INTERNATIONAL STANDARD

ISO/IEC 14776-150

First edition
2004-11

Information technology –
Small computer system interface (SCSI) –

Part 150:
Serial Attached SCSI (SAS)



Reference number
ISO/IEC 14776-150:2004(E)

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

--`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

INTERNATIONAL STANDARD

ISO/IEC 14776-150

First edition
2004-11

Information technology – Small computer system interface (SCSI) –

Part 150: Serial Attached SCSI (SAS)

© ISO/IEC 2004

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 the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland



PRICE CODE XK

For price, see current catalogue

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

CONTENTS

Foreword 25
Introduction 26
1 Scope 29

2 References 29
2.1 Normative references 29
2.2 References under development 29
2.3 Bibliography 29

3 Definitions, symbols, abbreviations, keywords, and conventions 30
3.1 Definitions 30
3.2 Symbols and abbreviations 39
3.3 Keywords 41
3.4 Editorial conventions 42
3.5 Object and class diagram conventions 43
3.6 State machine conventions 46
3.6.1 State machine conventions overview 46
3.6.2 Transitions 46
3.6.3 Messages, requests, indications, confirmations, responses, and event notifications 47
3.7 Bit and byte ordering 47
3.8 Notation for procedures and functions 48

4 General 49
4.1 Architecture 49
4.1.1 Architecture overview 49
4.1.2 Physical links and phys 50
4.1.3 Ports (narrow ports and wide ports) 52
4.1.4 SAS devices 54
4.1.5 Expander devices (edge expander devices and fanout expander devices) 55
4.1.6 Service delivery subsystem 57
4.1.7 Domains 57
4.1.8 Expander device topologies 59
4.1.8.1 Expander device topology overview 59
4.1.8.2 Edge expander device set 59
4.1.8.3 Expander device topologies 60
4.1.9 Pathways 63
4.1.10 Connections 64
4.2 Names and identifiers 66
4.2.1 Names and identifiers overview 66
4.2.2 SAS addresses 67
4.2.3 Hashed SAS address 68
4.2.4 Device names 68
4.2.5 Port names 68
4.2.6 Port identifiers 68
4.2.7 Phy identifiers 69
4.3 State machines 69
4.3.1 State machine overview 69
4.3.2 Transmit data path 71
4.3.3 State machines and SAS device, SAS port, and SAS phy objects 76
4.4 Resets 77
4.4.1 Reset overview 77
4.4.2 Hard reset 79
4.5 I_T nexus loss 79
4.6 Expander device model 79
4.6.1 Expander device model overview 79
4.6.2 Expander ports 80

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4.6.3 Expander connection manager (ECM) ... 81
4.6.4 Expander connection router (ECR) ... 81
4.6.5 Broadcast primitive processor (BPP) ... 81
4.6.6 Expander device interfaces ... 81
4.6.6.1 Expander device interface overview ... 81
4.6.6.2 Expander device interfaces detail ... 83
4.6.6.3 ECM interface ... 83
4.6.6.4 ECR interface ... 85
4.6.6.5 BPP interface ... 86
4.6.7 Expander device routing ... 86
4.6.7.1 Routing attributes and routing methods ... 86
4.6.7.2 Connection request routing ... 87
4.6.7.3 Expander route table ... 87
4.6.7.4 Discover process ... 88
4.6.7.5 Expander route index order ... 90

5 Physical layer ... 97
5.1 Physical layer overview ... 97
5.2 Passive interconnect ... 97
5.2.1 SATA cables and connectors ... 97
5.2.2 SAS cables and connectors ... 97
5.2.3 Connectors ... 100
5.2.3.1 Connectors overview ... 100
5.2.3.2 SAS plug connector ... 100
5.2.3.3 SAS internal cable receptacle connector ... 100
5.2.3.4 SAS backplane receptacle connector ... 101
5.2.3.5 SAS internal connector pin assignments ... 102
5.2.3.6 SAS external cable plug connector ... 103
5.2.3.7 SAS external receptacle connector ... 103
5.2.3.8 SAS external connector pin assignments ... 104
5.2.4 Cables ... 104
5.2.4.1 SAS internal cables ... 104
5.2.4.2 SAS external cables ... 107
5.2.5 Backplanes ... 107
5.3 Transmitter and receiver electrical characteristics ... 107
5.3.1 Compliance points ... 107
5.3.2 General interface specification ... 107
5.3.3 Eye masks ... 109
5.3.3.1 Eye masks overview ... 109
5.3.3.2 Receive eye mask at IR, CR, and XR ... 110
5.3.3.3 Jitter tolerance masks ... 110
5.3.4 Signal characteristics at IT, CT, and XT ... 111
5.3.5 Signal characteristics at IR, CR, and XR ... 113
5.3.6 Jitter ... 115
5.3.7 Receiver jitter tolerance ... 116
5.3.8 Compliant jitter test pattern (CJTPAT) ... 116
5.3.9 Impedance specifications ... 116
5.3.10 Electrical TxRx connections ... 117
5.3.11 Transmitter characteristics ... 118
5.3.12 Receiver characteristics ... 120
5.3.13 Spread spectrum clocking ... 121
5.3.14 Non-tracking clock architecture ... 121
5.4 READY LED signal electrical characteristics ... 121

6 Phy layer ... 122
6.1 Phy layer overview ... 122
6.2 Encoding (8b10b) ... 122
6.2.1 Encoding overview ... 122

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.2.2 8b10b coding introduction 122
6.2.3 8b10b coding notation conventions 122
6.3 Character encoding and decoding 123
6.3.1 Introduction 123
6.3.2 Transmission order 124
6.3.3 Valid and invalid transmission characters 124
6.3.3.1 Definitions 124
6.3.3.2 Generating transmission characters 128
6.3.3.3 Validity of received transmission characters 128
6.4 Bit order 128
6.5 Out of band (OOB) signals 130
6.6 Phy reset sequences 134
6.6.1 Phy reset sequences overview 134
6.6.2 SATA phy reset sequence 135
6.6.2.1 SATA OOB sequence 135
6.6.2.2 SATA speed negotiation sequence 136
6.6.3 SAS to SATA phy reset sequence 136
6.6.4 SAS to SAS phy reset sequence 137
6.6.4.1 SAS OOB sequence 137
6.6.4.2 SAS speed negotiation sequence 139
6.6.5 Phy reset sequence after devices are attached 141
6.7 SP (phy layer) state machine 142
6.7.1 SP state machine overview 142
6.7.2 SP transmitter and receiver 143
6.7.3 OOB sequence states 145
6.7.3.1 OOB sequence states overview 145
6.7.3.2 SP0:OOB_COMINIT state 146
6.7.3.2.1 State description 146
6.7.3.2.2 Transition SP0:OOB_COMINIT to SP1:OOB_AwaitCOMX 146
6.7.3.2.3 Transition SP0:OOB_COMINIT to SP3:OOB_AwaitCOMINIT_Sent 146
6.7.3.2.4 Transition SP0:OOB_COMINIT to SP4:OOB_COMSAS 146
6.7.3.3 SP1:OOB_AwaitCOMX state 146
6.7.3.3.1 State description 146
6.7.3.3.2 Transition SP1:OOB_AwaitCOMX to SP0:OOB_COMINIT 146
6.7.3.3.3 Transition SP1:OOB_AwaitCOMX to SP4:OOB_COMSAS 146
6.7.3.4 SP2:OOB_NoCOMSASTimeout state 146
6.7.3.4.1 State description 146
6.7.3.4.2 Transition SP2:OOB_NoCOMSASTimeout to SP0:OOB_COMINIT 146
6.7.3.4.3 Transition SP2:OOB_NoCOMSASTimeout to SP4:OOB_COMSAS 146
6.7.3.5 SP3:OOB_AwaitCOMINIT_Sent state 147
6.7.3.5.1 State description 147
6.7.3.5.2 Transition SP3:OOB_AwaitCOMINIT_Sent to SP4:OOB_COMSAS 147
6.7.3.6 SP4:OOB_COMSAS state 147
6.7.3.6.1 State description 147
6.7.3.6.2 Transition SP4:OOB_COMSAS to SP5:OOB_AwaitCOMSAS_Sent 147
6.7.3.6.3 Transition SP4:OOB_COMSAS to SP6:OOB_AwaitNoCOMSAS 147
6.7.3.6.4 Transition SP4:OOB_COMSAS to SP7:OOB_AwaitCOMSAS 147
6.7.3.7 SP5:OOB_AwaitCOMSAS_Sent state 147
6.7.3.7.1 State description 147
6.7.3.7.2 Transition SP5:OOB_AwaitCOMSAS_Sent to SP6:OOB_AwaitNoCOMSAS 147
6.7.3.8 SP6:OOB_AwaitNoCOMSAS state 147
6.7.3.8.1 State description 147
6.7.3.8.2 Transition SP6:OOB_AwaitNoCOMSAS to SP8:SAS_Start 147
6.7.3.9 SP7:OOB_AwaitCOMSAS state 147
6.7.3.9.1 State description 147
6.7.3.9.2 Transition SP7:OOB_AwaitCOMSAS to SP0:OOB_COMINIT 148
6.7.3.9.3 Transition SP7:OOB_AwaitCOMSAS to SP6:OOB_AwaitNoCOMSAS 148
6.7.3.9.4 Transition SP7:OOB_AwaitCOMSAS to SP16:SATA_COMWAKE 148

