY CAPABILITIES							
11	ATTACHED PHY CAPABILITIES							
12	Reserved							
15	Reserved							
16	Reserved							
17	Reserved							
18	Reserved		OPTICAL MODE ENABLED	NEGOTIATED SSC	NEGOTIATED PHYSICAL LINK RATE			
19	Reserved					ENABLE SLUMBER	ENABLE PARTIAL	HARDWARE MUXING SUPPORTED

The DESCRIPTOR LENGTH field contains the length in bytes that follow in the descriptor and shall be set to the value defined in table 178.

An ENABLE SLUMBER bit set to one specifies that the device server shall enable the management application layer to control the slumber phy power condition (see 4.10.1.4) on the phy specified by the PHY IDENTIFIER field. An ENABLE SLUMBER bit set to zero specifies that the device server shall disable control of the slumber phy power condition by the management application layer on the phy specified by the PHY IDENTIFIER field.

An ENABLE PARTIAL bit set to one specifies that the device server shall enable the management application layer to control the partial phy power condition (see 4.10.1.3) on the phy specified by the PHY IDENTIFIER field. An ENABLE PARTIAL bit set to zero specifies that the device server shall disable control of the partial phy power condition by the management application layer on the phy specified by the PHY IDENTIFIER field.

The fields in the enhanced phy control mode descriptor not defined in this subclause are defined in the SMP DISCOVER response (see 9.4.3.10). These fields shall not be changeable with the MODE SELECT command.

9.2.8 SCSI log parameters

9.2.8.1 Protocol-Specific Port log page

The Protocol-Specific Port log page for SAS SSP defined in table 179 provides the application client a means to determine information about phy events concerning the SAS target device's phy(s).

Table 179 — Protocol-Specific Port log page for SAS SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (18h)					
1	SUBPAGE CODE (00h)							
2	(MSB)	PAGE LENGTH (n - 3)						
3								(LSB)
Protocol-specific port log parameter list								
4	Protocol-specific port log parameter (first) (see table 180)							
...	...							
	Protocol-specific port log parameter (last) (see table 180)							
n								

The DISABLE SAVE (DS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit is defined in SPC-4 and shall be set to the value defined in table 179.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 179.

The SUBPAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 179.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 179.

The protocol-specific port log parameter list contains a protocol-specific port log parameter for each SCSI port in the SAS target device.

Table 180 defines the format for the Protocol-Specific Port log parameter for SAS. The SAS log parameter is a list parameter (i.e., not a data counter) and only has cumulative (i.e., not threshold) values (see SPC-4).

Table 180 — Protocol-Specific Port log parameter for SAS

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB) _____							
1	PARAMETER CODE (relative target port identifier) _____ (LSB)							
2	Parameter control byte							
	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
3	PARAMETER LENGTH (y - 3)							
4	Reserved				PROTOCOL IDENTIFIER (6h)			
5	Reserved							
6	GENERATION CODE							
7	NUMBER OF PHYS							
SAS phy log descriptor list								
8	_____							
8 + m	SAS phy log descriptor (first) (see table 182) _____							
...	_____							
y - m	_____							
y	SAS phy log descriptor (last) (see table 182) _____							

The PARAMETER CODE field is defined in SPC-4 and contains the relative target port identifier (see SPC-4) of the SSP target port that the log parameter describes.

Table 181 defines the values of the fields in the parameter control byte for the log parameter.

Table 181 — Parameter control byte in the Protocol-Specific Port log parameter for SAS

Field	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0	ignored	The DU bit is not defined for list parameters.
TSD	0	0 or 1	The device server shall support implicitly saving the log parameter at vendor specific intervals.
ETC	0	ignored	The ETC bit is not defined for list parameters.
TMC	00b	ignored	The TMC field is not defined for list parameters.
FORMAT AND LINKING	11b	11b	The log parameter is a binary format list parameter.

The PARAMETER LENGTH field is defined in SPC-4 and shall be set to the value defined in table 180.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 180.

The GENERATION CODE field is defined in the Phy Control and Discover mode page (see 9.2.7.5).

The NUMBER OF PHYS field contains the number of phys in the SAS target port (not in the entire SAS target device) and indicates the number of SAS phy log descriptors in the SAS phy log descriptor list.

The SAS phy log descriptor list contains SAS phy log descriptors.

Table 182 defines the SAS phy log descriptor.

Table 182 — SAS phy log descriptor (part 1 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	Reserved							
3	SAS PHY LOG DESCRIPTOR LENGTH (m - 3)							
4	Reserved	ATTACHED DEVICE TYPE			ATTACHED REASON			
5	REASON				NEGOTIATED LOGICAL LINK RATE			
6	Reserved				ATTACHED SSP INITIATOR PORT	ATTACHED STP INITIATOR PORT	ATTACHED SMP INITIATOR PORT	Reserved
7	Reserved				ATTACHED SSP TARGET PORT	ATTACHED STP TARGET PORT	ATTACHED SMP TARGET PORT	Reserved
8	SAS ADDRESS							
15								
16								
23								
24	ATTACHED SAS ADDRESS							
25	ATTACHED PHY IDENTIFIER							
31	Reserved							
32	(MSB)	INVALID DWORD COUNT						(LSB)
35								
36	(MSB)	RUNNING DISPARITY ERROR COUNT						(LSB)
39								
40	(MSB)	LOSS OF DWORD SYNCHRONIZATION						(LSB)
43								
44	(MSB)	PHY RESET PROBLEM						(LSB)
47								
48	Reserved							
49	Reserved							

Table 182 — SAS phy log descriptor (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
50	PHY EVENT DESCRIPTOR LENGTH							
51	NUMBER OF PHY EVENT DESCRIPTORS							
Phy event descriptor list								
52	Phy event descriptor (first) (see table 248 in 9.4.3.14.4)							
63								
...	...							
m - 11	Phy event descriptor (last) (see table 248 in 9.4.3.14.4)							
m								

The SAS PHY LOG DESCRIPTOR LENGTH field indicates the number of bytes that follow in the SAS phy log descriptor and shall be set to the value defined in table 182. A SAS PHY LOG DESCRIPTOR LENGTH field set to 00h indicates that there are 44 additional bytes.

NOTE 85 Logical units compliant with SAS and SAS-1.1 only support a 48 byte SAS phy log descriptor.

The INVALID DWORD COUNT field, RUNNING DISPARITY ERROR COUNT field, LOSS OF DWORD SYNCHRONIZATION field, and PHY RESET PROBLEM COUNT field are defined in the SMP REPORT PHY ERROR LOG response (see 9.4.3.11).

For the INVALID DWORD COUNT field, RUNNING DISPARITY ERROR COUNT field, LOSS OF DWORD SYNCHRONIZATION COUNT field, and PHY RESET PROBLEM COUNT field, the phy may maintain any size counter but should maintain a 32-bit counter. If it reaches its maximum value, then the counter shall stop and the device server shall set the field to FFFFFFFFh in the SAS phy log descriptor.

The PHY EVENT DESCRIPTOR LENGTH field indicates the number of bytes in the phy event descriptor (see 9.4.3.14.4).

The NUMBER OF PHY EVENT DESCRIPTORS field indicates the number of phy event descriptors in the phy event descriptor list.

Each phy event descriptor follows the format defined for the SMP REPORT PHY EVENT function in table 248 (see 9.4.3.14.4).

The fields in the SAS phy log descriptor not defined in this subclause are defined in the SMP DISCOVER response (see 9.4.3.10). These fields shall not be changeable with the LOG SELECT command.

9.2.9 SCSI diagnostic parameters

9.2.9.1 SCSI diagnostic parameters overview

Table 183 defines diagnostic pages supported by logical units in SCSI target devices in SAS domains (i.e., with SSP target ports) that support the SEND DIAGNOSTIC or RECEIVE DIAGNOSTIC RESULTS commands.

Table 183 — SSP target port diagnostic pages

Diagnostic page code	Description	Reference
3Fh	Protocol-Specific diagnostic page	9.2.9.2

An enclosure services process (see SES-3) describing elements in a SAS domain that are attached to a zoning expander device with zoning enabled (see 4.9) shall apply the zone permission table when providing access to those elements. Element types that may be subject to zoning include:

- a) Device Slot element;
- b) Array Device Slot element;
- c) Enclosure Services Controller Electronics element;
- d) SCC Controller Electronics element;
- e) SCSI Port/Transceiver element;
- f) SCSI Target Port element;
- g) SCSI Initiator Port element;
- h) SAS Expander element; and
- i) SAS Connector element.

Table 184 defines SCSI enclosure services diagnostic pages supported by logical units in SCSI target devices in SAS domains (e.g., with SSP target ports) that are affected by zoning.

Table 184 — Diagnostic pages affected by zoning

Diagnostic page code	Description	Reference
02h	Enclosure Control diagnostic page	SES-3 and 9.2.9.3
	Enclosure Status diagnostic page	SES-3 and 9.2.9.4
0Ah	Additional Element Status diagnostic page	SES-3 and 9.2.9.5

9.2.9.2 Protocol-Specific diagnostic page

The Protocol-Specific diagnostic page for SAS SSP provides a method for an application client to enable and disable phy test functions (see 4.12) for selected phys. The diagnostic page format is specified in SPC-4.

The Protocol-Specific diagnostic page is sent by an application client using the SEND DIAGNOSTIC command. If the device server receives a RECEIVE DIAGNOSTIC RESULTS command with the PAGE CODE field set to 3Fh, then it shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB Table 185 defines the Protocol-Specific diagnostic page for SAS SSP.

Table 185 — Protocol-Specific diagnostic page for SAS SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	PAGE CODE (3Fh)							
1	Reserved				PROTOCOL IDENTIFIER (6h)			
2	(MSB)							
3	PAGE LENGTH (001Ch)							
4	(LSB)							
5	PHY IDENTIFIER							
6	PHY TEST FUNCTION							
7	PHY TEST PATTERN							
8	Reserved	PHY TEST FUNCTION SATA	PHY TEST FUNCTION SSC		PHY TEST FUNCTION PHYSICAL LINK RATE			
10	Reserved							
11	PHY TEST PATTERN DWORDS CONTROL							
12	PHY TEST PATTERN DWORDS							
19	PHY TEST PATTERN DWORDS							
20	Reserved							
31	Reserved							

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 185.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 185 indicating this is a SAS SSP specific diagnostic page.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 185.

The PHY IDENTIFIER field specifies the phy identifier (see 4.2.10) of the phy that is to perform or to stop performing a phy test function (i.e., the selected phy). If the PHY IDENTIFIER field specifies a phy that does not exist, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The PHY TEST FUNCTION field specifies the phy test function to be performed and is defined in table 186. If the PHY TEST FUNCTION field specifies a phy test function that is not supported, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 186 — PHY TEST FUNCTION field

Code	Name	Description
00h	STOP	If the selected phy is performing a phy test function, then the selected phy shall stop performing the phy test function and originate a link reset sequence. If the selected phy is not performing a phy test function, then this function has no effect on the selected phy. ^a
01h	TRANSMIT PATTERN	If the selected phy is not performing a phy test function, then the selected phy shall perform the transmit pattern phy test function (see 4.12.2) using the phy test pattern specified by the PHY TEST PATTERN field and the physical link rate specified by the PHY TEST FUNCTION PHYSICAL LINK RATE field. If the selected phy is performing a phy test function, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to PHY TEST FUNCTION IN PROGRESS. ^a
02h to EFh	Reserved	
F0h to FFh	Vendor specific	
^a If there is no SSP target port available to receive a SEND DIAGNOSTIC command to stop a phy from performing a phy test function, then a power on may be required to cause the phy to stop performing the function and originate a phy reset sequence.		

If the PHY TEST FUNCTION field is set to 01h (i.e., TRANSMIT_PATTERN), then the PHY TEST PATTERN field specifies the phy test pattern to be transmitted as defined by table 187. If the PHY TEST PATTERN field specifies a phy test pattern that is not supported by the specified SAS phy, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 187 — PHY TEST PATTERN field

Code	Name	Description
00h	Reserved	
01h	JTPAT	The selected phy shall continuously transmit JTPAT for RD+ and RD- (see A.1).
02h	CJTPAT	The selected phy shall continuously transmit CJTPAT (see A.2).
03h to 0Fh	Reserved	
10h	TRAIN	The selected phy shall continuously transmit the TRAIN pattern (see 5.7.4.2.3.4).
11h	TRAIN_DONE	The selected phy shall continuously transmit the TRAIN_DONE pattern (see 5.7.4.2.3.4).
12h	IDLE	The selected phy shall continuously transmit idle dwords (see 6.4).
13h	SCRAMBLED_0	The selected phy shall continuously transmit a repeating pattern of at least 58 dwords (i.e., 2 320 bits on the physical link) set to 00000000h that are transmitted scrambled and 8b10b encoded (see 6.4). The scrambler shall be reinitialized at the beginning of each pattern. See table E.2 in E.4.
14h to 3Fh	Reserved	
40h	TWO_DWORDS	The selected phy shall continuously transmit the dwords specified by the PHY TEST PATTERN DWORDS CONTROL field and the PHY TEST PATTERN DWORDS field without scrambling. This pattern is only for use for characterization of the transmitter device and the passive interconnect. Phys are not required to support all patterns that may be specified.
41h to EFh	Reserved	
F0h to FFh	Vendor specific	

A PHY TEST FUNCTION SATA bit set to one specifies that the phy shall transmit as a SATA phy during the phy test function. A PHY TEST FUNCTION SATA bit set to zero specifies that the phy shall transmit as a SAS phy during the phy test function. If the PHY TEST FUNCTION SATA bit is set to one and the phy does not support SATA, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The PHY TEST FUNCTION SSC field specifies the SSC modulation type (see SAS-2.1) that the phy shall use during transmission during the phy test function and is defined in table 188. If the SSC modulation type specified by the PHY TEST FUNCTION SSC field is not supported (e.g., if the phy is a SAS phy, then it only supports no-spreading and down-spreading SSC), then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 188 — PHY TEST FUNCTION SSC field

Code	Description
00b	No-spreading
01b	Center-spreading SSC ^a
10b	Down-spreading SSC
11b	Reserved
^a If the PHY TEST FUNCTION SATA bit is set to one (i.e., a SATA phy is requested to transmit center-spreading), then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.	

The PHY TEST FUNCTION PHYSICAL LINK RATE field specifies the physical link rate at which the phy test function shall be performed and is defined in table 189. If the physical link rate specified by the PHY TEST FUNCTION PHYSICAL LINK RATE field is less than the hardware minimum physical link rate or greater than the hardware maximum physical link rate, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 189 — PHY TEST FUNCTION PHYSICAL LINK RATE field

Code	Description
0h to 7h	Reserved
8h	1.5 Gbps
9h	3 Gbps
Ah	6 Gbps
Bh to Fh	Reserved for future physical link rates

The PHY TEST PATTERN DWORDS CONTROL field and PHY TEST PATTERN DWORDS field are only used if the PHY TEST PATTERN field is set to 40h (i.e., TWO_DWORDS) (see table 187).

The PHY TEST PATTERN DWORDS CONTROL field defined in table 190 controls whether the bytes in the PHY TEST PATTERN DWORDS field are sent as control characters or data characters.

Table 190 — PHY TEST PATTERN DWORDS CONTROL field

Code	Description
00h	Each byte in the PHY TEST PATTERN DWORDS field shall be sent as a data character (i.e., Dxx.y) (see 5.3.6) without scrambling.
08h	The fifth byte in the PHY TEST PATTERN DWORDS field shall be sent as a control character (i.e., Kxx.y) (see 5.3.7). Each other byte shall be sent as a data character without scrambling.
80h	The first byte in the PHY TEST PATTERN DWORDS field shall be sent as a control character. Each other byte shall be sent as a data character without scrambling.
88h	The first and fifth bytes in the PHY TEST PATTERN DWORDS field shall each be sent as a control character. Each other byte shall be sent as a data character without scrambling.
All others	Reserved

The PHY TEST PATTERN DWORDS field contains the two dwords that are sent during a TWO_DWORDS test pattern. Whether each byte in the dwords is sent as a control character or a data character is specified by the PHY TEST PATTERN DWORDS CONTROL field. A byte specifying a control character shall only specify a control character which is used in this standard (see table 42 in 5.3.7) and is supported by the phy (i.e., all phys support K28.5 (i.e., BCh), but only phys supporting STP support K28.3 (i.e., 7Ch) or K28.6 (i.e., DCh)).