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.3.9.5 Transition SP7:OOB_AwaitCOMSAS to SP2:OOB_NoCOMSASTimeout 148
6.7.4 SAS speed negotiation states 148
6.7.4.1 SAS speed negotiation states overview 148
6.7.4.2 SP8:SAS_Start state 150
6.7.4.2.1 State description 150
6.7.4.2.2 Transition SP8:SAS_Start to SP10:SAS_AwaitALIGN 150
6.7.4.2.3 Transition SP8:SAS_Start to SP9:SAS_RateNotSupported 150
6.7.4.3 SP9:SAS_RateNotSupported state 150
6.7.4.3.1 State description 150
6.7.4.3.2 Transition SP9:SAS_RateNotSupported to SP14:SAS_Fail 150
6.7.4.4 SP10:SAS_AwaitALIGN state 150
6.7.4.4.1 State description 150
6.7.4.4.2 Transition SP10:SAS_AwaitALIGN to SP0:OOB_COMINIT 150
6.7.4.4.3 Transition SP10:SAS_AwaitALIGN to SP11:SAS_AwaitALIGN1 150
6.7.4.4.4 Transition SP10:SAS_AwaitALIGN to SP12:SAS_AwaitSNW 151
6.7.4.4.5 Transition SP10:SAS_AwaitALIGN to SP14:SAS_Fail 151
6.7.4.5 SP11:SAS_AwaitALIGN1 state 151
6.7.4.5.1 State description 151
6.7.4.5.2 Transition SP11:SAS_AwaitALIGN1 to SP0:OOB_COMINIT 151
6.7.4.5.3 Transition SP11:SAS_AwaitALIGN1 to SP14:SAS_Fail 151
6.7.4.5.4 Transition SP11:SAS_AwaitALIGN1 to SP12:SAS_AwaitSNW 151
6.7.4.6 SP12:SAS_AwaitSNW state 151
6.7.4.6.1 State description 151
6.7.4.6.2 Transition SP12:SAS_AwaitSNW to SP0:OOB_COMINIT 151
6.7.4.6.3 Transition SP12:SAS_AwaitSNW to SP13:SAS_Pass 151
6.7.4.7 SP13:SAS_Pass state 151
6.7.4.7.1 State description 151
6.7.4.7.2 Transition SP13:SAS_Pass to SP0:OOB_COMINIT 152
6.7.4.7.3 Transition SP13:SAS_Pass to SP8:SAS_Start 152
6.7.4.7.4 Transition SP13:SAS_Pass to SP15:SAS_PHY_Ready 152
6.7.4.8 SP14:SAS_Fail state 152
6.7.4.8.1 State description 152
6.7.4.8.2 Transition SP14:SAS_Fail to SP1:OOB_AwaitCOMX 152
6.7.4.8.3 Transition SP14:SAS_Fail to SP8:SAS_Start 152
6.7.4.9 SP15:SAS_PHY_Ready state 153
6.7.4.9.1 State description 153
6.7.4.9.2 Transition SP15:SAS_PHY_Ready to SP0:OOB_COMINIT 153
6.7.5 SATA host emulation states 153
6.7.5.1 SATA host emulation states overview 153
6.7.5.2 SP16:SATA_COMWAKE state 154
6.7.5.2.1 State description 154
6.7.5.2.2 Transition SP16:SATA_COMWAKE to SP17:SATA_AwaitCOMWAKE 155
6.7.5.3 SP17:SATA_AwaitCOMWAKE state 155
6.7.5.3.1 State description 155
6.7.5.3.2 Transition SP17:SATA_AwaitCOMWAKE to SP18:SATA_AwaitNoCOMWAKE 155
6.7.5.4 SP18:SATA_AwaitNoCOMWAKE state 155
6.7.5.4.1 State description 155
6.7.5.4.2 Transition SP18:SATA_AwaitNoCOMWAKE to SP19:SATA_AwaitALIGN 155
6.7.5.5 SP19:SATA_AwaitALIGN state 155
6.7.5.5.1 State description 155
6.7.5.5.2 Transition SP19:SATA_AwaitALIGN to SP20:SATA_AdjustSpeed 155
6.7.5.5.3 Transition SP19:SATA_AwaitALIGN to SP0:OOB_COMINIT 155
6.7.5.6 SP20:SATA_AdjustSpeed state 155
6.7.5.6.1 State description 155
6.7.5.6.2 Transition SP20:SATA_AdjustSpeed to SP0:OOB_COMINIT 155
6.7.5.6.3 Transition SP20:SATA_AdjustSpeed to SP21:SATA_TransmitALIGN 156
6.7.5.7 SP21:SATA_TransmitALIGN state 156
6.7.5.7.1 State description 156

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.7.5.7.2 Transition SP21:SATA_TransmitALIGN to SP0:OOB_COMINIT 156
6.7.5.7.3 Transition SP21:SATA_TransmitALIGN to SP22:SATA_PHY_Ready 156
6.7.5.8 SP22:SATA_PHY_Ready state 156
6.7.5.8.1 State description 156
6.7.5.8.2 Transition SP22:SATA_PHY_Ready to SP1:OOB_COMINIT 156
6.7.5.8.3 Transition SP22:SATA_PHY_Ready to SP23:SATA_PM_Partial 156
6.7.5.8.4 Transition SP22:SATA_PHY_Ready to SP24:SATA_PM_Slumber 156
6.7.5.9 SP23:SATA_PM_Partial state 156
6.7.5.9.1 State description 156
6.7.5.9.2 Transition SP23:SATA_PM_Partial to SP16:SATA_COMWAKE 156
6.7.5.9.3 Transition SP23:SATA_PM_Partial to SP18:SATA_AwaitNoCOMWAKE 156
6.7.5.10 SP24:SATA_PM_Slumber state 157
6.7.5.10.1 State description 157
6.7.5.10.2 Transition SP24:SATA_PM_Slumber to SP16:SATA_COMWAKE 157
6.7.5.10.3 Transition SP24:SATA_PM_Slumber to SP18:SATA_AwaitNoCOMWAKE 157
6.8 SP_DWS (phy layer dword synchronization) state machine 157
6.8.1 SP_DWS state machine overview 157
6.8.2 SP_DWS receiver 159
6.8.3 SP_DWS0:AcquireSync state 159
6.8.3.1 State description 159
6.8.3.2 Transition SP_DWS0:AcquireSync to SP_DWS1:Valid1 159
6.8.4 SP_DWS1:Valid1 state 159
6.8.4.1 State description 159
6.8.4.2 Transition SP_DWS1:Valid1 to SP_DWS0:AcquireSync 159
6.8.4.3 Transition SP_DWS1:Valid1 to SP_DWS2:Valid2 160
6.8.5 SP_DWS2:Valid2 state 160
6.8.5.1 State description 160
6.8.5.2 Transition SP_DWS2:Valid2 to SP_DWS0:AcquireSync 160
6.8.5.3 Transition SP_DWS2:Valid2 to SP_DWS3:SyncAcquired 160
6.8.6 SP_DWS3:SyncAcquired state 160
6.8.6.1 State description 160
6.8.6.2 Transition SP_DWS3:SyncAcquired to SP_DWS4:Lost1 160
6.8.7 SP_DWS4:Lost1 state 160
6.8.7.1 State description 160
6.8.7.2 Transition SP_DWS4:Lost1 to SP_DWS5:Lost1Recovered 160
6.8.7.3 Transition SP_DWS4:Lost1 to SP_DWS6:Lost2 160
6.8.8 SP_DWS5:Lost1Recovered state 160
6.8.8.1 State description 160
6.8.8.2 Transition SP_DWS5:Lost1Recovered to SP_DWS3:SyncAcquired 161
6.8.8.3 Transition SP_DWS5:Lost1Recovered to SP_DWS6:Lost2 161
6.8.9 SP_DWS6:Lost2 state 161
6.8.9.1 State description 161
6.8.9.2 Transition SP_DWS6:Lost2 to SP_DWS7:Lost2Recovered 161
6.8.9.3 Transition SP_DWS6:Lost2 to SP_DWS8:Lost3 161
6.8.10 SP_DWS7:Lost2Recovered state 161
6.8.10.1 State description 161
6.8.10.2 Transition SP_DWS7:Lost2Recovered to SP_DWS4:Lost1 161
6.8.10.3 Transition SP_DWS7:Lost2Recovered to SP_DWS8:Lost3 161
6.8.11 SP_DWS8:Lost3 state 161
6.8.11.1 State description 161
6.8.11.2 Transition SP_DWS8:Lost3 to SP_DWS9:Lost3Recovered 161
6.8.11.3 Transition SP_DWS8:Lost3 to SP_DWS0:AcquireSync 161
6.8.12 SP_DWS9:Lost3Recovered state 162
6.8.12.1 State description 162
6.8.12.2 Transition SP_DWS9:Lost3Recovered to SP_DWS6:Lost2 162
6.8.12.3 Transition SP_DWS9:Lost3Recovered to SP_DWS0:AcquireSync 162
6.9 Spin-up 162

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7 Link layer 163
7.1 Link layer overview 163
7.2 Primitives 163
7.2.1 Primitives overview 163
7.2.2 Primitive summary 164
7.2.3 Primitive encodings 168
7.2.4 Primitive sequences 172
7.2.4.1 Primitive sequences overview 172
7.2.4.2 Single primitive sequence 172
7.2.4.3 Repeated primitive sequence 172
7.2.4.4 Triple primitive sequence 172
7.2.4.5 Redundant primitive sequence 173
7.2.5 Primitives not specific to type of connections 174
7.2.5.1 AIP (Arbitration in progress) 174
7.2.5.2 ALIGN 175
7.2.5.3 BREAK 176
7.2.5.4 BROADCAST 176
7.2.5.5 CLOSE 176
7.2.5.6 EOAF (End of address frame) 177
7.2.5.7 ERROR 177
7.2.5.8 HARD_RESET 177
7.2.5.9 NOTIFY 177
7.2.5.10 OPEN_ACCEPT 178
7.2.5.11 OPEN_REJECT 178
7.2.5.12 SOAF (Start of address frame) 181
7.2.6 Primitives used only inside SSP and SMP connections 181
7.2.6.1 ACK (Acknowledge) 181
7.2.6.2 CREDIT_BLOCKED 181
7.2.6.3 DONE 181
7.2.6.4 EOF (End of frame) 182
7.2.6.5 NAK (Negative acknowledgement) 182
7.2.6.6 RRDY (Receiver ready) 182
7.2.6.7 SOF (Start of frame) 182
7.2.7 Primitives used only inside STP connections and on SATA physical links 182
7.2.7.1 SATA_ERROR 182
7.2.7.2 SATA_PMACK, SATA_PMNAK, SATA_PMREQ_P, and SATA_PMREQ_S (Power management acknowledgements and requests) 182
7.2.7.3 SATA_HOLD and SATA_HOLDA (Hold and hold acknowledge) 183
7.2.7.4 SATA_R_RDY and SATA_X_RDY (Receiver ready and transmitter ready) 183
7.2.7.5 Other primitives used inside STP connections and on SATA physical links 183
7.3 Clock skew management 183
7.4 Idle physical links 184
7.5 CRC 185
7.5.1 CRC overview 185
7.5.2 CRC generation 185
7.5.3 CRC checking 187
7.6 Scrambling 187
7.7 Bit order of CRC and scrambler 189
7.8 Address frames 192
7.8.1 Address frames overview 192
7.8.2 IDENTIFY address frame 194
7.8.3 OPEN address frame 196
7.9 Identification and hard reset sequence 198
7.9.1 Identification and hard reset sequence overview 198
7.9.2 SAS initiator device rules 199
7.9.3 Fanout expander device rules 199
7.9.4 Edge expander device rules 199
7.9.5 SL_IR (link layer identification and hard reset) state machines 199

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.9.5.1 SL_IR state machines overview 199
7.9.5.2 SL_IR transmitter and receiver 202
7.9.5.3 SL_IR_TIR (transmit IDENTIFY or HARD_RESET) state machine 202
7.9.5.3.1 SL_IR_TIR state machine overview 202
7.9.5.3.2 SL_IR_TIR1:Idle state 202
7.9.5.3.2.1 State description 202
7.9.5.3.2.2 Transition SL_IR_TIR1:Idle to SL_IR_TIR2:Transmit_Identify 202
7.9.5.3.2.3 Transition SL_IR_TIR1:Idle to SL_IR_TIR3:Transmit_Hard_Reset 202
7.9.5.3.3 SL_IR_TIR2:Transmit_Identify state 202
7.9.5.3.3.1 State description 202
7.9.5.3.3.2 Transition SL_IR_TIR2:Transmit_Identify to SL_IR_TIR4:Completed 203
7.9.5.3.4 SL_IR_TIR3:Transmit_Hard_Reset state 203
7.9.5.3.4.1 State description 203
7.9.5.3.4.2 Transition SL_IR_TIR3:Transmit_Hard_Reset to SL_IR_TIR4:Completed 203
7.9.5.3.5 SL_IR_TIR4:Completed state 203
7.9.5.4 SL_IR_RIF (receive IDENTIFY address frame) state machine 203
7.9.5.4.1 SL_IR_RIF state machine overview 203
7.9.5.4.2 SL_IR_RIF1:Idle state 203
7.9.5.4.2.1 State description 203
7.9.5.4.2.2 Transition SL_IR_RIF1:Idle to SL_IR_RIF2:Receive_Identify_Frame 203
7.9.5.4.3 SL_IR_RIF2:Receive_Identify_Frame state 203
7.9.5.4.3.1 State description 203
7.9.5.4.3.2 Transition SL_IR_RIF2:Receive_Identify_Frame to SL_IR_RIF3:Completed 204
7.9.5.4.4 SL_IR_RIF3:Completed state 204
7.9.5.5 SL_IR_IRC (identification and hard reset control) state machine 204
7.9.5.5.1 SL_IR_IRC state machine overview 204
7.9.5.5.2 SL_IR_IRC1:Idle state 204
7.9.5.5.2.1 State description 204
7.9.5.5.2.2 Transition SL_IR_IRC1:Idle to SL_IR_IRC2:Wait 204
7.9.5.5.3 SL_IR_IRC2:Wait state 205
7.9.5.5.3.1 State description 205
7.9.5.5.3.2 Transition SL_IR_IRC2:Wait to SL_IR_IRC3:Completed 205
7.9.5.5.4 SL_IR_IRC3:Completed state 205
7.10 Power management 205
7.11 SAS domain changes 206
7.12 Connections 206
7.12.1 Connections overview 206
7.12.2 Opening a connection 206
7.12.2.1 Connection request 206
7.12.2.2 Connection responses 207
7.12.3 Arbitration fairness 207
7.12.4 Arbitration and resource management in an expander device 208
7.12.4.1 Arbitration overview 208
7.12.4.2 Arbitration status 209
7.12.4.3 Partial Pathway Timeout timer 209
7.12.4.4 Pathway recovery 210
7.12.5 Expander devices and connection requests 210
7.12.5.1 All expander devices 210
7.12.5.2 Edge expander devices 210
7.12.5.3 Fanout expander devices 210
7.12.6 Aborting a connection request 211
7.12.7 Closing a connection 213
7.12.8 Breaking a connection 214
7.13 Rate matching 214
7.14 SL (link layer for SAS phys) state machines 216
7.14.1 SL state machines overview 216
7.14.2 SL transmitter and receiver 218
7.14.3 SL_RA (receive OPEN address frame) state machine 219

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.14.4 SL_CC (connection control) state machine 220
7.14.4.1 SL_CC state machine overview 220
7.14.4.2 SL_CC0:Idle state 220
7.14.4.2.1 State description 220
7.14.4.2.2 Transition SL_CC0:Idle to SL_CC1:ArbSel 221
7.14.4.2.3 Transition SL_CC0:Idle to SL_CC2:Selected 221
7.14.4.3 SL_CC1:ArbSel state 221
7.14.4.3.1 State description 221
7.14.4.3.2 Transition SL_CC1:ArbSel to SL_CC0:Idle 222
7.14.4.3.3 Transition SL_CC1:ArbSel to SL_CC2:Selected 222
7.14.4.3.4 Transition SL_CC1:ArbSel to SL_CC3:Connected 222
7.14.4.3.5 Transition SL_CC1:ArbSel to SL_CC5:BreakWait 222
7.14.4.3.6 Transition SL_CC1:ArbSel to SL_CC6:Break 222
7.14.4.4 SL_CC2:Selected state 223
7.14.4.4.1 State description 223
7.14.4.4.2 Transition SL_CC2:Selected to SL_CC0:Idle 223
7.14.4.4.3 Transition SL_CC2:Selected to SL_CC3:Connected 223
7.14.4.4.4 Transition SL_CC2:Selected to SL_CC6:Break 223
7.14.4.5 SL_CC3:Connected state 223
7.14.4.5.1 State description 223
7.14.4.5.2 Transition SL_CC3:Connected to SL_CC4:DisconnectWait 224
7.14.4.5.3 Transition SL_CC3:Connected to SL_CC5:BreakWait 224
7.14.4.5.4 Transition SL_CC3:Connected to SL_CC6:Break 224
7.14.4.5.5 Transition SL_CC3:Connected to SL_CC7:CloseSTP 224
7.14.4.6 SL_CC4:DisconnectWait state 224
7.14.4.6.1 State description 224
7.14.4.6.2 Transition SL_CC4:DisconnectWait to SL_CC0:Idle 224
7.14.4.6.3 Transition SL_CC4:DisconnectWait to SL_CC5:BreakWait 225
7.14.4.6.4 Transition SL_CC4:DisconnectWait to SL_CC6:Break 225
7.14.4.7 SL_CC5:BreakWait state 225
7.14.4.7.1 State description 225
7.14.4.7.2 Transition SL_CC5:BreakWait to SL_CC0:Idle 225
7.14.4.8 SL_CC6:Break state 225
7.14.4.8.1 State description 225
7.14.4.8.2 Transition SL_CC6:Break to SL_CC0:Idle 225
7.14.4.9 SL_CC7:CloseSTP state 225
7.14.4.9.1 State description 225
7.14.4.9.2 Transition SL_CC7:CloseSTP to SL_CC0:Idle 225
7.15 XL (link layer for expander phys) state machine 226
7.15.1 XL state machine overview 226
7.15.2 XL transmitter and receiver 230
7.15.3 XL0:Idle state 230
7.15.3.1 State description 230
7.15.3.2 Transition XL0:Idle to XL1:Request_Path 230
7.15.3.3 Transition XL0:Idle to XL5:Forward_Open 231
7.15.4 XL1:Request_Path state 231
7.15.4.1 State description 231
7.15.4.2 Transition XL1:Request_Path to XL2:Request_Open 231
7.15.4.3 Transition XL1:Request_Path to XL4:Open_Reject 231
7.15.4.4 Transition XL1:Request_Path to XL0:Idle 232
7.15.4.5 Transition XL1:Request_Path to XL9:Break 232
7.15.5 XL2:Request_Open state 232
7.15.5.1 State description 232
7.15.5.2 Transition XL2:Request_Open to XL3:Open_Confirm_Wait 232
7.15.6 XL3:Open_Confirm_Wait state 232
7.15.6.1 State description 232
7.15.6.2 Transition XL3:Open_Confirm_Wait to XL0:Idle 233
7.15.6.3 Transition XL3:Open_Confirm_Wait to XL1:Request_Path 233