The device server shall terminate a SEND DIAGNOSTIC command specifying any unsupported combination with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 191 lists some examples of TWO_DWORDS phy test patterns.

Table 191 — TWO_DWORDS phy test pattern examples

PHY TEST PATTERN DWORDS CONTROL field	PHY TEST PATTERN DWORDS field	Description
00h	4A4A4A4A 4A4A4A4Ah	D10.2 characters (see table 40 in 5.3.6). This pattern contains 01b repeating and has the highest possible frequency. This pattern may be used for measuring intra-pair skew, rise time, fall time, and RJ (see SAS-2.1).
00h	B5B5B5B5 B5B5B5B5h	D21.5 characters (see table 40 in 5.3.6). This pattern contains 10b repeating and has the highest possible frequency. This pattern may be used for measuring intra-pair skew, rise time, fall time, and RJ (see SAS-2.1).
00h	78787878 78787878h	D24.3 characters (see table 40 in 5.3.6). This pattern contains 0011b or 1100b repeating (depending on starting disparity) and has half the highest possible frequency. This pattern may be used for calibrating the JTF, calibrating the reference transmitter test load, and measuring transmitter device S-parameters (see SAS-2.1).
00h	D926D926 D926D926h	Pairs of D25.6 and D6.1 characters (see table 40 in 5.3.6). This pattern contains 1001b repeating and has half the highest possible frequency.
00h	7E7E7E7E 7E7E7E7Eh	D30.3 characters (see table 40 in 5.3.6). This pattern contains four bits of one polarity, three bits of the other polarity, and three bits of the first polarity (e.g., 1111000111b), followed by the inverse (e.g., 0000111000b). This pattern may be used for measuring transmitter equalization and SSC-induced jitter (see SAS-2.1).
88h	BC4A4A7B BC4A4A7Bh	ALIGN (0) primitives (see table 64 in 6.2.3). This pattern appears during OOB bursts (see 5.6), the SATA speed negotiation sequence (see 5.7.2.2), and the SAS speed negotiation sequence (see 5.7.4.2).
88h	BC070707 BC070707h	ALIGN (1) primitives (see table 64 in 6.2.3). This pattern appears during the SAS speed negotiation sequences (see 5.7.4.2).
80h	BC4A4A7B 4A787E7Eh	Pairs of an ALIGN (0) (see table 64 in 6.2.3) and a dword containing D10.2, D24.3, D30.3, and D30.3 characters (see table 40 in 5.3.6).

9.2.9.3 Enclosure Control diagnostic page

If the SELECT bit is set to one for any element that represents a device attached to an expander phy for which the SAS initiator port performing the SEND DIAGNOSTIC command does not have access according to the zone permission table, then the enclosure services process shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

9.2.9.4 Enclosure Status diagnostic page

The enclosure services process shall set the ELEMENT STATUS CODE field to 8h (i.e., No Access Allowed) for each element that represents a device attached to an expander phy for which the SAS initiator port performing the RECEIVE DIAGNOSTIC RESULTS command does not have access according to the zone permission table.

9.2.9.5 Additional Element Status diagnostic page

The enclosure services process shall set the INVALID bit to one in the Additional Element Status descriptor (see SES-3) for each element that represents a device attached to an expander phy for which the SAS initiator port performing the RECEIVE DIAGNOSTIC RESULTS command does not have access according to the zone permission table.

9.2.10 SCSI power conditions

9.2.10.1 SCSI power conditions overview

The logical unit power condition states from the Power Condition mode page (see SPC-4) and START STOP UNIT command (see SBC-3), if implemented, shall interact with NOTIFY (ENABLE SPINUP) (see 6.2.5.3) to control temporary consumption of additional power (e.g., to spin up rotating media) as described in this subclause.

The logical unit uses NOTIFY (ENABLE SPINUP) to delay:

- a) initial temporary consumption of additional power after power on;
- b) temporary consumption of additional power requested by START STOP UNIT commands; and
- c) temporary consumption of additional power while making a change from a standby power condition to a higher power condition.

9.2.10.2 SA_PC (SCSI application layer power condition) state machine

9.2.10.2.1 SA_PC state machine overview

The SA_PC (SCSI application layer power condition) state machine describes how the SAS target device processes logical unit power condition state change requests and NOTIFY (ENABLE SPINUP) if it is a SCSI target device.

Logical units with peripheral device types other than direct-access block device that implement an additional power condition that consumes more peak power during a change from that additional power condition to a higher power condition than the typical peak power consumption in the active power condition shall use NOTIFY (ENABLE SPINUP) to delay consumption of additional power when making the change to a higher power condition.

NOTE 86 This state machine is an enhanced version of the logical unit power condition state machines described in SPC-4 and SBC-3 that adds the interactions with NOTIFY (ENABLE SPINUP) primitive.

This state machine consists of the states shown in table 192.

Table 192 — Summary of states in the SA_PC state machine

State	Reference	Modified ^a	States that contribute to definition	
			SPC-4	SBC-3
SA_PC_0:Powered_On ^b	9.2.10.2.2	No	PC0:Powered_On	SSU_PC0:Powered_On
SA_PC_1:Active	9.2.10.2.3	No	PC1:Active	SSU_PC1:Active
SA_PC_2:Idle	9.2.10.2.4	No	PC2:Idle	SSU_PC2:Idle
SA_PC_3:Standby	9.2.10.2.5	No	PC3:Standby	SSU_PC3:Standby
SA_PC_4:Active_Wait	9.2.10.2.6	Yes	PC4:Active_Wait	SSU_PC4:Active_Wait
SA_PC_5:Wait_Idle	9.2.10.2.7	No	PC5:Wait_Idle	SSU_PC5:Wait_Idle
SA_PC_6:Wait_Standby	9.2.10.2.8	No	PC6:Wait_Standby	SSU_PC6:Wait_Standby
SA_PC_7:Idle_Wait	9.2.10.2.9	Yes		SSU_PC7:Idle_Wait
SA_PC_8:Stopped	9.2.10.2.10	No		SSU_PC8:Stopped
SA_PC_9:Standby_Wait	9.2.10.2.11	No		SSU_PC9:Standby_Wait
SA_PC_10:Wait_Stopped	9.2.10.2.12	No		SSU_PC10:Wait_Stopped
^a Yes indicates that this standard adds requirements to a state. No indicates that this standard does not alter or enhance the requirements defined in SPC-4 and SBC-3.				
^b SA_PC_0:Powered_On is the initial state.				

While in the following SA_PC states the logical unit may be increasing power usage to enter a higher power condition:

- a) SA_PC_4:Active_Wait;
- b) SA_PC_7:Idle_Wait; or
- c) SA_PC_9:Standby_Wait.

While in the following SA_PC states the logical unit may be decreasing power usage to enter a lower power condition:

- a) SA_PC_5:Wait_Idle;
- b) SA_PC_6:Wait_Standby; or
- c) SA_PC_10:Wait_Stopped.

Any command causing a state machine transition (e.g., a START STOP UNIT command with the IMMED bit set to zero) shall not complete with GOOD status until this state machine reaches the state (i.e., power condition) required by the command.

This state machine shall start in the SA_PC_0:Powered_On state after power on. For direct-access block devices, the SA_PC state machine shall be configured to transition to the SA_PC_8:Stopped state or the SA_PC_4:Active_Wait state after power on by a mechanism outside the scope of this standard.

This state machine uses the following timers that are controlled by the Power Condition mode page (see SPC-4):

- a) the idle condition timers; and
- b) the standby condition timers.

If the device server processes a START STOP UNIT command with the IMMED bit set to one, then the device server shall complete the command before completing the transition, if any, specified by a START STOP UNIT command.

Figure 148 describes the SA_PC state machine.

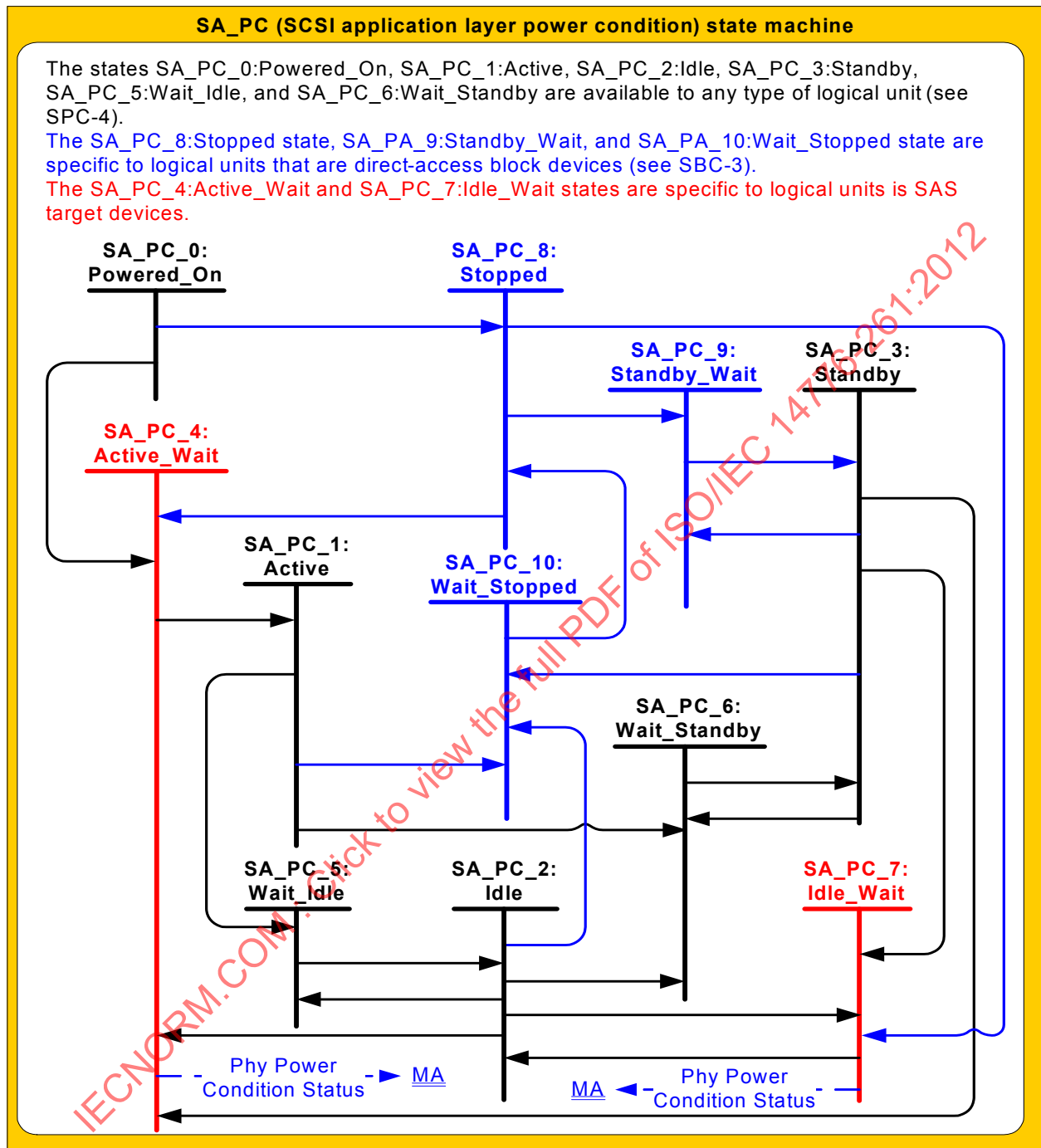


Figure 148 — SA_PC (SCSI application layer power condition) state machine for SAS

9.2.10.2.2 SA_PC_0:Powered_On state

9.2.10.2.2.1 State description

See the PC0:Power_On state in SPC-4 for details about this state.

9.2.10.2.2.2 Transition SA_PC_0:Powered_On to SA_PC_4:Active_Wait

For devices that are not direct-access block devices, see the PC0:Power_On to PC4:Active_Wait transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC0:Power_On to SSU_PC4:Active_Wait transition in SBC-3 for details about this transition.

9.2.10.2.2.3 Transition SA_PC_0:Powered_On to SA_PC_8:Stopped

For direct-access block devices, see the SSU_PC0:Power_On to SSU_PC8:Stopped transition in SBC-3 for details about this transition.

9.2.10.2.3 SA_PC_1:Active state**9.2.10.2.3.1 State description**

See the PC1:Active state in SPC-4 for details about this state.

9.2.10.2.3.2 Transition SA_PC_1:Active to SA_PC_5:Wait_Idle

For devices that are not direct-access block devices, see the PC1:Active to PC5:Wait_Idle transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC1:Active to SSU_PC5:Wait_Idle transition in SBC-3 for details about this transition.

9.2.10.2.3.3 Transition SA_PC_1:Active to SA_PC_6:Wait_Standby

For devices that are not direct-access block devices, see the PC1:Active to PC6:Wait_Standby transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC1:Active to SSU_PC6:Wait_Standby transition in SBC-3 for details about this transition.

9.2.10.2.3.4 Transition SA_PC_1:Active to SA_PC_10:Wait_Stopped

For direct-access block devices, see the SSU_PC1:Active to SSU_PC10:Wait_Stopped transition in SBC-3 for details about this transition.

9.2.10.2.4 SA_PC_2:Idle state**9.2.10.2.4.1 State description**

See the PC2:Idle state in SPC-4 for details about this state.

9.2.10.2.4.2 Transition SA_PC_2:Idle to SA_PC_4:Active_Wait

For devices that are not direct-access block devices, see the PC2:Idle to PC4:Active_Wait transition in SPC-4 for details about this transition.

For direct access block devices, see the SSU_PC2:Idle to SSU_PC4:Active_Wait transition in SBC-3 for details about this transition.

9.2.10.2.4.3 Transition SA_PC_2:Idle to SA_PC_7:Idle_Wait

For direct-access block devices, see the SSU_PC2:Idle to SSU_PC7:Idle_Wait transition in SBC-3 for details about this transition.

9.2.10.2.4.4 Transition SA_PC_2:Idle to SA_PC_5:Wait_Idle

For devices that are not direct-access block devices, see the PC2:Idle to PC5:Wait_Idle transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC2:Idle to SSU_PC5:Wait_Idle transition in SBC-3 for details about this transition.

9.2.10.2.4.5 Transition SA_PC_2:Idle to SA_PC_6:Wait_Standby

For devices that are not direct-access block device, see the PC2:Idle to PC6:Wait_Standby transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC2:Idles to SSU_PC6:Wait_Standby transition in SBC-3 for details about this transition.

9.2.10.2.4.6 Transition SA_PC_2:Idle to SA_PC_10:Wait_Stopped

- a) For direct-access block devices, see the SSU_PC2:Idle to SSU_PC10:Wait_Stopped transition in SBC-3 for details about this transition.

9.2.10.2.5 SA_PC_3:Standby state

9.2.10.2.5.1 State description

See the PC3:Standby state in SPC-4 for details about this state.

9.2.10.2.5.2 Transition SA_PC_3:Standby to SA_PC_4:Active_Wait

For devices that are not direct-access block devices, see the PC3:Standby to PC4:Active_Wait transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC3:Standby to SSU_PC4:Active_Wait transition in SBC-3 for details about this transition.

9.2.10.2.5.3 Transition SA_PC_3:Standby to SA_PC_7:Idle_Wait

For direct-access block devices, see the SSU_PC3:Standby to SSU_PC7:Idle_Wait transition in SBC-3 for details about this transition.

9.2.10.2.5.4 Transition SA_PC_3:Standby to SA_PC_6:Wait_Standby

For devices that are not direct-access block devices, see the PC3:Standby to PC6:Wait_Standby transition in SPC-4 for details about this transition.

For direct-access block devices, see the SSU_PC3:Standby to SSU_PC6:Wait_Standby transition in SBC-3 for details about this transition.