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.15.6.4 Transition XL3:Open_Confirm_Wait to XL5:Forward_Open 233
7.15.6.5 Transition XL3:Open_Confirm_Wait to XL7:Connected 233
7.15.6.6 Transition XL3:Open_Confirm_Wait to XL9:Break 233
7.15.6.7 Transition XL3:Open_Confirm_Wait to XL10:Break_Wait 233
7.15.7 XL4:Open_Reject state 233
7.15.7.1 State description 233
7.15.7.2 Transition XL4:Open_Reject to XL0:Idle 233
7.15.8 XL5:Forward_Open state 233
7.15.8.1 State description 233
7.15.8.2 Transition XL5:Forward_Open to XL6:Open_Response_Wait 233
7.15.9 XL6:Open_Response_Wait state 234
7.15.9.1 State description 234
7.15.9.2 Transition XL6:Open_Response_Wait to XL0:Idle 235
7.15.9.3 Transition XL6:Open_Response_Wait to XL1:Request_Path 235
7.15.9.4 Transition XL6:Open_Response_Wait to XL2:Request_Open 235
7.15.9.5 Transition XL6:Open_Response_Wait to XL7:Connected 235
7.15.9.6 Transition XL6:Open_Response_Wait to XL9:Break 235
7.15.9.7 Transition XL6:Open_Response_Wait to XL10:Break_Wait 235
7.15.10 XL7:Connected state 235
7.15.10.1 State description 235
7.15.10.2 Transition XL7:Connected to XL8:Close_Wait 235
7.15.10.3 Transition XL7:Connected to XL9:Break 235
7.15.10.4 Transition XL7:Connected to XL10:Break_Wait 235
7.15.11 XL8:Close_Wait state 236
7.15.11.1 State description 236
7.15.11.2 Transition XL8:Close_Wait to XL0:Idle 236
7.15.11.3 Transition XL8:Close_Wait to XL9:Break 236
7.15.11.4 Transition XL8:Close_Wait to XL10:Break_Wait 236
7.15.12 XL9:Break state 236
7.15.12.1 State description 236
7.15.12.2 Transition XL9:Break to XL0:Idle 236
7.15.13 XL10:Break_Wait state 236
7.15.13.1 State description 236
7.15.13.2 Transition XL10:Break_Wait to XL0:Idle 236
7.16 SSP link layer 237
7.16.1 Opening an SSP connection 237
7.16.2 Full duplex 237
7.16.3 SSP frame transmission and reception 237
7.16.4 SSP flow control 237
7.16.5 Interlocked frames 238
7.16.6 Closing an SSP connection 239
7.16.7 SSP (link layer for SSP phys) state machines 240
7.16.7.1 SSP state machines overview 240
7.16.7.2 SSP transmitter and receiver 243
7.16.7.3 SSP_TIM (transmit interlocked frame monitor) state machine 244
7.16.7.4 SSP_TCM (transmit frame credit monitor) state machine 245
7.16.7.5 SSP_D (DONE control) state machine 245
7.16.7.6 SSP_TF (transmit frame control) state machine 246
7.16.7.6.1 SSP_TF state machine overview 246
7.16.7.6.2 SSP_TF1:Connected_Idle state 246
7.16.7.6.2.1 State description 246
7.16.7.6.2.2 Transition SSP_TF1:Connected_Idle to SSP_TF2:Tx_Wait 246
7.16.7.6.2.3 Transition SSP_TF1:Connected_Idle to SSP_TF4:Indicate_DONE_Tx 246
7.16.7.6.3 SSP_TF2:Tx_Wait state 247
7.16.7.6.3.1 State description 247
7.16.7.6.3.2 Transition SSP_TF2:Tx_Wait to SSP_TF3:Indicate_Frame_Tx 247
7.16.7.6.3.3 Transition SSP_TF2:Tx_Wait to SSP_TF4:Indicate_DONE_Tx 247
7.16.7.6.4 SSP_TF3:Indicate_Frame_Tx state 247

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.16.7.6.4.1 State description 247
7.16.7.6.4.2 Transition SSP_TF3:Indicate_Frame_Tx to SSP_TF1:Connected_Idle 247
7.16.7.6.5 SSP_TF4:Indicate_DONE_Tx state 248
7.16.7.7 SSP_RF (receive frame control) state machine 248
7.16.7.8 SSP_RCM (receive frame credit monitor) state machine 249
7.16.7.9 SSP_RIM (receive interlocked frame monitor) state machine 249
7.16.7.10 SSP_TC (transmit credit control) state machine 250
7.16.7.11 SSP_TAN (transmit ACK/NAK control) state machine 250
7.17 STP link layer 250
7.17.1 STP frame transmission and reception 250
7.17.2 STP flow control 250
7.17.3 Affiliations 253
7.17.4 Opening an STP connection 253
7.17.5 Closing an STP connection 253
7.17.6 STP connection management examples 254
7.17.7 STP (link layer for STP phys) state machines 257
7.17.8 SMP target port support 257
7.18 SMP link layer 257
7.18.1 SMP frame transmission and reception 257
7.18.2 SMP flow control 257
7.18.3 Closing an SMP connection 257
7.18.4 SMP (link layer for SMP phys) state machines 257
7.18.4.1 SMP state machines overview 257
7.18.4.2 SMP transmitter and receiver 258
7.18.4.3 SMP_IP (link layer for SMP initiator phys) state machine 258
7.18.4.3.1 SMP_IP state machine overview 258
7.18.4.3.2 SMP_IP1:Idle state 259
7.18.4.3.2.1 State description 259
7.18.4.3.2.2 Transition SMP_IP1:Idle to SMP_IP2:Transmit_Frame 259
7.18.4.3.3 SMP_IP2:Transmit_Frame state 260
7.18.4.3.3.1 State description 260
7.18.4.3.3.2 Transition SMP_IP2:Transmit_Frame to SMP_IP3:Receive_Frame 260
7.18.4.3.4 SMP_IP3:Receive_Frame state 260
7.18.4.4 SMP_TP (link layer for SMP target ports) state machine 260
7.18.4.4.1 SMP_TP state machine overview 260
7.18.4.4.2 SMP_TP1:Receive_Frame state 261
7.18.4.4.2.1 State description 261
7.18.4.4.2.2 Transition SMP_TP1:Receive_Frame to SMP_TP2:Transmit_Frame 261
7.18.4.4.3 SMP_TP2:Transmit_Frame state 262

8 Port layer 263
8.1 Port layer overview 263
8.2 PL (port layer) state machines 263
8.2.1 PL state machines overview 263
8.2.2 PL_OC (port layer overall control) state machine 265
8.2.2.1 PL_OC state machine overview 265
8.2.2.2 PL_OC1:Idle state 266
8.2.2.2.1 PL_OC1:Idle state description 266
8.2.2.2.2 Transition PL_OC1:Idle to PL_OC2:Overall_Control 267
8.2.2.3 PL_OC2:Overall_Control state 267
8.2.2.3.1 PL_OC2:Overall_Control state overview 267
8.2.2.3.2 PL_OC2:Overall_Control state establishing connections 267
8.2.2.3.3 PL_OC2:Overall_Control state connection established 270
8.2.2.3.4 PL_OC2:Overall_Control state unable to establish a connection 270
8.2.2.3.5 PL_OC2:Overall_Control state connection management 271
8.2.2.3.6 PL_OC2:Overall_Control state frame transmission 272
8.2.2.3.7 PL_OC2:Overall_Control state frame transmission cancellations 273
8.2.2.3.8 Transition PL_OC2:Overall_Control to PL_OC1:Idle 273

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

8.2.3 PL_PM (port layer phy manager) state machine 273
8.2.3.1 PL_PM state machine overview 273
8.2.3.2 PL_PM1:Idle state 276
8.2.3.2.1 PL_PM1:Idle state description 276
8.2.3.2.2 Transition PL_PM1:Idle to PL_PM2:Req_Wait 277
8.2.3.2.3 Transition PL_PM1:Idle to PL_PM3:Connected 277
8.2.3.3 PL_PM2:Req_Wait state 277
8.2.3.3.1 PL_PM2:Req_Wait state overview 277
8.2.3.3.2 PL_PM2:Req_Wait establishing a connection 277
8.2.3.3.3 PL_PM2:Req_Wait connection established 277
8.2.3.3.4 PL_PM2:Req_Wait unable to establish a connection 278
8.2.3.3.5 PL_PM2:Req_Wait connection management 278
8.2.3.3.6 Transition PL_PM2:Req_Wait to PL_PM1:Idle 278
8.2.3.3.7 Transition PL_PM2:Req_Wait to PL_PM3:Connected 279
8.2.3.3.8 Transition PL_PM2:Req_Wait to PL_PM4:Wait_For_Close 279
8.2.3.4 PL_PM3:Connected state 279
8.2.3.4.1 PL_PM3:Connected state description 279
8.2.3.4.2 Transition PL_PM3:Connected to PL_PM1:Idle 280
8.2.3.5 PL_PM4:Wait_For_Close state 281
8.2.3.5.1 PL_PM4:Wait_For_Close state description 281
8.2.3.5.2 Transition PL_PM4:Wait_For_Close to PL_PM1:Idle 281

9 Transport layer 282
9.1 Transport layer overview 282
9.2 SSP transport layer 283
9.2.1 SSP frame format 283
9.2.2 Information units 285
9.2.2.1 COMMAND information unit 285
9.2.2.2 TASK information unit 286
9.2.2.3 XFER_RDY information unit 288
9.2.2.4 DATA information unit 288
9.2.2.5 RESPONSE information unit 289
9.2.2.5.1 RESPONSE information unit overview 289
9.2.2.5.2 RESPONSE information unit NO_DATA format 291
9.2.2.5.3 RESPONSE information unit RESPONSE_DATA format 291
9.2.2.5.4 RESPONSE information unit SENSE_DATA format 292
9.2.3 Sequences of SSP frames 292
9.2.4 SSP transport layer handling of link layer errors 294
9.2.4.1 COMMAND frame 294
9.2.4.2 TASK frame 294
9.2.4.3 XFER_RDY frame 294
9.2.4.4 DATA frame 295
9.2.4.5 RESPONSE frame 295
9.2.5 SSP transport layer error handling 295
9.2.5.1 SSP target port error handling 295
9.2.5.2 SSP initiator port error handling 296
9.2.6 ST (transport layer for SSP ports) state machines 297
9.2.6.1 ST state machines overview 297
9.2.6.2 ST_I (transport layer for SSP initiator ports) state machines 297
9.2.6.2.1 ST_I state machines overview 297
9.2.6.2.2 ST_ISF (initiator send frame) state machine 299
9.2.6.2.2.1 ST_ISF state machine overview 299
9.2.6.2.2.2 ST_ISF1:Send_Frame state 299
9.2.6.2.2.2.1 State description 299
9.2.6.2.2.2.2 Transition ST_ISF1:Send_Frame to ST_ISF2:Prepare_Command_Task 301
9.2.6.2.2.2.3 Transition ST_ISF1:Send_Frame to ST_ISF3:Prepare_Data_Out 301
9.2.6.2.2.3 ST_ISF2:Prepare_Command_Task state 301
9.2.6.2.2.3.1 State description 301

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9.2.6.2.2.3.2 Transition ST_ISF2:Prepare_Command_Task to ST_ISF1:Send_Frame 301
9.2.6.2.2.4 ST_ISF3:Prepare_Data_Out state 301
9.2.6.2.2.4.1 State description 301
9.2.6.2.2.4.2 Transition ST_ISF3:Prepare_Data_Out to ST_ISF1:Send_Frame 302
9.2.6.2.3 ST_IPD (initiator process data) state machine 302
9.2.6.2.4 ST_IPR (initiator process response) state machine 302
9.2.6.2.5 ST_IFR (initiator frame router) state machine 303
9.2.6.3 ST_T (transport layer for SSP target ports) state machines 304
9.2.6.3.1 ST_T state machines overview 304
9.2.6.3.2 ST_TFR (target frame router) state machine 305
9.2.6.3.3 ST_TTS (target transport server) state machine 307
9.2.6.3.3.1 ST_TTS state machine overview 307
9.2.6.3.3.2 ST_TTS1:Start state 307
9.2.6.3.3.2.1 State description 307
9.2.6.3.3.2.2 Transition ST_TTS1:Start to ST_TTS2:Send_Frame 308
9.2.6.3.3.2.3 Transition ST_TTS1:Start to ST_TTS7:Prepare_Response 308
9.2.6.3.3.3 ST_TTS2:Send_Frame state 308
9.2.6.3.3.3.1 State description 308
9.2.6.3.3.3.2 Transition ST_TTS2:Send_Frame to ST_TTS3:Prepare_Data_In 309
9.2.6.3.3.3.3 Transition ST_TTS2:Send_Frame to ST_TTS4:Receive_Data_Out 310
9.2.6.3.3.3.4 Transition ST_TTS2:Send_Frame to ST_TTS7:Prepare_Response 310
9.2.6.3.3.4 ST_TTS3:Prepare_Data_In state 310
9.2.6.3.3.4.1 State description 310
9.2.6.3.3.4.2 Transition ST_TTS3:Prepare_Data_In to ST_TTS2:Send_Frame 310
9.2.6.3.3.5 ST_TTS4:Receive_Data_Out state 310
9.2.6.3.3.5.1 State description 310
9.2.6.3.3.5.2 Transition ST_TTS4:Receive_Data_Out to ST_TTS5:Prepare_Xfer_Rdy 311
9.2.6.3.3.5.3 Transition ST_TTS4:Receive_Data_Out to ST_TTS6:Process_Data_Out 311
9.2.6.3.3.6 ST_TTS5:Prepare_Xfer_Rdy state 311
9.2.6.3.3.6.1 State description 311
9.2.6.3.3.6.2 Transition ST_TTS5:Prepare_Xfer_Rdy to ST_TTS2:Send_Frame 312
9.2.6.3.3.7 ST_TTS6:Process_Data_Out state 312
9.2.6.3.3.7.1 State description 312
9.2.6.3.3.7.2 Transition ST_TTS6:Process_Data_Out to ST_TTS4:Receive_Data_Out 312
9.2.6.3.3.8 ST_TTS7:Prepare_Response state 312
9.2.6.3.3.8.1 State description 312
9.2.6.3.3.8.2 Transition ST_TTS7:Prepare_Response to ST_TTS2:Send_Frame 313
9.3 STP transport layer 313
9.3.1 Initial FIS 313
9.3.2 BIST Activate FIS 313
9.3.3 TT (transport layer for STP ports) state machines 313
9.4 SMP transport layer 314
9.4.1 SMP transport layer overview 314
9.4.2 SMP_REQUEST frame 315
9.4.3 SMP_RESPONSE frame 315
9.4.4 Sequence of SMP frames 316
9.4.5 MT (transport layer for SMP ports) state machines 316
9.4.5.1 SMP transport layer state machines overview 316
9.4.5.2 MT_IP (transport layer for SMP initiator ports) state machine 316
9.4.5.2.1 MT_IP state machine overview 316
9.4.5.2.2 MT_IP1:Idle state 317
9.4.5.2.2.1 State description 317
9.4.5.2.2.2 Transition MT_IP1:Idle to MT_IP2:Send 317
9.4.5.2.3 MT_IP2:Send state 317
9.4.5.2.3.1 State description 317
9.4.5.2.3.2 Transition MT_IP2:Send to MT_IP1:Idle 318
9.4.5.2.3.3 Transition MT_IP2:Send to MT_IP3:Receive 318
9.4.5.2.4 MT_IP3:Receive state 318

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9.4.5.2.4.1 State description 318
9.4.5.2.4.2 Transition MT_IP3:Receive to MT_IP1:Idle 318
9.4.5.3 MT_TP (transport layer for SMP target ports) state machine 318
9.4.5.3.1 MT_TP state machine overview 318
9.4.5.3.2 MT_TP1:Idle state 319
9.4.5.3.2.1 State description 319
9.4.5.3.2.2 Transition MT_TP1:Idle to MT_TP2:Respond 319
9.4.5.3.3 MT_TP2:Respond state 319
9.4.5.3.3.1 State description 319
9.4.5.3.3.2 Transition MT_TP2:Respond to MT_TP1:Idle 319

10 Application layer 321
10.1 Application layer overview 321
10.2 SCSI application layer 321
10.2.1 SCSI transport protocol services 321
10.2.1.1 SCSI transport protocol services overview 321
10.2.1.2 Send SCSI Command transport protocol service 322
10.2.1.3 SCSI Command Received transport protocol service 323
10.2.1.4 Send Command Complete transport protocol service 324
10.2.1.5 Command Complete Received transport protocol service 324
10.2.1.6 Send Data-In transport protocol service 325
10.2.1.7 Data-In Delivered transport protocol service 326
10.2.1.8 Receive Data-Out transport protocol service 326
10.2.1.9 Data-Out Received transport protocol service 327
10.2.1.10 Send Task Management Request transport protocol service 327
10.2.1.11 Task Management Request Received transport protocol service 328
10.2.1.12 Task Management Function Executed transport protocol service 328
10.2.1.13 Received Task Management Function-Executed transport protocol service 329
10.2.2 Application client error handling 330
10.2.3 Device server error handling 331
10.2.4 SCSI transport protocol event notifications 331
10.2.5 SCSI commands 331
10.2.5.1 INQUIRY command 331
10.2.5.2 LOG SELECT and LOG SENSE commands 332
10.2.5.3 MODE SELECT and MODE SENSE commands 332
10.2.5.4 START STOP UNIT command 332
10.2.6 SCSI mode parameters 332
10.2.6.1 Disconnect-Reconnect mode page 332
10.2.6.1.1 Disconnect-Reconnect mode page overview 332
10.2.6.1.2 BUS INACTIVITY TIME LIMIT field 333
10.2.6.1.3 MAXIMUM CONNECT TIME LIMIT field 334
10.2.6.1.4 MAXIMUM BURST SIZE field 334
10.2.6.1.5 FIRST BURST SIZE field 334
10.2.6.2 Protocol-Specific Port mode page 335
10.2.6.2.1 Protocol-Specific Port mode page overview 335
10.2.6.2.2 Protocol-Specific Port mode page - short format 335
10.2.6.2.3 Protocol-Specific Port mode page - Phy Control And Discover subpage 336
10.2.6.3 Protocol-Specific Logical Unit mode page 339
10.2.7 SCSI log parameters 339
10.2.7.1 Protocol-Specific log page 339
10.2.8 SCSI power conditions 342
10.2.8.1 SCSI power conditions overview 342
10.2.8.2 SA_PC (SCSI application layer power condition) state machine 342
10.2.8.2.1 SA_PC state machine overview 342
10.2.8.2.2 SA_PC_0:Powered_On state 343
10.2.8.2.2.1 State description 343
10.2.8.2.2.2 Transition SA_PC_0:Powered_On to SA_PC_4:Stopped 343
10.2.8.2.2.3 Transition SA_PC_0:Powered_On to SA_PC_5:Active_Wait 344