9.2.10.2.5.5 Transition SA_PC_3:Standby to SA_PC_9: Standby_Wait

For direct-access block devices, see the SSU_PC3:Standby to SSU_PC9:Standby_Wait transition in SBC-3 for details about this transition.

9.2.10.2.5.6 Transition SA_PC_3:Standby to SA_PC_10:Wait_Stopped

For direct-access block devices, see the SSU_PC3:Standby to SSU_PC10:Wait_Stopped transition in SBC-3 for details about this transition.

9.2.10.2.6 SA_PC_4:Active_Wait state

9.2.10.2.6.1 State description

Upon entry into this state:

- a) if this state was entered with a Transitioning From Power On argument, Transitioning From Standby argument, or Transitioning From Stopped argument, then this state shall:
 - A) send a Phy Power Condition Status (Disable Low Phy Power Conditions) message to the management application layer.

While in this state:

- a) each idle condition timer that is enabled and not expired is running;
- b) each standby condition timer that is enabled and not expired is running;
- c) the device server shall provide pollable sense data (see SPC-4) with the sense key set to NO SENSE and the additional sense code set to LOGICAL UNIT TRANSITIONING TO ANOTHER POWER CONDITION; and
- d) the logical unit is performing the operations required for it to be in the SA_PC_1:Active state (e.g., a hard disk drive spins up its media).

If this state was entered with a Transitioning From Idle argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero (see SBC-3), that the device server is able to process and complete while in the SA_PC_2:Idle state;
- b) the peak power consumed in this state shall be no more than the typical peak power consumed in the SA_PC_1: Active state; and
- c) if:
 - A) this state was entered with a Transitioning From Idle_c argument; and
 - B) from idle command processing control is enabled (i.e., the FIDCPC field in the Power Condition mode page (see SPC-4) is set to 10b),

then the device server shall terminate any command, except START STOP UNIT commands, that requires the logical unit be in the SA_PC_1:Active state to continue processing, with CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT IS IN PROCESS OF BECOMING READY.

If the FIDCPC field contains 00b, then see SAS-2;

If this state was entered with a Transitioning From Standby argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero, that the device server is able to process and complete while in the SA_PC_3:Standby state;
- b) if from standby command processing control is enabled (i.e., the FSBCPC field in the Power Condition mode page (see SPC-4) is set to 10b), then the device server shall terminate any command, except START STOP UNIT commands, that requires the logical unit be in the SA_PC_1:Active state or SA_PC_2:Idle state to continue processing, with CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to:
 - A) LOGICAL UNIT NOT READY, NOTIFY (ENABLE SPINUP) REQUIRED, if NOTIFY (ENABLE SPINUP) has not been detected since entry into this state; or
 - B) LOGICAL UNIT IN THE PROCESS OF BECOMING READY, if NOTIFY (ENABLE SPINUP) has been detected since entry into this state.

If the FSBCPC field contains 00b, then see SAS-2;

- c) if NOTIFY (ENABLE SPINUP) has not been detected since entry into this state, then the peak power consumption shall be no more than the typical peak power consumed in the SA_PC_1: Active state; and
- d) if NOTIFY (ENABLE SPINUP) has been detected since entry into this state, then the peak power consumption is not limited by this standard.

If this state was entered with a Transitioning From Stopped argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero, that the device server is able to process and complete while in the SA_PC_8:Stopped state;
- b) if from stopped command processing control is enabled (i.e., the FSTCPC field in the Power Conditions mode page (see SPC-4) set to 10b), then the device server shall terminate any TEST UNIT READY command or media access command, with CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to:
 - A) LOGICAL UNIT NOT READY, NOTIFY (ENABLE SPINUP) REQUIRED, if NOTIFY (ENABLE SPINUP) has not been detected since entry into this state; or
 - B) LOGICAL UNIT IN THE PROCESS OF BECOMING READY, if NOTIFY (ENABLE SPINUP) has been detected since entry into this state.

If the FSTCPC field contains 00b, then see SAS-2;

- c) if NOTIFY (ENABLE SPINUP) has not been detected since entry into this state, then the peak power consumption shall be no more than the typical peak power consumed in the SA_PC_1: Active state; and
- d) if NOTIFY (ENABLE SPINUP) has been detected since entry into this state, then the peak power consumption is not limited by this standard.

If this state was entered with a Transitioning From Powered On argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero, that the device server is able to process and complete while in the SA_PC_8:Stopped state for direct-access devices;
- b) the device server shall terminate any TEST UNIT READY command or media access command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code set to:
 - A) LOGICAL UNIT NOT READY, NOTIFY (ENABLE SPINUP) REQUIRED, if NOTIFY (ENABLE SPINUP) has not been detected since entry into this state; or
 - B) LOGICAL UNIT IN THE PROCESS OF BECOMING READY, if NOTIFY (ENABLE SPINUP) has been detected since entry into this state.
- c) if NOTIFY (ENABLE SPINUP) has not been detected since entry into this state, then the peak power consumption shall be no more than the typical peak power consumed in the SA_PC_1: Active state; and
- d) if NOTIFY (ENABLE SPINUP) has been detected since entry into this state, then the peak power consumption is not limited by this standard.

If an idle condition timer or a standby condition timer is enabled and expires, then that timer shall be ignored.

9.2.10.2.6.2 Transition SA_PC_4:Active_Wait to SA_PC_1:Active

This transition shall occur when:

- a) the device server meets the requirements for the logical unit being in the SA_PC_1:Active state.

Before the transition, this state shall send a Phy Power Condition Status (Enable Low Phy Power Conditions) message to the management application layer.

9.2.10.2.7 SA_PC_5:Wait_Idle state

9.2.10.2.7.1 SA_PC_5:Wait_Idle state description

See the PC5:Wait_Idle state in SPC-4 for details about this state.

9.2.10.2.7.2 Transition SA_PC_5:Wait_Idle to SA_PC_2:Idle

See the PC5:Wait_Idle to PC2:Idle transition in SPC-4 for details about this transition.

9.2.10.2.8 SA_PC_6:Wait_Standby state**9.2.10.2.8.1 SA_PC_6:Wait_Standby state description**

See the PC6:Wait_Standby state in SPC-4 for details about this state.

9.2.10.2.8.2 Transition SA_PC_6:Wait_Standby to SA_PC_3:Standby

See the PC6:Wait_Standby to PC3:Standby transition in SPC-4 for details about this transition.

9.2.10.2.9 SA_PC_7:Idle_Wait state**9.2.10.2.9.1 State description**

Upon entry into this state:

- a) if this state was entered with a Transitioning From Standby argument or Transitioning From Stopped argument, then this state shall:
 - A) send a Phy Power Condition Status (Disable Low Phy Power Conditions) message to the management application layer.

While in this state:

- a) each idle condition timer that is enabled and not expired is running;
- b) each standby condition timer that is enabled and not expired is running;
- c) the device server shall provide pollable sense data (see SPC-4) with the sense key set to NO SENSE and the additional sense code set to LOGICAL UNIT TRANSITIONING TO ANOTHER POWER CONDITION; and
- d) the logical unit is performing the operations required for it to be in the SA_PC_2:Idle state (e.g., a hard disk drive spinning up its media).

If this state was entered with a Transitioning From Idle argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero (see SBC-3), that the device server is able to process and complete while in the SA_PC_2:Idle state; and
- b) the peak power consumed in this state shall be no more than the typical peak power consumed in the SA_PC_1: Active state.

If this state was entered with a Transitioning From Standby argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero, that the device server is able to process and complete while in the SA_PC_3:Standby state;
- b) if from standby command processing control is enabled (i.e., the FSBCPC field in the Power Condition mode page (see SPC-4) is set to 10b), then the device server shall terminate any command, except a START STOP UNIT command, that requires the logical unit be in the SA_PC_1:Active state or SA_PC_2:Idle state to continue processing, with CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to:
 - A) LOGICAL UNIT NOT READY, NOTIFY (ENABLE SPINUP) REQUIRED, if a NOTIFY (ENABLE SPINUP) has not been detected since entry into this state; or
 - B) LOGICAL UNIT IN THE PROCESS OF BECOMING READY, if a NOTIFY (ENABLE SPINUP) has been detected since entry into this state.

If the FSBCPC field contains 00b, then see SAS-2;

- c) if a NOTIFY (ENABLE SPINUP) has not been detected since entry into this state, then the peak power consumption shall be no more than the typical peak power consumed in the SA_PC_1: Active state; and
- d) if a NOTIFY (ENABLE SPINUP) has been detected since entry into this state, then the peak power consumption is not limited by this standard.

If this state was entered with a Transitioning From Stopped argument, then:

- a) the device server is capable of processing and completing the same commands, except a START STOP UNIT command with the IMMED bit set to zero, that the device server is able to process and complete while in the SA_PC_8:Stopped state;
- b) if from stopped command processing control is enabled (i.e., the FSTCPC field in the Power Conditions mode page (see SPC-4) set to 10b), then the device server shall terminate any TEST UNIT READY command or media access command, with CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to:
 - A) LOGICAL UNIT NOT READY, NOTIFY (ENABLE SPINUP) REQUIRED, if a NOTIFY (ENABLE SPINUP) has not been detected since entry into this state; or
 - B) LOGICAL UNIT IN THE PROCESS OF BECOMING READY, if a NOTIFY (ENABLE SPINUP) has been detected since entry into this state.

If the FSTCPC field contains 00b, then see SAS-2;

- c) if a NOTIFY (ENABLE SPINUP) has not been detected, then the peak power consumption shall be no more than the typical peak power consumed in the SA_PC_1: Active state; and
- d) if a NOTIFY (ENABLE SPINUP) has been detected, then the peak power consumption is not limited.

If an idle condition timer or a standby condition timer is enabled and expires, then that timer shall be ignored.

9.2.10.2.9.2 Transition SA_PC_5:Idle_Wait to SA_PC_2:Idle

This transition shall occur when:

- a) the device server meets the requirements for the logical unit being in the:
 - A) idle_a power condition, if this was entered with a Transitioning To Idle_a argument;
 - B) idle_b power condition, if this was entered with a Transitioning To Idle_b argument; or
 - C) idle_c power condition, if this was entered with a Transitioning To Idle_c argument.

Before the transition, this state shall send a Phy Power Condition Status (Enable Low Phy Power Conditions) message to the management application layer.

9.2.10.2.10 SA_PC_8:Stopped state

9.2.10.2.10.1 State description

This state is only implemented in logical units that are direct-access block devices.

See the SSU_PC8:Stopped state in SBC-3 for details about this state.

9.2.10.2.10.2 Transition SA_PC_8:Stopped to SA_PC_4:Active_Wait

For direct-access block devices, see the SSU_PC8:Stopped to SSU_PC4:Active_Wait transition in SBC-3 for details about this transition.

The transition shall include a From Stopped argument.

9.2.10.2.10.3 Transition SA_PC_8:Stopped to SA_PC_7:Idle_Wait

For direct-access block devices, see the SSU_PC8:Stopped to SSU_PC7:Idle_Wait transition in SBC-3 for details about this transition.

9.2.10.2.10.4 Transition SA_PC_8:Stopped to SA_PC_9:Standby_Wait

For direct-access block devices, see the SSU_PC8:Stopped to SSU_PC9:Standby_Wait transition in SBC-3 for details about this transition.

9.2.10.2.11 SA_PC_9:Standby_Wait state

9.2.10.2.11.1 SA_PC_9:Standby_Wait state description

This state is only implemented in logical units that are direct-access block devices.

See the SSU_PC9:Standby_Wait in SBC-3 for details about this state.

9.2.10.2.11.2 Transition SA_PC_9:Standby_Wait to SA_PC_3:Standby

For direct-access block devices, see the SSU_PC9:Standby_Wait to SSU_PC3:Standby transition in SBC-3 for details about this transition.

9.2.10.2.12 SA_PC_10:Wait_Stopped state

9.2.10.2.12.1 SA_PC_10:Wait_Stopped state description

This state is only implemented in logical units that are direct-access block devices.

See the SSU_PC10:Standby_Wait in SBC-3 for details about this state.

9.2.10.2.12.2 Transition SA_PC_10:Wait_Stopped to SA_PC_8:Stopped

For direct-access block devices, see the SSU_PC10:Wait_Stopped to SSU_PC8:Stopped transition in SBC-3 for details about this transition.

9.2.11 SCSI vital product data (VPD)

9.2.11.1 SCSI vital product data (VPD) overview

Table 193 lists VPD pages for which this standard defines special requirements.

Table 193 — VPD pages with special requirements for SAS SSP

Page code	VPD Page Name	Reference	Support Requirements
83h	Device Identification VPD page	9.2.11.2 and SPC-4	Mandatory
90h	Protocol-Specific Logical Unit Information VPD page	9.2.11.3 and SPC-4	See ^a
^a Mandatory if the target port and logical unit support the TLR CONTROL field set to a non-zero value in the SSP frame header, otherwise optional.			

9.2.11.2 Device Identification VPD page

In the Device Identification VPD page (83h) returned in response to an INQUIRY command (see SPC-4), each logical unit in a SAS target device shall include the designation descriptors for the target port identifier (see 4.2.9) and the relative target port identifier (see SAM-5 and SPC-4) listed in table 194.

Table 194 — Device Identification VPD page designation descriptors for the SAS target port

Field in designation descriptor	Designation descriptor	
	Target port identifier	Relative target port identifier
DESIGNATOR TYPE	3h (i.e., NAA)	4h (i.e., relative target port identifier)
ASSOCIATION	01b (i.e., SCSI target port)	01b (i.e., SCSI target port)
CODE SET	1h (i.e., binary)	1h (i.e., binary)
DESIGNATOR LENGTH	8	4
PIV (protocol identifier valid)	1	1
PROTOCOL IDENTIFIER	6h (i.e., SAS)	6h (i.e., SAS)
DESIGNATOR	SAS address ^a (see 4.2.4)	Relative port identifier ^b as described in SAM-5 and SPC-4
^a The DESIGNATOR field contains the SAS address of the SSP target port through which the INQUIRY command was received. ^b The DESIGNATOR field contains the relative port identifier of the SSP target port through which the INQUIRY command was received.		

In the Device Identification VPD page (83h) returned in response to an INQUIRY command (see SPC-4), each logical unit in a SAS target device shall include a designation descriptor for the SAS target device name (see 4.2.6) using NAA format and may include a designation descriptor for the SAS target device name using the SCSI name string format as listed in table 195.

Table 195 — Device Identification VPD page designation descriptors for the SAS target device

Field in designation descriptor	Designation descriptor for SAS target device	
	NAA format (required)	SCSI name string format (optional)
DESIGNATOR TYPE	3h (i.e., NAA)	8h (i.e., SCSI name string)
ASSOCIATION	10b (i.e., SCSI target device)	10b (i.e., SCSI target device)
CODE SET	1h (i.e., binary)	3h (i.e., UTF-8)
DESIGNATOR LENGTH	8	24
PIV (protocol identifier valid)	1	0
PROTOCOL IDENTIFIER	6h (i.e., SAS)	0h ^a
DESIGNATOR	Device name of the SAS target device in SAS address format (see 4.2.4)	Device name of the SAS target device in SCSI name string format (e.g., "naa." followed by 16 hexadecimal digits followed by 4 ASCII null characters)
^a The PROTOCOL IDENTIFIER field is reserved when the PIV bit is set to zero.		

Logical units may include designation descriptors in addition to those required by this standard (e.g., SCSI target devices with SCSI target ports using other SCSI transport protocols may return additional SCSI target device names for those other SCSI transport protocols).

9.2.11.3 Protocol-Specific Logical Unit Information VPD page

The Protocol-Specific Logical Unit Information VPD page (see SPC-4) contains parameters for the logical unit that are protocol-specific based on the I_T nexus being used to access the logical unit.

Table 196 defines the Protocol-Specific Logical Unit Information VPD page for logical units with SAS target ports.

Table 196 — Protocol-Specific Logical Unit Information VPD page for SAS SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (90h)							
2	(MSB)	PAGE LENGTH (n - 3)						
3								(LSB)
Logical unit information descriptor list								
4	Logical unit information descriptor (first) (see table 197)							
...	...							
	Logical unit information descriptor (last) (see table 197)							
n								

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in SPC-4.