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.8.2.3 SA_PC_1:Active state 344
10.2.8.2.3.1 State description 344
10.2.8.2.3.2 Transition SA_PC_1:Active to SA_PC_2:Idle 344
10.2.8.2.3.3 Transition SA_PC_1:Active to SA_PC_3:Standby 344
10.2.8.2.3.4 Transition SA_PC_1:Active to SA_PC_4:Stopped 344
10.2.8.2.4 SA_PC_2:Idle state 344
10.2.8.2.4.1 State description 344
10.2.8.2.4.2 Transition SA_PC_2:Idle to SA_PC_1:Active 344
10.2.8.2.4.3 Transition SA_PC_2:Idle to SA_PC_3:Standby 344
10.2.8.2.4.4 Transition SA_PC_2:Idle to SA_PC_4:Stopped 344
10.2.8.2.5 SA_PC_3:Standby state 345
10.2.8.2.5.1 State description 345
10.2.8.2.5.2 Transition SA_PC_3:Standby to SA_PC_4:Stopped 345
10.2.8.2.5.3 Transition SA_PC_3:Standby to SA_PC_5:Active_Wait 345
10.2.8.2.5.4 Transition SA_PC_3:Standby to SA_PC_6:Idle_Wait 345
10.2.8.2.6 SA_PC_4:Stopped state 345
10.2.8.2.6.1 State description 345
10.2.8.2.6.2 Transition SA_PC_4:Stopped to SA_PC_3:Standby 345
10.2.8.2.6.3 Transition SA_PC_4:Stopped to SA_PC_5:Active_Wait 345
10.2.8.2.6.4 Transition SA_PC_4:Stopped to SA_PC_6:Idle_Wait 346
10.2.8.2.7 SA_PC_5:Active_Wait state 346
10.2.8.2.7.1 State description 346
10.2.8.2.7.2 Transition SA_PC_5:Active_Wait to SA_PC_1:Active 346
10.2.8.2.7.3 Transition SA_PC_5:Active_Wait to SA_PC_3:Standby 346
10.2.8.2.7.4 Transition SA_PC_5:Active_Wait to SA_PC_4:Stopped 346
10.2.8.2.7.5 Transition SA_PC_5:Active_Wait to SA_PC_6:Idle_Wait 346
10.2.8.2.8 SA_PC_6:Idle_Wait state 346
10.2.8.2.8.1 State description 346
10.2.8.2.8.2 Transition SA_PC_6:Idle_Wait to SA_PC_2:Idle 347
10.2.8.2.8.3 Transition SA_PC_6:Idle_Wait to SA_PC_3:Standby 347
10.2.8.2.8.4 Transition SA_PC_6:Idle_Wait to SA_PC_4:Stopped 347
10.2.8.2.8.5 Transition SA_PC_6:Idle_Wait to SA_PC_5:Active_Wait 347
10.2.9 SCSI vital product data (VPD) 348
10.3 ATA application layer 348
10.4 Management application layer 349
10.4.1 READY LED signal behavior 349
10.4.2 Management protocol services 349
10.4.3 SMP functions 350
10.4.3.1 SMP function request frame format 350
10.4.3.2 SMP function response frame format 352
10.4.3.3 REPORT GENERAL function 354
10.4.3.4 REPORT MANUFACTURER INFORMATION function 356
10.4.3.5 DISCOVER function 358
10.4.3.6 REPORT PHY ERROR LOG function 363
10.4.3.7 REPORT PHY SATA function 365
10.4.3.8 REPORT ROUTE INFORMATION function 367
10.4.3.9 CONFIGURE ROUTE INFORMATION function 370
10.4.3.10 PHY CONTROL function 372

Annex A (normative) Compliant jitter test pattern (CJTPAT) 376

Annex B (informative) SAS to SAS phy reset sequence examples 383

Annex C (informative) CRC 385
C.1 CRC generator and checker implementation examples 385
C.2 CRC implementation in C 385
C.3 CRC implementation with XORs 386
C.4 CRC examples 388

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex D (informative) SAS address hashing 389
D.1 SAS address hashing overview 389
D.2 Hash collision probability 389
D.3 Hash generation 390
D.4 Hash implementation in C 390
D.5 Hash implementation with XORs 391
D.6 Hash examples 392

Annex E (informative) Scrambling 395
E.1 Scrambler implementation example 395
E.2 Scrambler implementation in C 395
E.3 Scrambler implementation with XORs 396
E.4 Scrambler examples 397

Annex F (informative) ATA architectural notes 398
F.1 STP differences from Serial ATA (SATA) 398
F.2 STP differences from Serial ATA II 398
F.3 Affiliation policies 398
F.3.1 Affiliation policies overview 398
F.3.2 Affiliation policy for static STP initiator port to STP target port mapping 399
F.3.3 Affiliation policy with SATA queued commands and multiple STP initiator ports 399
F.3.4 Applicability of affiliation for STP target ports 399

Annex G (informative) Expander device handling of connections 400
G.1 Expander device handling of connections overview 400
G.2 Connection request - OPEN_ACCEPT 402
G.3 Connection request - OPEN_REJECT by end device 403
G.4 Connection request - OPEN_REJECT by expander device 404
G.5 Connection request - arbitration lost 405
G.6 Connection request - backoff and retry 406
G.7 Connection request - backoff and reverse path 407
G.8 Connection close - single step 408
G.9 Connection close - simultaneous 409
G.10 BREAK handling during path arbitration 410
G.11 BREAK handling during connection 411
G.12 STP connection - originated by STP initiator port 412
G.13 STP connection - originated by STP target port in an STP/SATA bridge 413
G.14 STP connection close - originated by STP initiator port 414
G.15 STP connection close - originated by STP target port in an STP/SATA bridge 415
G.16 Pathway blocked and pathway recovery example 416

Annex H (informative) Primitive encoding 417

Annex I (informative) Messages between state machines 420
I.1 Messages between phy layer and other layers 420
I.2 Messages between link layer, port layer, and management application layer for all protocols 420
I.3 Messages between link layer, port layer, and transport layer for SSP 422
I.4 Messages between link layer, port layer, and transport layer for SMP 424
I.5 Messages from transport layer to application layer for SSP 425
I.6 Messages from transport layer to application layer for SMP 426

Annex J (informative) Discover process example implementation 427
J.1 Discover process example implementation overview 427
J.2 Header file 427
J.3 Source file 440

Annex K (informative) SAS icon 451

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

List of figures

Page
Figure 1 — SCSI document relationships 26
Figure 2 — ATA document relationships 26
Organization of this standard 28
Figure 3 — Object and class diagram conventions 44
Figure 4 — Class diagram conventions for aggregation and generalization 45
Figure 5 — State machine conventions 46
Figure 6 — SAS object model 50
Figure 7 — Physical links and phys 51
Figure 8 — Phy object classes 52
Figure 9 — Ports (narrow ports and wide ports) 53
Figure 10 — Port object classes 54
Figure 11 — SAS devices 55
Figure 12 — Expander device 56
Figure 13 — Expander device object classes 56
Figure 14 — Domains 57
Figure 15 — SAS domain bridging to ATA domains 58
Figure 16 — Devices spanning SAS domains 58
Figure 17 — Edge expander device set 60
Figure 18 — Maximum expander device set topology 61
Figure 19 — Fanout expander device topology 62
Figure 20 — Edge expander device set to edge expander device set topology 63
Figure 21 — Potential pathways 64
Figure 22 — Multiple connections on wide ports 66
Figure 23 — State machines for SAS devices 69
Figure 24 — State machines for expander devices 70
Figure 25 — Transmit data path in a SAS phy 71
Figure 26 — SSP link, port, SSP transport, and SCSI application layer state machines 72
Figure 27 — SMP link, port, SMP transport, and management application layer state machines 73
Figure 28 — STP link, port, STP transport, and ATA application layer state machines 74
Figure 29 — Transmit data path and state machines in an expander phy 75
Figure 30 — State machine and SAS device, SAS port, and SAS phy objects 76
Figure 31 — State machine and expander device, expander port, and expander phy objects 77
Figure 32 — Reset terminology 78
Figure 33 — Expander device model 80
Figure 34 — Expander device interfaces 82
Figure 35 — Expander device interface detail 83
Figure 36 — Expander route table example 88
Figure 37 — Level-order traversal example 89
Figure 38 — Expander route index levels example 91
Figure 39 — Expander route index levels example with fanout expander device 92
Figure 40 — Expander route index order example 95
Figure 41 — SATA cables and connectors 97
Figure 42 — SAS cables and connectors - external environment 98
Figure 43 — SAS cables and connectors - internal environment 99
Figure 44 — SAS single-port internal cable assembly and destination pin assignments 105
Figure 45 — SAS dual-port internal cable assembly and destination pin assignments 106
Figure 46 — Transmitter transient test circuit 108
Figure 47 — Receiver transient test circuit 108
Figure 48 — Eye mask at IR, CR, and XR 110
Figure 49 — Deriving a tolerance mask at IR, CR, or XR 110
Figure 50 — Sinusoidal jitter mask 111
Figure 51 — Compliance interconnect test load 119
Figure 52 — Zero-length test load 119
Figure 53 — ISI loss example at 3,0 Gbps 120
Figure 54 — ISI loss example at 1,5 Gbps 120