The PAGE CODE field is defined in SPC-4 and shall be set to the value defined in table 196.

The PAGE LENGTH field is defined in SPC-4 and shall be set to the value defined in table 196.

The logical unit information descriptor list is defined in SPC-4 and shall contain a logical unit information descriptor for each SAS target port known to the device server.

Table 197 defines the logical unit information descriptor for logical units with SAS target ports.

Table 197 — Logical unit information descriptor for SAS SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB) _____							
1	RELATIVE PORT IDENTIFIER _____ (LSB)							
2	Reserved				PROTOCOL IDENTIFIER (6h)			
3	Reserved _____							
5								
6	(MSB) _____							
7	DESCRIPTOR LENGTH (0004h) _____ (LSB)							
Per logical unit SCSI transport specific data								
8	Reserved							TLR CONTROL SUPPORTED
9	Reserved _____							
11								

The RELATIVE PORT IDENTIFIER field is defined in SPC-4.

The PROTOCOL IDENTIFIER field is defined in SPC-4 and shall be set to the value defined in table 197 indicating that this is a SAS SSP specific descriptor.

The DESCRIPTOR LENGTH field is defined in SPC-4 and shall be set to the value defined in table 197.

A TLR CONTROL SUPPORTED bit set to one indicates that the combination of the SCSI target port and logical unit support the TLR CONTROL field in the SSP frame header (see 8.2.1). A TLR CONTROL SUPPORTED bit set to zero indicates that the combination of the SCSI target port and logical unit do not support the TLR CONTROL field in the SSP frame header.

9.3 ATA application layer

No SAS-specific ATA features are defined by this standard.

9.4 Management application layer

9.4.1 READY LED signal behavior

A SAS target device uses the READY LED signal to activate an externally visible LED that indicates the state of readiness and activity of the SAS target device. The READY LED signal electrical characteristics are described in SAS-2.1.

The system is not required to generate any visual output when the READY LED signal is asserted. Additional vendor specific flashing patterns may be used to signal vendor specific conditions.

SAS target devices without SSP target ports may transmit the READY LED signal using vendor specific methods and patterns.

SAS target devices with SSP target ports shall follow the behavior indicated by the READY LED MEANING bit in the Protocol-Specific Port mode page (see 9.2.7.4) as described in table 198.

Table 198 — READY LED signal behavior

Power condition ^a (see 9.2.10) or activity	READY LED MEANING bit set to zero ^b	READY LED MEANING bit set to one
active power condition or an idle power condition	The SAS target device shall: a) when not processing a command, assert the READY LED signal continuously; or b) when processing a command, toggle the READY LED signal between the asserted and negated states in a vendor-specified manner. (i.e., the LED is usually on, but flashes on and off when commands are processed)	The SAS target device shall: a) when not processing a command, negate the READY LED signal continuously; or b) when processing a command, toggle the READY LED signal between the asserted and negated states in a vendor-specified manner. (i.e., The LED is usually off, but flashes on and off when commands are processed)
a standby power condition or stopped power condition	The SAS target device shall: a) when not processing a command, negate the READY LED signal continuously; or b) when processing a command, toggle the READY LED signal between the asserted and negated states in a vendor-specified manner. (i.e., the LED is usually off, but flashes on and off when commands are processed) After a vendor specific amount of time in this state, SAS target devices with rotating media may be removed with minimum risk of mechanical or electrical damage.	
spinup/spin down	If the SAS target device has rotating media and is in the process of performing a spinup or spindown, then the SAS target device shall toggle the READY LED signal between the asserted and negated states with a 1 s ± 0.1 s cycle using a 50 % ± 10 % duty cycle (e.g., the LED is on for 0.5 s and off for 0.5 s).	
formatting media	If the SAS target device is in the process of formatting media, then the SAS target device shall toggle the READY LED signal between the asserted and negated states in a vendor-specified manner (e.g., with each cylinder change on a disk drive).	
^a If the SAS target device has more than one logical unit and any logical unit is active or idle, then the power condition of that logical unit should be used to control the READY LED signal. ^b If the target device has rotating media, a READY LED MEANING bit set to zero results in a READY LED signal behavior that provides an indication of the target device's readiness for removal. A target device with rotating media that is not in a state for safe removal shall either toggle the READY LED signal at a significant rate during spinup, during spindown, and while formatting media, or assert the READY LED signal continuously. When removal is safe from a mechanical standpoint, the READY LED signal shall be negated.		

9.4.2 Management protocol services

The management application client and management device server use a four-step process to perform management functions:

- 1) The management application client invokes Send SMP Function;
- 2) The SMP target port invokes SMP Function Received;
- 3) The management device server invokes Send SMP Function Response; and

- 4) The SMP initiator port invokes Received SMP Function Complete.

9.4.3 SMP functions

9.4.3.1 SMP functions overview

Table 199 defines the SMP functions.

Table 199 — SMP functions (FUNCTION field) (part 1 of 2)

Function code	SMP function	Description	Reference
SMP input functions (00h to 7Fh)			
General SMP input functions (00h to 0Fh)			
00h	REPORT GENERAL	Return general information about the device	9.4.3.4
01h	REPORT MANUFACTURER INFORMATION	Return vendor and product identification	9.4.3.5
02h	Obsolete		
03h	REPORT SELF-CONFIGURATION STATUS	Return status of the discover process in a self-configuring expander device	9.4.3.6
04h	REPORT ZONE PERMISSION TABLE	Return zone permission table values	9.4.3.7
05h	REPORT ZONE MANAGER PASSWORD	Return the zone manager password	9.4.3.8
06h	REPORT BROADCAST	Return information about Broadcast counters	9.4.3.9
07h	Restricted for SFF		
08h to 0Fh	Reserved for general SMP input functions		
Phy-based SMP input functions (10h to 1Fh)			
10h	DISCOVER	Return information about the specified phy	9.4.3.10
11h	REPORT PHY ERROR LOG	Return error logging information about the specified phy	9.4.3.11
12h	REPORT PHY SATA	Return information about a phy currently attached to a SATA phy	9.4.3.12
13h	REPORT ROUTE INFORMATION	Return phy-based expander route table information	9.4.3.13
14h	REPORT PHY EVENT	Return phy events for the specified phy	9.4.3.14
15h to 1Fh	Reserved for phy-based SMP input functions		
Descriptor list-based SMP input functions (20h to 2Fh)			
20h	DISCOVER LIST	Return information about the specified phys	9.4.3.15
21h	REPORT PHY EVENT LIST	Return phy events	9.4.3.16
22h	REPORT EXPANDER ROUTE TABLE LIST	Return contents of the expander-based expander route table	9.4.3.17
23h to 2Fh	Reserved for descriptor list-based SMP input functions		
Other			
30h to 3Fh	Reserved for SMP input functions		

Table 199 — SMP functions (FUNCTION field) (part 2 of 2)

Function code	SMP function	Description	Reference
40h to 7Fh	Vendor specific		
SMP output functions (80h to FFh)			
General SMP output functions (80h to 8Fh)			
80h	CONFIGURE GENERAL	Configure the device	9.4.3.18
81h	ENABLE DISABLE ZONING	Enable or disable zoning	9.4.3.19
82h	Obsolete		
83h	Restricted for SFF		
84h	Reserved for general SMP output functions		
85h	ZONED BROADCAST	Transmit the specified Broadcast on the expander ports in the specified zone group(s)	9.4.3.20
86h	ZONE LOCK	Lock a zoning expander device	9.4.3.21
87h	ZONE ACTIVATE	Set the zoning expander current values equal to the zoning expander shadow values	9.4.3.22
88h	ZONE UNLOCK	Unlock a zoning expander device	9.4.3.23
89h	CONFIGURE ZONE MANAGER PASSWORD	Configure the zone manager password	9.4.3.24
8Ah	CONFIGURE ZONE PHY INFORMATION	Configure zone phy information	9.4.3.25
8Bh	CONFIGURE ZONE PERMISSION TABLE	Configure the zone permission table	9.4.3.26
8Ch to 8Fh	Reserved for general SMP output functions		
Phy-based SMP output functions (90h to 9Fh)			
90h	CONFIGURE ROUTE INFORMATION	Change phy-based expander route table information	9.4.3.27
91h	PHY CONTROL	Request actions by the specified phy	9.4.3.28
92h	PHY TEST FUNCTION	Request a test function by the specified phy	9.4.3.29
93h	CONFIGURE PHY EVENT	Configure phy events for the specified phy	9.4.3.30
94h to 9Fh	Reserved for phy-based SMP output functions		
Other			
A0h to BFh	Reserved for SMP output functions		
C0h to FFh	Vendor specific		

9.4.3.2 SMP function request frame format

9.4.3.2.1 SMP function request frame format overview

An SMP request frame is sent by a management application client via an SMP initiator port to request an SMP function be performed by a management device server. Table 200 defines the SMP request frame format.

Table 200 — SMP request frame format

Byte ^a \Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION							
2	ALLOCATED RESPONSE LENGTH							
3	REQUEST LENGTH ((00h or ((n - 7) / 4))							
4	Additional request bytes							
n - 4								
n - 3	(MSB)	CRC						
n								(LSB)

^a Shaded byte numbers (e.g., bytes 0 through 3 and (n - 3) through n) show the bytes that are included in the request frame when the REQUEST LENGTH field is set to 00h. Functions defined in SAS-1.1 may be defined as containing more than eight bytes when the REQUEST LENGTH field is set to 00h.

9.4.3.2.2 SMP FRAME TYPE field

The SMP FRAME TYPE field is defined by the SMP transport layer (see 8.4.1) and parsed by the SMP transport layer state machines (see 8.4.5). The SMP FRAME TYPE field is set to the value defined in table 200 (see 9.4.3.2.1).

9.4.3.2.3 FUNCTION field

The FUNCTION field specifies which SMP function is being requested and is defined in table 199 (see 9.4.3.1). If the management device server does not support the value in the FUNCTION field, then it shall return a function result of UNKNOWN SMP FUNCTION as described in table 202 (see 9.4.3.3.4).

9.4.3.2.4 ALLOCATED RESPONSE LENGTH field

The ALLOCATED RESPONSE LENGTH field specifies the maximum number of dwords that the management application client has allocated in the data-in buffer for the additional response bytes in the response frame (see 9.4.3.3).

For compatibility with SAS-1.1, an ALLOCATED RESPONSE LENGTH field set to 00h specifies that a specific number of dwords are to be transferred as defined in the SMP function description. If the SMP function description does not specify a specific number of dwords, then the number of dwords to be transferred is zero. This condition shall not be considered as an error.

If the LONG RESPONSE bit is set to one in the REPORT GENERAL response (see 9.4.3.4), then the management application client may set the ALLOCATED RESPONSE LENGTH field to a non-zero value in all SMP request frames. If the LONG RESPONSE bit is set to zero in the REPORT GENERAL response, then the management application client shall set the ALLOCATED RESPONSE LENGTH field to 00h in all SMP request frames.

If the ALLOCATED RESPONSE LENGTH field is set to a non-zero value, then the management device server shall truncate the additional response bytes to the number of dwords specified by the ALLOCATED RESPONSE LENGTH field.

If the ALLOCATED RESPONSE LENGTH field is set to 00h, then the management device server shall truncate the additional response bytes to the number of dwords specified by the SMP function description.

The allocated response length is used to limit the maximum amount of variable length data returned to the management application client. Fields in the additional response bytes (e.g., fields containing counts of the number of dwords in some or all of the data) shall not be altered to reflect the truncation, if any, that results from an insufficient allocated response length.

9.4.3.2.5 REQUEST LENGTH field

A REQUEST LENGTH field set to 00h specifies that either:

- a) no dwords follow the REQUEST LENGTH field before the CRC field; or
- b) a non-zero number of dwords follow the REQUEST LENGTH field before the CRC field. This is for compatibility with SAS-1.1.

The function description defines the interpretation of a REQUEST LENGTH field set to 00h.

A REQUEST LENGTH field set to a non-zero value (i.e., the non-zero value defined in table 200 (see 9.4.3.2.1)) specifies the number of dwords that follow the REQUEST LENGTH field before the CRC field (i.e., the length of the entire request frame minus two).

If the LONG RESPONSE bit is set to one in the REPORT GENERAL response (see 9.4.3.4), then the management application client may set the REQUEST LENGTH field to a non-zero value in the SMP request frame for any SMP function. If the LONG RESPONSE bit is set to zero in the REPORT GENERAL response, then the management application client shall set the REQUEST LENGTH field to 00h in the SMP request frame for every SMP function.

If the request frame size including the CRC field is less than 8 bytes, or the REQUEST LENGTH field does not match the request frame size, then the management device server shall return a function result of INVALID REQUEST FRAME LENGTH.

The management device server shall consider any fields not included in the request frame to be set to zero.

9.4.3.2.6 Additional request bytes

The additional request bytes definition and length are based on the SMP function.

The number of additional request bytes are an integer multiple of four, so the CRC field is aligned on a four byte boundary.

The maximum number of additional request bytes is 1 020, making the maximum size of the frame 1 028 bytes (i.e., 4 bytes of header + 1 020 bytes of data + 4 bytes of CRC).

NOTE 87 Management application clients compliant with SAS-1.1 may send a vendor specific SMP request frame containing 1 024 additional request bytes. The SMP_TP state machine discards SMP request frames that exceed 1 023 request bytes (see 6.19.5.4.2.2). SMP request frames defined in those versions of this standard did not have more than 36 additional request bytes.

If the management device server receives more additional request bytes than it expects (e.g., the management device server complies with a version of this standard defining 24 additional request bytes, but receives a request frame containing 36 additional request bytes), then it shall return a function result of INVALID REQUEST FRAME LENGTH.

For additional request bytes containing a DESCRIPTOR LENGTH field and a descriptor list, if the management device server receives more bytes in each descriptor than it expects (e.g., the management device server complies with a version of this standard defining that a descriptor is 12 bytes, but receives a request frame containing a descriptor list with 16 byte descriptors), then it shall return a function result of INVALID REQUEST FRAME LENGTH.

9.4.3.2.7 CRC field

The CRC field is defined by the SMP transport layer (see 8.4.1) and parsed by the SMP link layer state machines (see 6.19.5).

9.4.3.3 SMP function response frame format

9.4.3.3.1 SMP function response frame format overview

An SMP response frame is sent by a management device server via an SMP target port in response to an SMP request frame. Table 201 defines the SMP response frame format.

Table 201 — SMP response frame format

Byte ^a \Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION							
2	FUNCTION RESULT							
3	RESPONSE LENGTH (00h or ((n - 7) / 4))							
4	Additional response bytes							
n - 4								
n - 3	(MSB)	CRC						
n								(LSB)

^a Shaded byte numbers (e.g., bytes 0 through 3 and (n - 3) through n) show the bytes that are included in the response frame when the ALLOCATED RESPONSE LENGTH field is set to 00h in the request frame. Functions defined in SAS-1.1 may be defined as returning more than eight bytes when the ALLOCATED RESPONSE LENGTH field is set to 00h.

9.4.3.3.2 SMP FRAME TYPE field

The SMP FRAME TYPE field is defined by the SMP transport layer (see 8.4.1) and parsed by the MT state machines (see 8.4.5). The SMP FRAME TYPE field is set to the value defined in table 201 (see 9.4.3.3.1).

9.4.3.3.3 FUNCTION field

The FUNCTION field indicates the SMP function to which this frame is a response, and is defined in table 199 (see 9.4.3.1).

9.4.3.3.4 FUNCTION RESULT field

The FUNCTION RESULT field is defined in table 202.

Table 202 — FUNCTION RESULT field (part 1 of 4)

Code	Name	SMP function(s)	Description
00h	SMP FUNCTION ACCEPTED	All	The management device server supports the SMP function and processed the SMP request.
01h	UNKNOWN SMP FUNCTION	Unknown	The management device server does not support the requested SMP function.
02h	SMP FUNCTION FAILED	All	The requested SMP function failed.
03h	INVALID REQUEST FRAME LENGTH	All	The SMP request frame length was invalid (see 9.4.3.2).
04h	INVALID EXPANDER CHANGE COUNT	CONFIGURE GENERAL, ENABLE DISABLE ZONING, ZONE LOCK, ZONE ACTIVATE, CONFIGURE ZONE MANAGER PASSWORD, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ZONE PERMISSION TABLE, CONFIGURE ROUTE INFORMATION, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT	The management device server supports the SMP function, but the EXPECTED EXPANDER CHANGE COUNT field does not match the current expander change count.
05h	BUSY	ZONE UNLOCK, ENABLE DISABLE ZONING, CONFIGURE ZONE MANAGER PASSWORD, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ZONE PERMISSION TABLE	For ZONE UNLOCK, the locked zoning expander device is processing the activate step. For the other functions, the management device server is currently saving zoning values.
06h	INCOMPLETE DESCRIPTOR LIST	ZONED BROADCAST, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ZONE PERMISSION TABLE, CONFIGURE PHY EVENT	The request frame length results in the truncation of a multi-byte field or descriptor list (e.g., in the ZONED BROADCAST request (see table 267), the request frame is not large enough to contain the number of Broadcast source zone groups specified by the NUMBER OF BROADCAST SOURCE ZONE GROUPS field).

Table 202 — FUNCTION RESULT field (part 2 of 4)

Code	Name	SMP function(s)	Description
10h	PHY DOES NOT EXIST	DISCOVER, REPORT PHY ERROR LOG, REPORT PHY SATA, REPORT ROUTE INFORMATION, REPORT PHY EVENT, DISCOVER LIST, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ROUTE INFORMATION, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT	The phy specified by the PHY IDENTIFIER field or the STARTING PHY IDENTIFIER field in the SMP request frame does not exist (e.g., the value is not less than the value indicated in the NUMBER OF PHYS field in the SMP REPORT GENERAL response).
11h	INDEX DOES NOT EXIST	REPORT ROUTE INFORMATION, CONFIGURE ROUTE INFORMATION	The phy specified by the PHY IDENTIFIER field in the SMP request frame does not have the table routing attribute (see 4.6.7.1), or the expander route index specified by the EXPANDER ROUTE INDEX field does not exist (i.e., the value is not in the range of 0000h to the value of the EXPANDER ROUTE INDEXES field in the SMP REPORT GENERAL response). The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
12h	PHY DOES NOT SUPPORT SATA	REPORT PHY SATA, PHY CONTROL	See 9.4.3.12 and 9.4.3.28
13h	UNKNOWN PHY OPERATION	PHY CONTROL	See 9.4.3.28
14h	UNKNOWN PHY TEST FUNCTION	PHY TEST FUNCTION	See 9.4.3.29
15h	PHY TEST FUNCTION IN PROGRESS	PHY TEST FUNCTION	See 9.4.3.29
16h	PHY VACANT	DISCOVER, REPORT PHY ERROR LOG, REPORT PHY SATA, REPORT ROUTE INFORMATION, REPORT PHY EVENT, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ROUTE INFORMATION, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT	The management device server processing the SMP request frame does not have access to a specified phy (e.g., because of zoning or vendor specific reasons), although the value is less than the value indicated in the NUMBER OF PHYS field in the SMP REPORT GENERAL response.

Table 202 — FUNCTION RESULT field (part 3 of 4)

Code	Name	SMP function(s)	Description
17h	UNKNOWN PHY EVENT SOURCE	CONFIGURE PHY EVENT	See 9.4.3.30.3
18h	UNKNOWN DESCRIPTOR TYPE	DISCOVER LIST	The descriptor type specified by the DESCRIPTOR TYPE field is not supported.
19h	UNKNOWN PHY FILTER	DISCOVER LIST	The phy filter specified by the PHY FILTER field is not supported.
1Ah	AFFILIATION VIOLATION	PHY CONTROL	The specified phy operation is not allowed due to the current state of affiliations.
20h	SMP ZONE VIOLATION	CONFIGURE GENERAL, ZONED BROADCAST, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT	Zoning is enabled and the SMP initiator port does not have access to a necessary zone group according to the zone permission table (see 4.9.3.2).
21h	NO MANAGEMENT ACCESS RIGHTS	REPORT ZONE MANAGER PASSWORD, ZONE LOCK, CONFIGURE ZONE MANAGER PASSWORD	For ZONE LOCK see 9.4.3.21. For REPORT ZONE MANAGER PASSWORD, see 9.4.3.8. For CONFIGURE ZONE MANAGER PASSWORD, see 9.4.3.24.
22h	UNKNOWN ENABLE DISABLE ZONING VALUE	ENABLE DISABLE ZONING	See 9.4.3.19
23h	ZONE LOCK VIOLATION	ENABLE DISABLE ZONING, ZONE LOCK, ZONE ACTIVATE, ZONE UNLOCK, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ZONE PERMISSION TABLE	Zoning is enabled and: a) an unlocked zoning expander device receives an SMP zone configuration function request, a ZONE ACTIVATE request, or a ZONE UNLOCK request; or b) a locked zoning expander device receives an SMP zone configuration function request, a ZONE ACTIVATE request, or a ZONE UNLOCK request from an SMP initiator port that is not the active zone manager.
24h	NOT ACTIVATED	ZONE UNLOCK	The following conditions are true: a) the ACTIVATE REQUIRED bit in the request is set to one; and b) the locked zoning expander device has not processed the activate step.
25h	ZONE GROUP OUT OF RANGE	CONFIGURE ZONE PHY INFORMATION, CONFIGURE ZONE PERMISSION TABLE	The ZONE GROUP field or NUMBER OF ZONE GROUPS field contains a value that is not supported.
26h	NO PHYSICAL PRESENCE	CONFIGURE ZONE MANAGER PASSWORD	The following conditions are true: a) the NEW ZONE MANAGER PASSWORD field is set to DISABLED (see table 27 in 4.9.1); and b) physical presence is not asserted.

Table 202 — FUNCTION RESULT field (part 4 of 4)

Code	Name	SMP function(s)	Description
27h	SAVING NOT SUPPORTED	REPORT ZONE PERMISSION TABLE, REPORT ZONE MANAGER PASSWORD, ENABLE DISABLE ZONING, CONFIGURE ZONE MANAGER PASSWORD, CONFIGURE ZONE PHY INFORMATION, CONFIGURE ZONE PERMISSION TABLE	For REPORT ZONE PERMISSION TABLE, see 9.4.3.7 For REPORT ZONE MANAGER PASSWORD, see 9.4.3.8 For ENABLE DISABLE ZONING, CONFIGURE ZONE MANAGER PASSWORD, CONFIGURE ZONE PHY INFORMATION, and CONFIGURE ZONE PERMISSION TABLE, the following conditions are true: a) The SAVE field is set to 01b or 11b; and b) the management device server does not support saved values for the specified information.
28h	SOURCE ZONE GROUP DOES NOT EXIST	REPORT ZONE PERMISSION TABLE	See 9.4.3.7
29h	DISABLED PASSWORD NOT SUPPORTED	CONFIGURE ZONE MANAGER PASSWORD	See 9.4.3.24
All others	Reserved		

Table 203 defines the priority of the SMP function results defined in table 202.

Table 203 — Function result priority (part 1 of 4)

SMP function	SMP function result priority
REPORT GENERAL (see 9.4.3.4)	1) INVALID REQUEST FRAME LENGTH; 2) SMP FUNCTION FAILED; and 3) SMP FUNCTION ACCEPTED
REPORT MANUFACTURER INFORMATION (see 9.4.3.5)	1) INVALID REQUEST FRAME LENGTH; 2) SMP FUNCTION FAILED; and 3) SMP FUNCTION ACCEPTED
REPORT SELF-CONFIGURATION STATUS (see 9.4.3.6)	1) INVALID REQUEST FRAME LENGTH; 2) SMP FUNCTION FAILED; and 3) SMP FUNCTION ACCEPTED
REPORT ZONE PERMISSION TABLE (see 9.4.3.7)	1) INVALID REQUEST FRAME LENGTH; 2) SOURCE ZONE GROUP DOES NOT EXIST; 3) SAVING NOT SUPPORTED; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
REPORT ZONE MANAGER PASSWORD (see 9.4.3.8)	1) INVALID REQUEST FRAME LENGTH; 2) NO MANAGEMENT ACCESS RIGHTS; 3) SAVING NOT SUPPORTED; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
REPORT BROADCAST (see 9.4.3.9)	1) INVALID REQUEST FRAME LENGTH; 2) SMP FUNCTION FAILED; and 3) SMP FUNCTION ACCEPTED

Table 203 — Function result priority (part 2 of 4)

SMP function	SMP function result priority
DISCOVER (see 9.4.3.10)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
REPORT PHY ERROR LOG (see 9.4.3.11)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
REPORT PHY SATA (see 9.4.3.12)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) PHY DOES NOT SUPPORT SATA; 5) SMP FUNCTION FAILED; and 6) SMP FUNCTION ACCEPTED
REPORT ROUTE INFORMATION (see 9.4.3.13)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) INDEX DOES NOT EXIST; 5) SMP FUNCTION FAILED; and 6) SMP FUNCTION ACCEPTED
REPORT PHY EVENT (see 9.4.3.14)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
DISCOVER LIST (see 9.4.3.15)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) UNKNOWN DESCRIPTOR TYPE; 4) UNKNOWN PHY FILTER; 5) SMP FUNCTION FAILED; and 6) SMP FUNCTION ACCEPTED
REPORT PHY EVENT LIST (see 9.4.3.16)	1) INVALID REQUEST FRAME LENGTH; 2) SMP FUNCTION FAILED; and 3) SMP FUNCTION ACCEPTED
REPORT EXPANDER ROUTE TABLE LIST (see 9.4.3.17)	1) INVALID REQUEST FRAME LENGTH; 2) SMP FUNCTION FAILED; and 3) SMP FUNCTION ACCEPTED
CONFIGURE GENERAL (see 9.4.3.18)	1) INVALID REQUEST FRAME LENGTH; 2) SMP ZONE VIOLATION; 3) INVALID EXPANDER CHANGE COUNT; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED

Table 203 — Function result priority (part 3 of 4)

SMP function	SMP function result priority
ENABLE DISABLE ZONING (see 9.4.3.19)	1) INVALID REQUEST FRAME LENGTH; 2) ZONE LOCK VIOLATION; 3) UNKNOWN ENABLE DISABLE ZONING VALUE; 4) INVALID EXPANDER CHANGE COUNT; 5) SAVING NOT SUPPORTED; 6) SMP FUNCTION FAILED; and 7) SMP FUNCTION ACCEPTED
ZONED BROADCAST (see 9.4.3.20)	1) INVALID REQUEST FRAME LENGTH; 2) INCOMPLETE DESCRIPTOR LIST; 3) SMP ZONE VIOLATION; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
ZONE LOCK (see 9.4.3.21)	1) INVALID REQUEST FRAME LENGTH; 2) NO MANAGEMENT ACCESS RIGHTS; 3) INVALID EXPANDER CHANGE COUNT; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
ZONE ACTIVATE (see 9.4.3.22)	1) INVALID REQUEST FRAME LENGTH; 2) ZONE LOCK VIOLATION; 3) INVALID EXPANDER CHANGE COUNT; 4) SMP FUNCTION FAILED; and 5) SMP FUNCTION ACCEPTED
ZONE UNLOCK (see 9.4.3.23)	1) INVALID REQUEST FRAME LENGTH; 2) ZONE LOCK VIOLATION; 3) NOT ACTIVATED; 4) BUSY; 5) SMP FUNCTION FAILED; and 6) SMP FUNCTION ACCEPTED
CONFIGURE ZONE MANAGER PASSWORD (see 9.4.3.24)	1) INVALID REQUEST FRAME LENGTH; 2) INVALID EXPANDER CHANGE COUNT; 3) NO MANAGEMENT ACCESS RIGHTS; 4) NO PHYSICAL PRESENCE; 5) SAVING NOT SUPPORTED; 6) DISABLED PASSWORD NOT SUPPORTED; 7) SMP FUNCTION FAILED; and 8) SMP FUNCTION ACCEPTED
CONFIGURE ZONE PHY INFORMATION (see 9.4.3.25)	1) INVALID REQUEST FRAME LENGTH; 2) INCOMPLETE DESCRIPTOR LIST; 3) PHY DOES NOT EXIST; 4) PHY VACANT; 5) ZONE LOCK VIOLATION; 6) INVALID EXPANDER CHANGE COUNT; 7) SAVING NOT SUPPORTED; 8) ZONE GROUP OUT OF RANGE; 9) SMP FUNCTION FAILED; and 10) SMP FUNCTION ACCEPTED

Table 203 — Function result priority (part 4 of 4)

SMP function	SMP function result priority
CONFIGURE ZONE PERMISSION TABLE (see 9.4.3.26)	1) INVALID REQUEST FRAME LENGTH; 2) INCOMPLETE DESCRIPTOR LIST; 3) ZONE LOCK VIOLATION; 4) INVALID EXPANDER CHANGE COUNT; 5) SAVING NOT SUPPORTED; 6) ZONE GROUP OUT OF RANGE; 7) SMP FUNCTION FAILED; and 8) SMP FUNCTION ACCEPTED
CONFIGURE ROUTE INFORMATION (see 9.4.3.27)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) INVALID EXPANDER CHANGE COUNT; 5) INDEX DOES NOT EXIST; 6) SMP FUNCTION FAILED; and 7) SMP FUNCTION ACCEPTED
PHY CONTROL (see 9.4.3.28)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) SMP ZONE VIOLATION; 5) INVALID EXPANDER CHANGE COUNT; 6) UNKNOWN PHY OPERATION; 7) PHY DOES NOT SUPPORT SATA; 8) AFFILIATION VIOLATION; 9) SMP FUNCTION FAILED; and 10) SMP FUNCTION ACCEPTED
PHY TEST FUNCTION (see 9.4.3.29)	1) INVALID REQUEST FRAME LENGTH; 2) PHY DOES NOT EXIST; 3) PHY VACANT; 4) SMP ZONE VIOLATION; 5) INVALID EXPANDER CHANGE COUNT; 6) UNKNOWN PHY TEST FUNCTION; 7) PHY TEST FUNCTION IN PROGRESS; 8) SMP FUNCTION FAILED; and 9) SMP FUNCTION ACCEPTED
CONFIGURE PHY EVENT (see 9.4.3.30)	1) INVALID REQUEST FRAME LENGTH; 2) INCOMPLETE DESCRIPTOR LIST; 3) PHY DOES NOT EXIST; 4) PHY VACANT; 5) SMP ZONE VIOLATION; 6) INVALID EXPANDER CHANGE COUNT; 7) UNKNOWN PHY EVENT SOURCE; 8) SMP FUNCTION FAILED; and 9) SMP FUNCTION ACCEPTED

9.4.3.3.5 RESPONSE LENGTH field

A RESPONSE LENGTH field set to 00h indicates that either:

- a) no dwords follow the RESPONSE LENGTH field before the CRC field; or
- b) a non-zero number of dwords follow the RESPONSE LENGTH field before the CRC field. This is for compatibility with SAS-1.1.

The function description defines the interpretation of a RESPONSE LENGTH field set to 00h.

A RESPONSE LENGTH field set to a non-zero value (i.e., the non-zero value defined in table 201 (see 9.4.3.3.1)) indicates the number of dwords that follow the RESPONSE LENGTH field before the CRC field (i.e., the length of the entire response frame minus two).

9.4.3.3.6 Additional response bytes

If the FUNCTION RESULT field is set to 00h, then the additional response bytes definition depends on the SMP function requested. If the FUNCTION RESULT field is set to a value other than 00h, then the additional response bytes may be present but shall be ignored.

The number of additional response bytes are an integer multiple of four, so the CRC field is aligned on a four byte boundary.

The maximum number of additional response bytes is 1 020, making the maximum size of the frame 1 028 bytes (i.e., 4 bytes of header + 1 020 bytes of data + 4 bytes of CRC).

NOTE 88 Management device servers compliant with SAS-1.1 may send a vendor specific SMP response frame containing 1 024 additional response bytes. The SMP_IP state machine discards SMP response frames that exceed 1 023 request bytes (see 6.19.5.3.4). SMP response frames defined in those versions of this standard did not have more than 56 additional response bytes.