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 55 — SAS bit transmission logic 129
Figure 56 — SAS bit reception logic 130
Figure 57 — OOB signal transmission 132
Figure 58 — OOB signal detection 134
Figure 59 — SATA OOB sequence 135
Figure 60 — SATA speed negotiation sequence 136
Figure 61 — SAS to SATA OOB sequence 137
Figure 62 — SAS to SAS OOB sequence 138
Figure 63 — SAS speed negotiation window 139
Figure 64 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G2 only) 140
Figure 65 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G1, G2) that fails 141
Figure 66 — Hot-plug and the phy reset sequence 142
Figure 67 — SP (phy layer) state machine - OOB sequence states 145
Figure 68 — SP (phy layer) state machine - SAS speed negotiation states 149
Figure 69 — SP (phy layer) state machine - SATA host emulation states 154
Figure 70 — SP_DWS (phy layer dword synchronization) state machine 158
Figure 71 — Repeated primitive sequence 172
Figure 72 — Triple primitive sequence 173
Figure 73 — Redundant primitive sequence 174
Figure 74 — Elasticity buffers 183
Figure 75 — CRC generator bit order 186
Figure 76 — STP CRC bit ordering 187
Figure 77 — Transmit path bit ordering 189
Figure 78 — Receive path bit ordering 190
Figure 79 — STP transmit path bit ordering 191
Figure 80 — STP receive path bit ordering 192
Figure 81 — SL_IR (link layer identification and hard reset) state machines 201
Figure 82 — Aborting a connection request with BREAK 212
Figure 83 — Connection request timeout example 213
Figure 84 — Closing a connection example 214
Figure 85 — Rate matching example 215
Figure 86 — SL (link layer for SAS phys) state machines (part 1) 217
Figure 87 — SL (link layer for SAS phys) state machines (part 2) 218
Figure 88 — XL (link layer for expander phys) state machine (part 1) 227
Figure 89 — XL (link layer for expander phys) state machine (part 2) 228
Figure 90 — XL (link layer for expander phys) state machine (part 3) 229
Figure 91 — SSP frame transmission 237
Figure 92 — Interlocked frames 238
Figure 93 — Non-interlocked frames with the same tag 239
Figure 94 — Non-interlocked frames with different tags 239
Figure 95 — Closing an SSP connection example 240
Figure 96 — SSP (link layer for SSP phys) state machines (part 1 - frame transmission) 242
Figure 97 — SSP (link layer for SSP phys) state machines (part 2 - frame reception) 243
Figure 98 — STP frame transmission 250
Figure 99 — STP flow control 252
Figure 100 — STP initiator port opening an STP connection 255
Figure 101 — STP target port opening an STP connection 256
Figure 102 — SMP frame transmission 257
Figure 103 — SMP_IP (link layer for SMP initiator phys) state machine 259
Figure 104 — SMP_TP (link layer for SMP target phys) state machine 261
Figure 105 — Port layer examples 264
Figure 106 — PL_OC (port layer overall control) state machine 266
Figure 107 — PL_PM (port layer phy manager) state machine (part 1) 275
Figure 108 — PL_PM (port layer phy manager) state machine (part 2) 276
Figure 109 — Task management function sequence of SSP frames 292
Figure 110 — Write command sequence of SSP frames 293
Figure 111 — Read command sequence of SSP frames 293
Figure 112 — Bidirectional command sequence of SSP frames 294

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 113 — ST_I (transport layer for SSP initiator ports) state machines 298
Figure 114 — ST_T (transport layer for SSP target ports) state machines 305
Figure 115 — Sequence of SMP frames 316
Figure 116 — MT_IP (transport layer for SMP initiator ports) state machine 317
Figure 117 — MT_TP (transport layer for SMP target ports) state machine 319
Figure 118 — SA_PC (SCSI application layer power condition) state machine for SAS 343
Figure B.1 — SAS speed negotiation sequence (phy A: G1 only, phy B: G1 only) 383
Figure B.2 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G1, G2) 384
Figure C.1 — CRC generator example 385
Figure C.2 — CRC checker example 385
Figure D.1 — BCH(69, 39, 9) code generator 390
Figure E.1 — Scrambler 395
Figure G.1 — Example topology 400
Figure G.2 — Connection request - OPEN_ACCEPT 402
Figure G.3 — Connection request - OPEN_REJECT by end device 403
Figure G.4 — Connection request - OPEN_REJECT by expander device 404
Figure G.5 — Connection request - arbitration lost 405
Figure G.6 — Connection request - backoff and retry 406
Figure G.7 — Connection request - backoff and reverse path 407
Figure G.8 — Connection close - single step 408
Figure G.9 — Connection close - simultaneous 409
Figure G.10 — BREAK handling during path arbitration 410
Figure G.11 — BREAK handling during a connection 411
Figure G.12 — STP connection - originated by STP initiator port 412
Figure G.13 — STP connection - originated by STP target port in an STP/SATA bridge 413
Figure G.14 — STP connection close - originated by STP initiator port 414
Figure G.15 — STP connection close - originated by STP target port in an STP/SATA bridge 415
Figure G.16 — Partial pathway recovery 416
Figure K.1 — SAS icon 451

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Tables

Page
Table 1 ISO and American numbering conventions ... 43
Table 2 Data dword containing a value ... 47
Table 3 Data dword containing four one-byte fields ... 48
Table 4 Names and identifiers ... 67
Table 5 SAM-3 object mapping ... 67
Table 6 SAS address format ... 67
Table 7 Hashed SAS address code parameter ... 68
Table 8 Expander phy to ECM requests ... 83
Table 9 Expander phy to ECM responses ... 84
Table 10 ECM to expander phy confirmations ... 84
Table 11 Expander phy to ECR to expander phy requests and indications ... 85
Table 12 Expander phy to ECR to expander phy responses and confirmations ... 85
Table 13 Expander phy to BPP requests ... 86
Table 14 BPP to expander phy indications ... 86
Table 15 Expander route table levels for edge expander device R or fanout expander device R ... 93
Table 16 Expander route table levels for edge expander device N ... 94
Table 17 Expander route entries for edge expander E0 phy 0 ... 96
Table 18 Expander route entries for fanout expander device F phy 0 ... 96
Table 19 Connectors ... 100
Table 20 SAS target device connector pin assignments ... 102
Table 21 Physical link usage in SAS external connector ... 104
Table 22 Compliance points ... 107
Table 23 General interface characteristics ... 109
Table 24 Signal characteristics at IT, CT, XT ... 112
Table 25 Signal characteristics at IR, CR, and XR ... 113
Table 26 Maximum allowable jitter at IR, CR, XR ... 115
Table 27 Receiver jitter tolerance ... 116
Table 28 Impedance requirements ... 116
Table 29 Output characteristics of the READY LED signal ... 121
Table 30 Special character usage ... 122
Table 31 Bit designations ... 123
Table 32 Conversion example ... 123
Table 33 Valid data characters ... 125
Table 34 Valid special characters ... 127
Table 35 Delayed code violation example ... 128
Table 36 OOB signal timing specifications ... 131
Table 37 OOB signal transmitter requirements ... 131
Table 38 OOB signal receiver burst time detection requirements ... 133
Table 39 OOB signal receiver idle time detection requirements ... 133
Table 40 OOB signal receiver negation time detection requirements ... 133
Table 41 Phy reset sequence timing specifications ... 135
Table 42 SATA speed negotiation sequence timing specifications ... 136
Table 43 SAS speed negotiation sequence timing specifications ... 139
Table 44 SP state machine timers ... 143
Table 45 SP_DWS timers ... 158
Table 46 Primitive format ... 163
Table 47 Primitives not specific to type of connection ... 164
Table 48 Primitives used only inside SSP and SMP connections ... 166
Table 49 Primitives used only inside STP connections and on SATA physical links ... 167
Table 50 Primitive encoding for primitives not specific to type of connection ... 168
Table 51 Primitive encoding for primitives used only inside SSP and SMP connections ... 170
Table 52 Primitive encoding for primitives used only inside STP connections and on SATA physical links 171
Table 53 Primitive sequences ... 172
Table 54 AIP primitives ... 175
Table 55 ALIGN primitives ... 175

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 56 BROADCAST primitives 176
Table 57 CLOSE primitives 177
Table 58 NOTIFY primitives 178
Table 59 OPEN_REJECT abandon primitives 179
Table 60 OPEN_REJECT retry primitives 180
Table 61 DONE primitives 181
Table 62 NAK primitives 182
Table 63 RRDY primitives 182
Table 64 Clock skew management ALIGN or NOTIFY insertion requirements 184
Table 65 CRC polynomials 185
Table 66 Scrambling for different data dword types 188
Table 67 Address frame format 193
Table 68 Address frame types 193
Table 69 IDENTIFY address frame format 194
Table 70 Device types 194
Table 71 OPEN address frame format 196
Table 72 Protocol 197
Table 73 Connection rate 197
Table 74 Arbitration wait time 198
Table 75 SL_IR timers 200
Table 76 Connection responses 207
Table 77 Arbitration priority for OPENs passing on a physical link 208
Table 78 Arbitration priority for contending path requests in the ECM 209
Table 79 Pathway recovery priority 210
Table 80 Abort connection responses 211
Table 81 Close connection responses 213
Table 82 Break connection responses 214
Table 83 SL_CC timers 220
Table 84 XL timers 226
Table 85 SSP frame interlock requirements 238
Table 86 SSP link layer timers 241
Table 87 PL_OC state machine timers 265
Table 88 Confirmations from Unable To Connect or Retry Open messages 271
Table 89 PL_PM state machine timers 274
Table 90 Messages from Open Failed confirmations 278
Table 91 SSP frame format 283
Table 92 FRAME TYPE field 284
Table 93 COMMAND information unit 285
Table 94 TASK ATTRIBUTE field 286
Table 95 TASK information unit 286
Table 96 Task management functions 287
Table 97 XFER_RDY information unit 288
Table 98 DATA information unit 288
Table 99 RESPONSE information unit 290
Table 100 DATAPRES field 290
Table 101 RESPONSE DATA field 291
Table 102 RESPONSE CODE field 291
Table 103 Delivery Failure to Command Complete Received mapping 303
Table 104 ST_T state machine timers 304
Table 105 Response Data argument to RESPONSE frame content mapping 312
Table 106 SMP frame format 314
Table 107 SMP FRAME TYPE field 314
Table 108 SMP_REQUEST frame format 315
Table 109 SMP_RESPONSE frame format 315
Table 110 MT_IP timers 316
Table 111 SCSI architecture mapping 322
Table 112 Send SCSI Command transport protocol service arguments 323
Table 113 SCSI Command Received transport protocol service arguments 323