The management application client should ignore any additional response bytes beyond those that it expects (e.g., if the management application client complies with a version of this standard defining 24 additional response bytes, but receives a response frame containing 36 additional response bytes, then it should ignore the last 12 additional response bytes).

For additional response bytes containing a DESCRIPTOR LENGTH field and a descriptor list, the management application client should ignore any bytes in each descriptor beyond those that it expects (e.g., if the management application client complies with a version of this standard defining that a descriptor has 24 bytes, but receives a response frame containing a descriptor list with 36 byte descriptors, then it should ignore the last 12 bytes of each descriptor).

9.4.3.3.7 CRC field

The CRC field is defined by the SMP transport layer (see 8.4.1) and parsed by the SMP link layer state machines (see 6.19.5).

9.4.3.4 REPORT GENERAL function

The REPORT GENERAL function returns general information about the SAS device (e.g., a SAS device contained in an expander device). This SMP function shall be implemented by all management device servers.

Table 204 defines the request format.

Table 204 — REPORT GENERAL request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (00h)							
2	ALLOCATED RESPONSE LENGTH							
3	REQUEST LENGTH (00h)							
4	(MSB)	CRC						
7								(LSB)

The SMP FRAME TYPE field is defined in 9.4.3.2.2 and shall be set to the value defined in table 204.

The FUNCTION field is defined in 9.4.3.2.3 and shall be set to the value defined in table 204.

The ALLOCATED RESPONSE LENGTH field is defined in 9.4.3.2.4.

If the ALLOCATED RESPONSE LENGTH field is set to 00h, then the management device server shall:

- a) set the RESPONSE LENGTH field to 00h in the response frame; and
- b) return the first 28 bytes defined in table 205 plus the CRC field as the response frame.

If the ALLOCATED RESPONSE LENGTH field is not set to 00h, then the management device server shall:

- a) set the RESPONSE LENGTH field in the response frame to the non-zero value defined in table 205; and
- b) return the response frame as defined in 9.4.3.2.4.

The REQUEST LENGTH field is defined in 9.4.3.2.5 and shall be set to the value defined in table 204.

The CRC field is defined in 9.4.3.2.7.

Table 205 defines the response format.

Table 205 — REPORT GENERAL response (part 1 of 3)

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (00h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH (00h or 11h)							
4	(MSB) EXPANDER CHANGE COUNT (LSB)							
5								
6	(MSB) EXPANDER ROUTE INDEXES (LSB)							
7								
8	LONG RESPONSE	Reserved						
9	NUMBER OF PHYS							
10	TABLE TO TABLE SUPPORTED	ZONE CONFIGURING	SELF CONFIGURING	STP CONTINUE AWT	OPEN REJECT RETRY SUPPORTED	CONFIGURES OTHERS	CONFIGURING	EXTERNALLY CONFIGURABLE ROUTE TABLE
11	Reserved							
12								
19	ENCLOSURE LOGICAL IDENTIFIER							
20								
27	Reserved							

Table 205 — REPORT GENERAL response (part 3 of 3)

Byte\Bit	7	6	5	4	3	2	1	0
60	(MSB) _____							
61	LAST SELF-CONFIGURATION STATUS DESCRIPTOR INDEX _____ (LSB)							
62	(MSB) _____							
63	MAXIMUM NUMBER OF STORED SELF-CONFIGURATION STATUS DESCRIPTORS _____ (LSB)							
64	(MSB) _____							
65	LAST PHY EVENT LIST DESCRIPTOR INDEX _____ (LSB)							
66	(MSB) _____							
67	MAXIMUM NUMBER OF STORED PHY EVENT LIST DESCRIPTORS _____ (LSB)							
68	(MSB) _____							
69	STP REJECT TO OPEN LIMIT _____ (LSB)							
70	Reserved _____							
71								
72	(MSB) _____							
75	CRC _____ (LSB)							

The SMP FRAME TYPE field is defined in 9.4.3.3.2 and shall be set to the value defined in table 205.

The FUNCTION field is defined in 9.4.3.3.3 and shall be set to the value defined in table 205.

The FUNCTION RESULT field is defined in 9.4.3.3.4.

The RESPONSE LENGTH field is defined in 9.4.3.3.5 and shall be set to one of the values defined in table 205 based on the ALLOCATED RESPONSE LENGTH field in the request frame.

The EXPANDER CHANGE COUNT field indicates the number of Broadcast (Change)s originated by an expander device (see 6.12). Management device servers in expander devices shall support this field. Management device servers in other device types (e.g., end devices) shall set this field to 0000h. This field shall be set to at least 0001h at power on. If the expander device has originated Broadcast (Change) for any reason described in 6.12 since transmitting any SMP response frame containing an EXPANDER CHANGE COUNT field, then it:

- shall increment this field at least once from the value in the previous REPORT GENERAL response; and
- shall not increment this field when forwarding a Broadcast (Change).

This field shall wrap to at least 0001h after reaching the maximum value (i.e., FFFFh).

NOTE 89 Application clients that use the EXPANDER CHANGE COUNT field should read it often enough to ensure that it does not increment a multiple of 65 535 times between reading the field in an expander device compliant with this standard or a multiple of 65 536 times between reading the field in an expander device compliant with SAS-1.1.

NOTE 90 Management device servers in expander devices compliant with SAS-1.1 may return an EXPANDER CHANGE COUNT field set to 0000h.

NOTE 91 The originated Broadcast (Change) count is also reported in the REPORT BROADCAST response (see 9.4.3.9).

The EXPANDER ROUTE INDEXES field indicates the maximum number of expander route indexes per phy for the expander device (see 4.6.7.4). Management device servers in externally configurable expander devices containing phy-based expander route tables shall support this field. Management device servers in other device types (e.g., end devices, externally configurable expander devices with expander-based expander route tables, and self-configuring expander devices) shall set the EXPANDER ROUTE INDEXES field to 0000h. Not all phys in an externally configurable expander device are required to support the maximum number indicated by this field.

A LONG RESPONSE bit set to one indicates that the management device server supports returning non-zero values in the RESPONSE LENGTH field of the response frame for any SMP function when the ALLOCATED RESPONSE LENGTH field in the request frame for that SMP function is set to a non-zero value. The LONG RESPONSE bit shall be set to one.

NOTE 92 Devices compliant with SAS-1.1 set the LONG RESPONSE bit to zero in the REPORT GENERAL response and set the RESPONSE LENGTH field to 00h in all SMP response frames.

The NUMBER OF PHYS field indicates the number of phys in the device, including any virtual phys and any vacant phys.

A TABLE TO TABLE SUPPORTED bit set to one indicates that the expander device is a self-configuring expander device that supports its table routing phys being attached to table routing phys in other expander devices (i.e., table-to-table attachment). The TABLE TO TABLE SUPPORTED bit shall only be set to one if the EXTERNALLY CONFIGURABLE ROUTE TABLE bit is set to zero. A TABLE TO TABLE SUPPORTED bit set to zero indicates that the expander device is not a self-configuring expander device that supports its table routing phys being attached to table routing phys in other expander devices.

A ZONE CONFIGURING bit set to one indicates that the zoning expander device is locked and the zoning expander shadow values differ from the zoning expander current values. A ZONE CONFIGURING bit set to zero indicates that is not true. Management device servers in zoning expander devices shall support this bit. Management device servers in non-zoning expander devices and in other device types shall set this bit to zero.

A SELF CONFIGURING bit set to one indicates that the management device server is in a self-configuring expander device, the self-configuring expander device's management application client is currently performing the discover process (see 4.7), and that management application client has identified at least one change to its expander routing table. Management device servers in self-configuring expander devices shall support this bit. Management device servers in externally configurable expander devices and in other device types shall set this bit to zero.

An STP CONTINUE AWT bit set to one specifies that the STP port shall not stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when the STP port receives an OPEN_REJECT (RETRY). An STP CONTINUE AWT bit set to zero specifies that the STP port shall stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when the STP port receives an OPEN_REJECT (RETRY).

An OPEN REJECT RETRY SUPPORTED bit set to one indicates that the expander device returns OPEN_REJECT (RETRY) for any connection requests that would otherwise have resulted in OPEN_REJECT (NO DESTINATION) while the SELF CONFIGURING bit is set to one (see 4.7.4) or the ZONE CONFIGURING bit is set to one (see 4.9.6.3). An OPEN REJECT RETRY SUPPORTED set to zero indicates that the expander device complies with SAS-1.1 (i.e., it returns OPEN_REJECT (NO DESTINATION) while the CONFIGURING bit is set to one). Self-configuring expander devices compliant with this standard shall set the OPEN REJECT RETRY SUPPORTED bit to one.

A CONFIGURES OTHERS bit set to one indicates that the expander device is a self-configuring expander device that performs the configuration subprocess defined in 4.8. A CONFIGURES OTHERS bit set to zero indicates that the expander device may or may not perform the configuration subprocess. Self-configuring expander devices compliant with this standard shall set the CONFIGURES OTHERS bit to one.

NOTE 93 If the CONFIGURES OTHERS bit is set to zero, then the expander device may configure all externally configurable expander devices in the SAS domain.

The CONFIGURING bit indicates the logical OR of the ZONE CONFIGURING bit and the SELF CONFIGURING bit. Changes in this bit from one to zero result in a Broadcast (Change) being originated (see 6.12). Management device servers that support the ZONE CONFIGURING bit or the SELF CONFIGURING bit shall support this bit.

An EXTERNALLY CONFIGURABLE ROUTE TABLE bit set to one indicates that the management device server is in an externally configurable expander device that has a phy-based expander route table that is required to be configured with the SMP CONFIGURE ROUTE INFORMATION function (see 4.6.7.4). An EXTERNALLY CONFIGURABLE ROUTE TABLE bit set to zero indicates that the management device server is not in an externally configurable expander device (e.g., the management device server is in an end device, in a self-configuring expander device, or in an expander device with no phys with table routing attributes).

The ENCLOSURE LOGICAL IDENTIFIER field identifies the enclosure, if any, in which the device is located, and is defined in SES-3. The ENCLOSURE LOGICAL IDENTIFIER field shall be set to the same value reported by the enclosure services process, if any, for the enclosure. An ENCLOSURE LOGICAL IDENTIFIER field set to 00000000 00000000h indicates no enclosure information is available.

The STP BUS INACTIVITY TIME LIMIT field indicates the bus inactivity time limit for STP connections, which is set by the CONFIGURE GENERAL function (see 9.4.3.18).

The STP MAXIMUM CONNECT TIME LIMIT field indicates the maximum connect time limit for STP connections, which is set by the CONFIGURE GENERAL function (see 9.4.3.18).

The STP SMP I_T NEXUS LOSS TIME field indicates the minimum time that an STP target port and an SMP initiator port retry certain connection requests which is set by the CONFIGURE GENERAL function (see 9.4.3.18).

The NUMBER OF ZONE GROUPS field indicates the number of zone groups (e.g., the number of entries in the zone group permission table) supported by the expander device and is defined in table 206.

Table 206 — NUMBER OF ZONE GROUPS field

Code	Description
00b	128 zone groups
01b	256 zone groups
All others	Reserved

A ZONE LOCKED bit set to one indicates that the zoning expander device is locked (see 4.9.6.2). A ZONE LOCKED bit set to zero indicates that the zoning expander device is not locked.

A PHYSICAL PRESENCE SUPPORTED bit set to one indicates that the expander device supports physical presence as a mechanism for allowing locking from phys in zone groups without access to zone group 2. A PHYSICAL PRESENCE SUPPORTED bit set to zero indicates that the expander device does not support physical presence as a mechanism for allowing locking.

A PHYSICAL PRESENCE ASSERTED bit set to one indicates that the expander device is currently detecting physical presence. A PHYSICAL PRESENCE ASSERTED bit set to zero indicates that the expander device is not currently detecting physical presence. The PHYSICAL PRESENCE ASSERTED bit shall be set to zero if the PHYSICAL PRESENCE SUPPORTED bit is set to zero.

A ZONING SUPPORTED bit set to one indicates that zoning is supported by the expander device (i.e., it is a zoning expander device). A ZONING SUPPORTED bit set to zero indicates that zoning is not supported by the expander device.

A ZONING ENABLED bit set to one indicates that zoning is enabled in the expander device. A ZONING ENABLED bit set to zero indicates that zoning is disabled in the expander device. The ZONING ENABLED bit shall be set to zero if the ZONING SUPPORTED bit is set to zero.

A SAVING bit set to one indicates that the management device server is currently saving zoning values to non-volatile storage and may return a function result of BUSY for zone management functions that access saved zoning values. A SAVING bit set to zero indicates that the management device server is not currently saving zoning values to non-volatile storage.

A SAVING ZONE MANAGER PASSWORD SUPPORTED bit set to one indicates that saving the zone manager password is supported. A SAVING ZONE MANAGER PASSWORD SUPPORTED bit set to zero indicates that saving the zone manager password is not supported.

A SAVING ZONE PHY INFORMATION SUPPORTED bit set to one indicates that saving the zone phy information is supported. A SAVING ZONE PHY INFORMATION SUPPORTED bit set to zero indicates that saving the zone phy information is not supported.

A SAVING ZONE PERMISSION TABLE SUPPORTED bit set to one indicates that saving the zone permission table is supported. A SAVING ZONE PERMISSION TABLE SUPPORTED bit set to zero indicates that saving the zone permission table is not supported.

A SAVING ZONING ENABLE SUPPORTED bit set to one indicates that saving the ZONING ENABLED bit is supported. A SAVING ZONING ENABLE SUPPORTED bit set to zero indicates that saving the ZONING ENABLED bit is not supported.

The MAXIMUM NUMBER OF ROUTED SAS ADDRESSES field indicates the number of routed SAS addresses in an expander-based expander route table (see 4.6.7.4 and 4.9.3.4). Management device servers in expander devices containing expander-based expander route tables shall support this field. Management device servers in other device types (e.g., end devices and expander devices with phy-based expander route tables) shall set this field to 0000h.

The ACTIVE ZONE MANAGER SAS ADDRESS field indicates the SAS address (see 4.2.4) of the zone manager that last locked the zoning expander device. If the zoning expander device is currently being configured by a vendor specific sideband method then the ACTIVE ZONE MANAGER SAS ADDRESS field shall be set to 00000000 00000000h. This field shall be set to 00000000 00000000h at power on.

The ZONE LOCK INACTIVITY TIME LIMIT field indicates the minimum time between any SMP ZONE LOCK requests, SMP zone configuration function requests, or SMP ZONE ACTIVATE requests from the active zone manager that the locked expander device allows and is set in the SMP ZONE LOCK request (see 9.4.3.21).

The FIRST ENCLOSURE CONNECTOR ELEMENT INDEX field indicates the lowest CONNECTOR ELEMENT INDEX field of all the expander phys in all the expander devices in the enclosure that have CONNECTOR TYPE fields set to 20h through 2Fh (i.e., an internal connector to an end device) in their SMP DISCOVER responses.

The NUMBER OF ENCLOSURE CONNECTOR ELEMENT INDEXES field indicates the number of expander phys in all the expander devices in the enclosure that have CONNECTOR TYPE fields set to 20h through 2Fh (i.e., an internal connector to an end device) in their SMP DISCOVER responses.

NOTE 94 The NUMBER OF ENCLOSURE CONNECTOR ELEMENT INDEXES field assumes that all internal connectors to end devices are assigned to a contiguous range of CONNECTOR ELEMENT INDEX field values.

A REDUCED FUNCTIONALITY bit set to one indicates that:

- a) the expander device is scheduled to reduce its functionality (see 4.6.8) in the time indicated in the TIME TO REDUCED FUNCTIONALITY field; or
- b) the expander device is currently operating with reduced functionality (see 4.6.8).

A REDUCED FUNCTIONALITY bit set to zero indicates that the expander device is not scheduled to reduce functionality and that the contents of the TIME TO REDUCED FUNCTIONALITY field shall be ignored.

If the REDUCED FUNCTIONALITY bit is set to one, then the TIME TO REDUCED FUNCTIONALITY field indicates the time, in 100 ms increments, remaining until the expander device is scheduled to reduce functionality. The expander device starts the reduced functionality delay timer after originating a Broadcast (Expander) (see 4.6.8).

The INITIAL TIME TO REDUCED FUNCTIONALITY field indicates the minimum period of time, in 100 ms increments, that an expander device waits from originating a Broadcast (Expander) to reducing functionality. The expander device should set the default value for the INITIAL TIME TO REDUCED FUNCTIONALITY field to at least 2 000 ms (i.e., 14h).

The MAXIMUM REDUCED FUNCTIONALITY TIME field indicates the maximum time, in seconds, that the expander device responds with OPEN_REJECT (RETRY) to connection requests that map to an expander phy or an

SMP target port that is not accessible during expander device reduced functionality. This timer starts after the reduced functionality delay timer expires.

The LAST SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field is defined in the REPORT SELF-CONFIGURATION STATUS response (see 9.4.3.6).

The MAXIMUM NUMBER OF STORED SELF-CONFIGURATION STATUS DESCRIPTORS field indicates the maximum number of self-configuration status descriptors (see 9.4.3.6.4) that the management device server supports.

The LAST PHY EVENT LIST DESCRIPTOR INDEX field is defined in the REPORT PHY EVENT LIST response (see 9.4.3.16).

The MAXIMUM NUMBER OF STORED PHY EVENT LIST DESCRIPTORS field indicates the maximum number of phy event list descriptors (see 9.4.3.14.4) that the management device server supports.

The STP REJECT TO OPEN LIMIT field indicates the minimum time in 10 μ s increments that an STP port waits to establish a connection request with an initiator port on an I_T nexus after receiving an OPEN_REJECT (RETRY), OPEN_REJECT (RESERVED CONTINUE 0), or OPEN_REJECT (RESERVED CONTINUE 1). An STP REJECT TO OPEN LIMIT field set to 0000h indicates that the time limit is vendor specific.

The CRC field is defined in 9.4.3.3.7.

9.4.3.5 REPORT MANUFACTURER INFORMATION function

The REPORT MANUFACTURER INFORMATION function returns vendor and product identification. This SMP function may be implemented by any management device server.

Table 207 defines the request format.

Table 207 — REPORT MANUFACTURER INFORMATION request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (01h)							
2	ALLOCATED RESPONSE LENGTH							
3	REQUEST LENGTH (00h)							
4	(MSB)	CRC						
7								(LSB)

The SMP FRAME TYPE field is defined in 9.4.3.2.2 and shall be set to the value defined in table 207.

The FUNCTION field is defined in 9.4.3.2.3 and shall be set to the value defined in table 207.

The ALLOCATED RESPONSE LENGTH field is defined in 9.4.3.2.4.

If the ALLOCATED RESPONSE LENGTH field is set to 00h, then the management device server shall:

- set the RESPONSE LENGTH field to 00h in the response frame; and
- return the first 60 bytes defined in table 208 plus the CRC field as the response frame.

If the ALLOCATED RESPONSE LENGTH field is not set to 00h, then the management device server shall:

- set the RESPONSE LENGTH field in the response frame to the non-zero value defined in table 208; and
- return the response frame as defined in 9.4.3.2.4.

The REQUEST LENGTH field is defined in 9.4.3.2.5 and shall be set to the value defined in table 207.

The CRC field is defined in 9.4.3.2.7.

Table 208 defines the response format.

Table 208 — REPORT MANUFACTURER INFORMATION response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (01h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH (00h or 0Eh)							
4	(MSB)	EXPANDER CHANGE COUNT						(LSB)
5								
6		Reserved						
7								
8		Reserved						SAS-1.1 FORMAT
9		Reserved						
11								
12	(MSB)	VENDOR IDENTIFICATION						(LSB)
19								
20	(MSB)	PRODUCT IDENTIFICATION						(LSB)
35								
36	(MSB)	PRODUCT REVISION LEVEL						(LSB)
39								
40	(MSB)	COMPONENT VENDOR IDENTIFICATION						(LSB)
47								
48	(MSB)	COMPONENT ID						(LSB)
49								
50		COMPONENT REVISION LEVEL						
51		Reserved						
52		Vendor specific						
59								
60	(MSB)	CRC						(LSB)
63								

The SMP FRAME TYPE field is defined in 9.4.3.3.2 and shall be set to the value defined in table 208.

The FUNCTION field is defined in 9.4.3.3.3 and shall be set to the value defined in table 208.

The FUNCTION RESULT field is defined in 9.4.3.3.4.

The RESPONSE LENGTH field is defined in 9.4.3.3.5 and shall be set to one of the values defined in table 208 based on the ALLOCATED RESPONSE LENGTH field in the request frame.

The EXPANDER CHANGE COUNT field is defined in the SMP REPORT GENERAL response (see 9.4.3.4).

A SAS-1.1 FORMAT bit set to one indicates that bytes 40 through 59 are as defined in this standard. A SAS-1.1 FORMAT bit set to zero indicates that bytes 40 through 59 are vendor specific as defined in the original version of this standard.

ASCII data fields (e.g., the VENDOR IDENTIFICATION field, the PRODUCT IDENTIFICATION field, and PRODUCT REVISION LEVEL field, and the COMPONENT VENDOR IDENTIFICATION field) shall contain only graphic codes (i.e., code values 20h through 7Eh). Left-aligned fields shall place any unused bytes at the end of the field (i.e., at the highest offset) and the unused bytes shall be filled with space characters (i.e., 20h).

The VENDOR IDENTIFICATION field contains eight bytes of ASCII data identifying the vendor of the subsystem (e.g., the board or enclosure) containing the component. The data shall be left-aligned within the field. The vendor identification string shall be one assigned by INCITS for use in the standard INQUIRY data VENDOR IDENTIFICATION field. A list of assigned vendor identification strings is in SPC-4 and on the T10 web site (see <http://www.t10.org>).

The PRODUCT IDENTIFICATION field contains 16 bytes of ASCII data identifying the type of the subsystem (e.g., the board or enclosure model number) containing the component, as defined by the vendor of the subsystem. The data shall be left-aligned within the field. The PRODUCT IDENTIFICATION field should be changed whenever the subsystem design changes in a way noticeable to a user (e.g., a different stock-keeping unit (SKU)).

The PRODUCT REVISION LEVEL field contains four bytes of ASCII data identifying the revision level of the subsystem (e.g., the board or enclosure) containing the component, as defined by the vendor of the subsystem. The data shall be left-aligned within the field. The PRODUCT REVISION LEVEL field should be changed whenever the subsystem design changes (e.g., any component change, even including resistor values).

All components on a subsystem should have the same values for their VENDOR IDENTIFICATION fields, PRODUCT IDENTIFICATION fields, and PRODUCT REVISION LEVEL fields.

NOTE 95 Application clients may use the VENDOR IDENTIFICATION field and PRODUCT IDENTIFICATION field to identify the subsystem (e.g., for a user interface). Application clients may use the VENDOR IDENTIFICATION field, PRODUCT IDENTIFICATION field, PRODUCT REVISION LEVEL field to perform workarounds for problems in a specific revision of a subsystem.

The COMPONENT VENDOR IDENTIFICATION field contains eight bytes of ASCII data identifying the vendor of the component (e.g., the expander device) containing the management device server. The data shall be left-aligned within the field. The component vendor identification string shall be one assigned by INCITS for use in the standard INQUIRY data VENDOR IDENTIFICATION field. A list of assigned vendor identification strings is in SPC-4 and on the T10 web site (see <http://www.t10.org>).

The COMPONENT ID field contains a 16-bit identifier identifying the type of the component (e.g., the expander device model number) containing the management device server, as defined by the vendor of the component. The COMPONENT ID field should be changed whenever the component's programming interface (e.g., the management device server definition) changes.

The COMPONENT REVISION LEVEL field contains an 8-bit identifier identifying the revision level of the component (e.g., the expander device) containing the management device server, as defined by the vendor of the component. The COMPONENT REVISION LEVEL field should be changed whenever the component changes but its programming interface does not change.

NOTE 96 Application clients may use the COMPONENT VENDOR IDENTIFICATION field and the COMPONENT ID field to interpret vendor specific information (e.g., vendor specific SMP functions) correctly for that component. Application clients may use the COMPONENT VENDOR IDENTIFICATION field, the COMPONENT ID field, and the COMPONENT REVISION LEVEL field to perform workarounds for problems in a specific revision of a component.

The vendor specific bytes are defined by the vendor of the subsystem (e.g., the board or enclosure) containing the component.

The CRC field is defined in 9.4.3.3.7.

9.4.3.6 REPORT SELF-CONFIGURATION STATUS function

9.4.3.6.1 REPORT SELF-CONFIGURATION STATUS function overview

The REPORT SELF-CONFIGURATION STATUS function returns self-configuration expander device status. This SMP function shall be implemented by the management device server in self-configuring expander devices and shall not be implemented by any other management device servers.

9.4.3.6.2 REPORT SELF-CONFIGURATION STATUS request

Table 209 defines the request format.

Table 209 — REPORT SELF-CONFIGURATION STATUS request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (03h)							
2	ALLOCATED RESPONSE LENGTH							
3	REQUEST LENGTH (01h)							
4	Reserved							
5								
6	(MSB)	STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX						
7								(LSB)
8	(MSB)	CRC						
11								(LSB)

The SMP FRAME TYPE field is defined in 9.4.3.2.2 and shall be set to the value defined in table 209.

The FUNCTION field is defined in 9.4.3.2.3 and shall be set to the value defined in table 209.

The ALLOCATED RESPONSE LENGTH field is defined in 9.4.3.2.4.

The REQUEST LENGTH field is defined in 9.4.3.2.5 and shall be set to the value defined in table 209.

The STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field specifies the first self-configuration status descriptor that the management device server shall return in the SMP response frame. If the specified index does not contain a valid self-configuration status descriptor, then the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field in the response may differ from the specified index. A STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field set to 0000h specifies that the device server shall return no self-configuration status descriptors.

The CRC field is defined in 9.4.3.2.7.

9.4.3.6.3 REPORT SELF-CONFIGURATION STATUS response

Table 210 defines the response format.

Table 210 — REPORT SELF-CONFIGURATION STATUS response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (03h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH ((n - 7) / 4)							
4	(MSB)	EXPANDER CHANGE COUNT						
5								(LSB)
6	(MSB)	STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX						
7								(LSB)
8	(MSB)	TOTAL NUMBER OF SELF-CONFIGURATION STATUS DESCRIPTORS						
9								(LSB)
10	(MSB)	LAST SELF-CONFIGURATION STATUS DESCRIPTOR INDEX						
11								(LSB)
12	SELF-CONFIGURATION STATUS DESCRIPTOR LENGTH							
13	Reserved							
18								
19	NUMBER OF SELF-CONFIGURATION STATUS DESCRIPTORS							
Self-configuration status descriptor list								
20	Self-configuration status descriptor (first) (see table 211 in 9.4.3.6.4)							
...	...							
n - 4	Self-configuration status descriptor (last) (see table 211 in 9.4.3.6.4)							
n - 3	(MSB)	CRC						
n								(LSB)

The SMP FRAME TYPE field is defined in 9.4.3.3.2 and shall be set to the value defined in table 210.

The FUNCTION field is defined in 9.4.3.3.3 and shall be set to the value defined in table 210.

The FUNCTION RESULT field is defined in 9.4.3.3.4.

The RESPONSE LENGTH field is defined in 9.4.3.3.5 and shall be set to the value defined in table 210. A RESPONSE LENGTH field set to 00h does not have a special meaning based on the ALLOCATED RESPONSE LENGTH field in the request frame.

The EXPANDER CHANGE COUNT field is defined in the SMP REPORT GENERAL response (see 9.4.3.4). If the management application client detects a change in the value of this field while retrieving multiple response frames, then it should retrieve the response frames again

The STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field indicates the index of the first self-configuration status descriptor being returned. If the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field in the SMP request is set to 0000h, then the management device server shall:

- a) set the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field to 0000h;
- b) set the TOTAL NUMBER OF SELF-CONFIGURATION STATUS DESCRIPTORS field to 0000h; and
- c) return no descriptors.

If the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field in the SMP request does not specify a valid descriptor, then the management device server shall:

- a) set the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field to the next index, in ascending order wrapping from FFFFh to 0001h, that contains a valid descriptor.

If the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field is not set to 0000h and specifies a valid descriptor, then this field shall be set to the same value as the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field in the SMP request.

The SELF-CONFIGURATION STATUS DESCRIPTOR LENGTH field indicates the length, in dwords, of the self-configuration status descriptor (see table 211 in 9.4.3.6.4).

The TOTAL NUMBER OF SELF-CONFIGURATION STATUS DESCRIPTORS field indicates the number of self-configuration status descriptors are available at this time from the management device server.

The LAST SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field indicates the index of the last recorded self-configuration status descriptor.

The NUMBER OF SELF-CONFIGURATION STATUS DESCRIPTORS field indicates the number of self-configuration status descriptors in the self-configuration status descriptor list.

The self-configuration status descriptor list contains self-configuration status descriptors (see table 211). The management device server shall return either all the self-configuration status descriptors that fit in one SMP response frame or all the self-configuration status descriptors until the index indicated in the LAST SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field is reached. The self-configuration status descriptor list shall start with the self-configuration status descriptor specified by the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field, and shall continue with self-configuration status descriptors sorted in ascending order, wrapping from FFFFh to 0001h, based on the self-configuration status descriptor index. The self-configuration status descriptor list shall not contain any truncated self-configuration status descriptors. If the STARTING SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field is equal to the LAST SELF-CONFIGURATION STATUS DESCRIPTOR INDEX field, then the self-configuration status descriptor at that index shall be returned.

The CRC field is defined in 9.4.3.3.7.

9.4.3.6.4 Self-configuration status descriptor

Each self-configuration status descriptor follows the format defined in table 211.

Table 211 — Self-configuration status descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	STATUS TYPE							
1	Reserved							FINAL
2	Reserved							
3	PHY IDENTIFIER							
4	Reserved							
7								
8								
15	SAS ADDRESS							

The STATUS TYPE field indicates the type of status being reported and is defined in table 212.

Table 212 — STATUS TYPE field (part 1 of 2)

Code	Description
00h	Reserved
01h	Error not related to a specific layer
02h	The expander device currently has a connection or is currently attempting to establish a connection with the SMP target port with the indicated SAS address.
03h	Expander route table is full. The expander device was not able to add the indicated SAS address to the expander route table.
04h	Expander device is out of resources (e.g., it discovered too many SAS addresses while performing the discover process through a subtractive port). This does not affect the expander route table.
05h to 1Fh	Reserved for status not related to specific layers
Status reported by the phy layer	
20h	Error reported by the phy layer
21h	All phys in the expander port containing the indicated phy lost dword synchronization
22h to 3Fh	Reserved for status reported by the phy layer
Status reported by the link layer	
40h	Error reported by the link layer
41h	Connection request failed: Open Timeout timer expired
42h	Connection request failed: Received an abandon-class OPEN_REJECT (e.g., BAD DESTINATION, PROTOCOL NOT SUPPORTED, ZONE VIOLATION, STP RESOURCES BUSY, WRONG DESTINATION)
43h	Connection request failed: Received a vendor specific number of retry-class OPEN_REJECTs (e.g. RETRY, PATHWAY BLOCKED)
44h	Connection request failed: I_T nexus loss occurred (e.g., OPEN_REJECT (NO DESTINATION) received for longer than the time specified by the STP SMP I_T NEXUS LOSS TIME field in the CONFIGURE GENERAL function)

Table 212 — STATUS TYPE field (part 2 of 2)

Code	Description
45h	Connection request failed: Received BREAK
46h	Connection established: SMP response frame had a CRC error
47h to 5Fh	Reserved for status reported by the link layer
Status reported by the port layer	
60h	Error reported by the port layer
61h	During an SMP connection, there was no SMP response frame within the maximum SMP connection time
62h to 7Fh	Reserved for status reported by the port layer
Status reported by the SMP transport layer	
80h	Error reported by the SMP transport layer
81h to 9Fh	Reserved for status reported by the SMP transport layer
Status reported by the management application layer	
A0h	Error reported by the management application layer
A1h	SMP response frame is too short
A2h	SMP response frame contains field(s) with unsupported values
A3h	SMP response frame contains results inconsistent with other SMP response frames (e.g., the DISCOVER response ATTACHED SAS ADDRESS field does not contain the SAS address the expander device expected)
A4h	The SAS ADDRESS field contains the SAS address of a self-configuring expander device that returned a REPORT GENERAL response with the CONFIGURING bit set to one, the SELF CONFIGURING bit set to zero, and the ZONE CONFIGURING bit set to zero (e.g., compliant with a previous version of this standard). Accesses to SAS addresses two or more levels beyond this expander device may not succeed until the indicated expander device completes configuration. This may or may not be an error.
A5h	The SAS ADDRESS field contains the SAS address of a self-configuring expander device that returned a REPORT GENERAL response with the SELF CONFIGURING bit set to one. Accesses to SAS addresses two or more levels beyond this expander device may not succeed until the indicated expander device completes configuration. This may or may not be an error.
A6h	The SAS ADDRESS field contains the SAS address of a self-configuring expander device that returned a REPORT GENERAL response with the ZONE CONFIGURING bit set to one. Accesses to SAS addresses two or more levels beyond this expander device may not succeed until the indicated expander device completes configuration. This may or may not be an error.
A7h to BFh	Reserved for status reported by the management application layer
Other status	
C0h to DFh	Reserved
E0h to FFh	Vendor specific

A FINAL bit set to one indicates that the expander device is no longer attempting to establish connections to the SMP target port with the indicated SAS address as part of the discover process because of the error indicated by the descriptor. A FINAL bit set to zero indicates that the expander device is still attempting to access the SMP target port with the indicated SAS address as part of the discover process.

The PHY IDENTIFIER field indicates the phy identifier (see 4.2.10) of the phy that was used to request a connection with the SMP target port with the indicated SAS address.

The SAS ADDRESS field indicates the SAS address (see 4.2.4) of the SMP target port to which the expander device established a connection or attempted to establish a connection.

9.4.3.7 REPORT ZONE PERMISSION TABLE function

9.4.3.7.1 REPORT ZONE PERMISSION TABLE function overview

The REPORT ZONE PERMISSION function returns a set of zone permission table entries. This function shall be supported by all zoning expander devices.

9.4.3.7.2 REPORT ZONE PERMISSION TABLE request

Table 213 defines the request format.

Table 213 — REPORT ZONE PERMISSION TABLE request

Byte\Bit	7	6	5	4	3	2	1	0		
0	SMP FRAME TYPE (40h)									
1	FUNCTION (04h)									
2	ALLOCATED RESPONSE LENGTH									
3	REQUEST LENGTH (01h)									
4	Reserved						REPORT TYPE			
5	Reserved									
6	STARTING SOURCE ZONE GROUP									
7	MAXIMUM NUMBER OF ZONE PERMISSION DESCRIPTORS									
8	(MSB)									
11	CRC									
	(LSB)									

The SMP FRAME TYPE field is defined in 9.4.3.2.2 and shall be set to the value defined in table 213.

The FUNCTION field is defined in 9.4.3.2.3 and shall be set to the value defined in table 213.

The ALLOCATED RESPONSE LENGTH field is defined in 9.4.3.2.4.

The REQUEST LENGTH field is defined in 9.4.3.2.5 and shall be set to the value defined in table 213.

The REPORT TYPE field specifies the zone permission table values that the management device server shall return and is defined in table 214.

Table 214 — REPORT TYPE field

Code	Description
00b	Current zone permission table
01b	Shadow zone permission table
10b	Saved zone permission table. If the expander device does not support saving, it shall return a function result of SAVING NOT SUPPORTED in the response frame (see table 202 in 9.4.3.3).
11b	Default zone permission table

The STARTING SOURCE ZONE GROUP field specifies the first source zone group, (i.e., s) returned. If the value in this field exceeds the end of the zone permission table, then the management device server shall return a function result of SOURCE ZONE GROUP DOES NOT EXIST in the response frame (see table 202 in 9.4.3.3).

The MAXIMUM NUMBER OF ZONE PERMISSION DESCRIPTORS field specifies the maximum number of complete zone permission descriptors that the management device server shall return.

The CRC field is defined in 9.4.3.2.7.

9.4.3.7.3 REPORT ZONE PERMISSION TABLE response

Table 215 defines the response format.

Table 215 — REPORT ZONE PERMISSION TABLE response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (04h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH ((n - 7) / 4)							
4	(MSB)							
5	EXPANDER CHANGE COUNT (LSB)							
6	ZONE LOCKED	Reserved					REPORT TYPE	
7	NUMBER OF ZONE GROUPS		Reserved					
8	Reserved							
12								
13	ZONE PERMISSION DESCRIPTOR LENGTH							
14	STARTING SOURCE ZONE GROUP							
15	NUMBER OF ZONE PERMISSION DESCRIPTORS							
Zone permission descriptor list								
16	Zone permission descriptor (first) (see table 217 or table 218 in 9.4.3.7.4)							
31 or 47								
...	...							
(n - 20) or (n - 36)	Zone permission descriptor (last) (see table 217 or table 218 in 9.4.3.7.4)							
n - 4								
n - 3	(MSB)							
n	CRC (LSB)							

The SMP FRAME TYPE field is defined in 9.4.3.3.2 and shall be set to the value defined in table 215.

The FUNCTION field is defined in 9.4.3.3.3 and shall be set to the value defined in table 215.

The FUNCTION RESULT field is defined in 9.4.3.3.4.

The RESPONSE LENGTH field is defined in 9.4.3.3.5 and shall be set to one of the values defined in table 215 based on the ALLOCATED RESPONSE LENGTH field in the request frame.

The EXPANDER CHANGE COUNT field is defined in the SMP REPORT GENERAL response (see 9.4.3.4). If the SMP initiator port detects a change in the value of this field while retrieving multiple response frames, then it should retrieve the response frames again because the status information returned is incomplete and inconsistent.

The ZONE LOCKED bit is defined in the SMP REPORT GENERAL response.

The REPORT TYPE field indicates the value of the REPORT TYPE field in the request frame.

The NUMBER OF ZONE GROUPS field indicates the number of zone groups supported by the expander device and is defined in the REPORT GENERAL response (see table 206 in 9.4.3.4).

The ZONE PERMISSION DESCRIPTOR LENGTH field indicates the length, in dwords, of the zone permission descriptor (see 9.4.3.7.4).

The STARTING SOURCE ZONE GROUP field indicates the first source zone group (i.e., *s*) being returned, and shall be set to the same value as the STARTING SOURCE ZONE GROUP field in the SMP request frame.

The NUMBER OF ZONE PERMISSION DESCRIPTORS field indicates the number of zone permission descriptors in the zone permission descriptor list.

The zone permission descriptor list contains a zone permission descriptor as defined in 9.4.3.7.4 for each source zone group in ascending order starting with the source zone group specified in the STARTING SOURCE ZONE GROUP field in the request.

The CRC field is defined in 9.4.3.3.7.

9.4.3.7.4 Zone permission descriptor

The zone permission descriptor format is based on the NUMBER OF ZONE GROUPS field as defined in table 216.

Table 216 — Zone permission descriptors

NUMBER OF ZONE GROUPS field	Zone permission descriptor format
00b	Table 217
01b	Table 218
All others	Reserved

Table 217 defines the zone permission descriptor containing 128 zone groups.

Table 217 — Zone permission descriptor for a source zone group (i.e., *s*) with 128 zone groups

Byte\Bit	7	6	5	4	3	2	1	0
0	ZP[s, 127]	ZP[s, 126]	ZP[s, 125]	ZP[s, 124]	ZP[s, 123]	ZP[s, 122]	ZP[s, 121]	ZP[s, 120]
...	...							
15	ZP[s, 7] (0b)	ZP[s, 6] (0b)	ZP[s, 5] (0b)	ZP[s, 4] (0b)	ZP[s, 3]	ZP[s, 2]	ZP[s, 1] (1b)	ZP[s, 0] (0b)

Table 218 defines the zone permission descriptor containing 256 zone groups.

Table 218 — Zone permission descriptor for a source zone group (i.e., s) with 256 zone groups

Byte\Bit	7	6	5	4	3	2	1	0
0	ZP[s, 255]	ZP[s, 254]	ZP[s, 253]	ZP[s, 252]	ZP[s, 251]	ZP[s, 250]	ZP[s, 249]	ZP[s, 248]
...	...							
31	ZP[s, 7] (0b)	ZP[s, 6] (0b)	ZP[s, 5] (0b)	ZP[s, 4] (0b)	ZP[s, 3]	ZP[s, 2]	ZP[s, 1] (1b)	ZP[s, 0] (0b)

The zone permission descriptor contains all of the zone permission table entries for the source zone group (i.e., s).

Table 219 defines how the zone permission descriptor bits shall be set by the management device server.

Table 219 — Zone permission descriptor bit requirements

Source zone group (i.e., s)	Management device server requirement(s) ^a
0	ZP[s, 0] shall be set to zero. ZP[s, 1] shall be set to one. ZP[s, 2 through (z-1)] shall be set to zero.
1	ZP[s, 0 through (z-1)] shall be set to one.
4, 5, 6 or 7	ZP[s, 0] shall be set to zero. ZP[s, 1] shall be set to one. ZP[s, 4 through (z-1)] shall be set to zero.
2, 3, or 8 through (z-1) ^a	ZP[s, 0] shall be set to zero. ZP[s, 1] shall be set to one. ZP[s, 2 through 3] shall be set to zero or one as specified by the CONFIGURE ZONE PERMISSION TABLE function (see 9.4.3.26). ZP[s, 4 through 7] shall be set to zero. ZP[s, 8 through (z-1)] shall be set to zero or one as specified by the CONFIGURE ZONE PERMISSION TABLE function.
^a The number of zone groups (i.e., z) is reported in NUMBER OF ZONE GROUPS field.	

9.4.3.8 REPORT ZONE MANAGER PASSWORD function

The REPORT ZONE MANAGER PASSWORD function returns the zone manager password (see 4.9.1). This SMP function may be implemented by a management device server in a zoning expander device and shall be implemented if the management device server supports the CONFIGURE ZONE MANAGER PASSWORD function (see 9.4.3.24). Other management device servers shall not support this SMP function. This function shall only be processed if:

- a) the request is received from an SMP initiator port that has access to zone group 2 (see 4.9.3.2); or
- b) the request is received from any SMP initiator port while physical presence is asserted.

If physical presence is not asserted and the SMP initiator port does not have access to zone group 2, then the management device server shall return a function result of NO MANAGEMENT ACCESS RIGHTS in the response frame (see table 202 in 9.4.3.3).

Table 220 defines the request format.

Table 220 — REPORT ZONE MANAGER PASSWORD request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (05h)							
2	ALLOCATED RESPONSE LENGTH							
3	REQUEST LENGTH (01h)							
4	Reserved						REPORT TYPE	
5	Reserved							
7								
8	(MSB)	CRC						(LSB)
11								

The SMP FRAME TYPE field is defined in 9.4.3.2.2 and shall be set to the value defined in table 220.

The FUNCTION field is defined in 9.4.3.2.3 and shall be set to the value defined in table 220.

The ALLOCATED RESPONSE LENGTH field is defined in 9.4.3.2.4.

The REQUEST LENGTH field is defined in 9.4.3.2.5 and shall be set to the value defined in table 220.

The REPORT TYPE field specifies the zone manager password value that the management device server shall return and is defined in table 221.

Table 221 — REPORT TYPE field

Code	Description
00b	Current zone manager password
01b	Reserved ^a
10b	Saved zone manager password. If the expander device does not support saving, it shall return a function result of SAVING NOT SUPPORTED in the response frame (see table 202 in 9.4.3.3).
11b	Default zone manager password
^a The CONFIGURE ZONE PASSWORD function updates the current zone manager password, not a shadow zone manager password.	

The CRC field is defined in 9.4.3.2.7.

Table 222 defines the response format.

Table 222 — REPORT ZONE MANAGER PASSWORD response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (05h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH (09h)							
4	(MSB)	EXPANDER CHANGE COUNT						
5								
6	Reserved						REPORT TYPE	
7	Reserved							
8	ZONE MANAGER PASSWORD							
39								
40	(MSB)	CRC						
43								

The SMP FRAME TYPE field is defined in 9.4.3.3.2 and shall be set to the value defined in table 222.

The FUNCTION field is defined in 9.4.3.3.3 and shall be set to the value defined in table 222.

The FUNCTION RESULT field is defined in 9.4.3.3.4.

The RESPONSE LENGTH field is defined in 9.4.3.3.5 and shall be set to the value defined in table 222. A RESPONSE LENGTH field set to 00h does not have a special meaning based on the ALLOCATED RESPONSE LENGTH field in the request frame.

The EXPANDER CHANGE COUNT field is defined in the SMP REPORT GENERAL response (see 9.4.3.4).

The REPORT TYPE field indicates the value of the REPORT TYPE field in the request frame.

The ZONE MANAGER PASSWORD field indicates the zone manager password of the type indicated by the REPORT TYPE field.

The CRC field is defined in 9.4.3.3.7.

9.4.3.9 REPORT BROADCAST function

9.4.3.9.1 REPORT BROADCAST function overview

The REPORT BROADCAST function returns information about Broadcasts (see 4.1.13) that were either originated from this expander device or SAS device, or received on a phy directly attached to an end device.

This SMP function may be implemented by any management device server.

9.4.3.9.2 REPORT BROADCAST request

Table 223 defines the request format.

Table 223 — REPORT BROADCAST request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (06h)							
2	ALLOCATED RESPONSE LENGTH							
3	REQUEST LENGTH (01h)							
4	Reserved				BROADCAST TYPE			
5	Reserved							
7								
8	(MSB)							
11	(LSB)							

The SMP FRAME TYPE field is defined in 9.4.3.2.2 and shall be set to the value defined in table 223.

The FUNCTION field is defined in 9.4.3.2.3 and shall be set to the value defined in table 223.

The ALLOCATED RESPONSE LENGTH field is defined in 9.4.3.2.4.

The REQUEST LENGTH field is defined in 9.4.3.2.5 and shall be set to the value defined in table 223.

The BROADCAST TYPE field, defined in the ZONED BROADCAST request (see table 268 in 9.4.3.20), specifies the type of Broadcast for which counts shall be returned in the response frame.

The CRC field is defined in 9.4.3.2.7.

9.4.3.9.3 REPORT BROADCAST response

Table 224 defines the response format.

Table 224 — REPORT BROADCAST response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (06h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH ((n - 7) / 4)							
4	(MSB)	EXPANDER CHANGE COUNT						
5								(LSB)
6	Reserved				BROADCAST TYPE			
7	Reserved							
9								
10	BROADCAST DESCRIPTOR LENGTH							
11	NUMBER OF BROADCAST DESCRIPTORS							
Broadcast descriptor list								
12	Broadcast descriptor (first) (see table 225 in 9.4.3.9.4)							
19								
...	...							
n - 11	Broadcast descriptor (last) (see table 225 in 9.4.3.9.4)							
n - 4								
n - 3	(MSB)	CRC						
n								(LSB)

The SMP FRAME TYPE field is defined in 9.4.3.3.2 and shall be set to the value defined in table 224.

The FUNCTION field is defined in 9.4.3.3.3 and shall be set to the value defined in table 224.

The FUNCTION RESULT field is defined in 9.4.3.3.4.

The RESPONSE LENGTH field is defined in 9.4.3.3.5 and shall be set to the value defined in table 224. A RESPONSE LENGTH field set to 00h does not have a special meaning based on the ALLOCATED RESPONSE LENGTH field in the request frame.

The EXPANDER CHANGE COUNT field is defined in the SMP REPORT GENERAL response (see 9.4.3.4).

The BROADCAST TYPE field indicates the value of the BROADCAST TYPE field in the request frame.

The BROADCAST DESCRIPTOR LENGTH field indicates the length, in dwords, of the Broadcast descriptor (see 9.4.3.9.4).

The NUMBER OF BROADCAST DESCRIPTORS field indicates the number of Broadcast descriptors in the Broadcast descriptor list.