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 114 Send Command Complete transport protocol service arguments 324
Table 115 Command Complete Received transport protocol service arguments 325
Table 116 Send Data-In transport protocol service arguments 326
Table 117 Data-In Delivered transport protocol service arguments 326
Table 118 Receive Data-Out transport protocol service arguments 327
Table 119 Data-Out Received transport protocol service arguments 327
Table 120 Send Task Management Request transport protocol service arguments 328
Table 121 Task Management Request Received transport protocol service arguments 328
Table 122 Task Management Function Executed transport protocol service arguments 329
Table 123 Received Task Management Function-Executed transport protocol service arguments 330
Table 124 Delivery Result to additional sense code mapping 331
Table 125 SCSI transport protocol events 331
Table 126 Disconnect-Reconnect mode page for SSP 333
Table 127 Protocol-Specific Port Control mode page subpages 335
Table 128 Protocol-Specific Port Control mode page for SAS SSP - short format 335
Table 129 I_T nexus loss time 336
Table 130 Protocol-Specific Port Control mode page for SAS SSP - Phy Control And Discover subpage 336
Table 131 SAS phy mode descriptor 338
Table 132 Protocol-Specific log page for SAS 339
Table 133 Protocol-Specific log parameter format for SAS 340
Table 134 Parameter control bits for SAS log parameters 340
Table 135 SAS phy log descriptor 341
Table 136 Device Identification VPD page required identification descriptors 348
Table 137 SMP request frame format 350
Table 138 SMP functions 351
Table 139 SMP response frame format 352
Table 140 Function results 353
Table 141 REPORT GENERAL request 354
Table 142 REPORT GENERAL response 355
Table 143 REPORT MANUFACTURER INFORMATION request 356
Table 144 REPORT MANUFACTURER INFORMATION response 357
Table 145 DISCOVER request 358
Table 146 DISCOVER response 359
Table 147 Attached device types 360
Table 148 Negotiated physical link rate 360
Table 149 Programmed minimum and maximum physical link rates 362
Table 150 Hardware minimum and maximum physical link rates 362
Table 151 Routing attributes 363
Table 152 REPORT PHY ERROR LOG request 363
Table 153 REPORT PHY ERROR LOG response 364
Table 154 REPORT PHY SATA request 365
Table 155 REPORT PHY SATA response 366
Table 156 REPORT ROUTE INFORMATION request 368
Table 157 REPORT ROUTE INFORMATION response 369
Table 158 CONFIGURE ROUTE INFORMATION request 371
Table 159 CONFIGURE ROUTE INFORMATION response 372
Table 160 PHY CONTROL request 373
Table 161 Phy operation 374
Table 162 Programmed minimum and maximum physical link rate 375
Table 163 PHY CONTROL response 375
Table A.1 CJTPAT for RD+ 376
Table A.2 CJTPAT for RD- 377
Table A.3 CJTPAT for RD+ and RD- 378
Table A.4 CJTPAT scrambled in an SSP DATA frame 379
Table C.1 CRC examples 388
Table D.1 Monte-Carlo simulation results 389
Table D.2 Hash results for simple SAS addresses 392
Table D.3 Hash results for realistic SAS addresses 392

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table D.4 Hash results for a walking ones pattern 393
Table D.5 Hash results for a walking zeros pattern 394
Table E.1 Scrambler examples 397
Table G.1 Column descriptions for connection examples 401
Table H.1 Primitives with Hamming distance of 8 417
Table I.1 Requests from management application layer or link layer to phy layer 420
Table I.2 Confirmations from phy layer to link layer 420
Table I.3 Requests between link layer and port layer 420
Table I.4 Confirmations between link layer and port layer 421
Table I.5 Requests from management application layer to link layer 421
Table I.6 Confirmations between link layer and port layer, link layer, or application layer 421
Table I.7 Requests between link layer, port layer, and transport layer for SSP 422
Table I.8 Confirmations from port layer to transport layer for SSP 422
Table I.9 Confirmations between SL link layer, port layer, and SSP transport layer 423
Table I.10 Confirmations between SSP link layer, port layer, and SSP transport layer 424
Table I.11 Requests between SL/SMP link layer, port layer, and SMP transport layer 424
Table I.12 Confirmations between link layer, port layer, and SMP transport layer 425
Table I.13 Requests and responses from SCSI application layer to SSP transport layer 425
Table I.14 Confirmations and indications from SSP transport layer to SCSI application layer 426
Table I.15 Requests from management application layer to SMP transport layer 426
Table I.16 Confirmations from SMP transport layer to management application layer 426
Table J.1 C program files 427

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INFORMATION TECHNOLOGY - SMALL COMPUTER SYSTEM INTERFACE - PART 150: Serial attached SCSI (SAS)

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 through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.
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) All users should ensure that they have the latest edition of this publication.
4) 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 National Committees 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, use of, or reliance upon, this ISO/IEC publication or any other IEC or IEC/ISO publications.
5) 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.
6) CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

   As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

   ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14776-150 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INTRODUCTION

SCSI family of standards

The SCSI family of standards provides for many different transport protocols. The SCSI transfer protocol standards define the rules for exchanging information between SCSI devices using interconnects.

Figure 1 shows the relationship of this standard, ISO/IEC 14776-150, to other standards and to related projects in the SCSI family of standards.

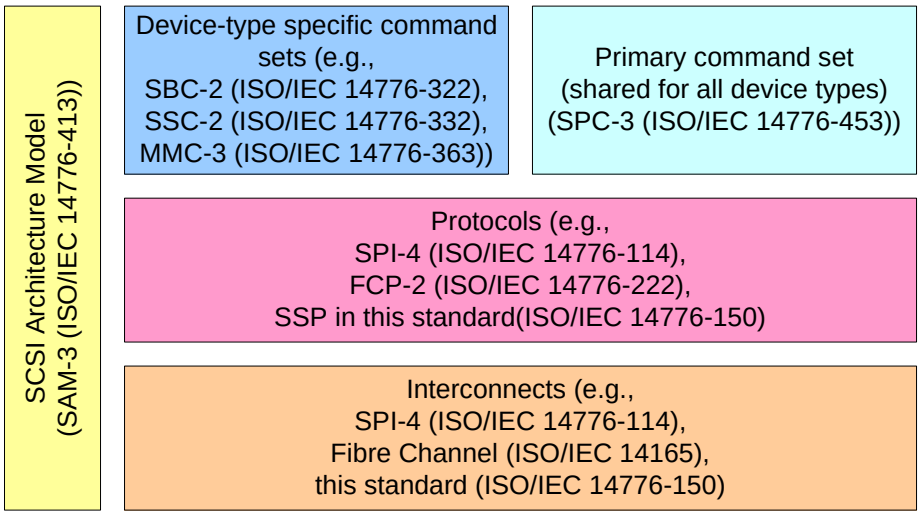


Figure 1 — SCSI document relationships

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 2 shows the relationship of this standard, SO/IEC 14776-150, to other standards and related projects in the ATA family of standards.

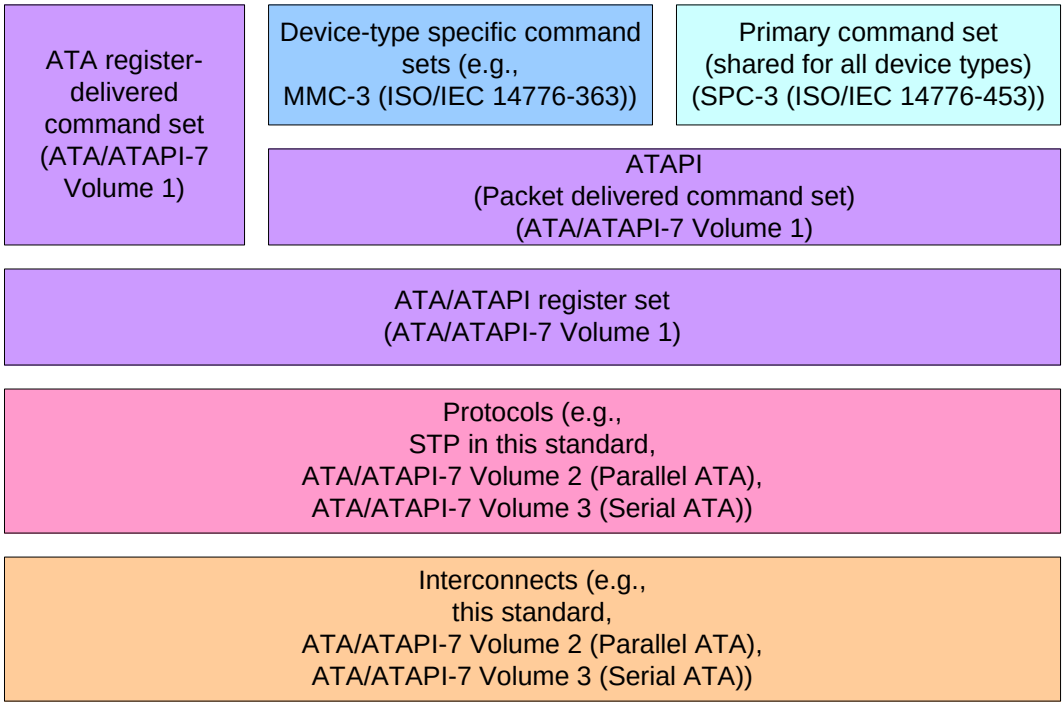


Figure 2 — ATA document relationships

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Figure 1 and figure 2 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.

This standard is intended to be used in conjunction with other SCSI and with ATA specifications.

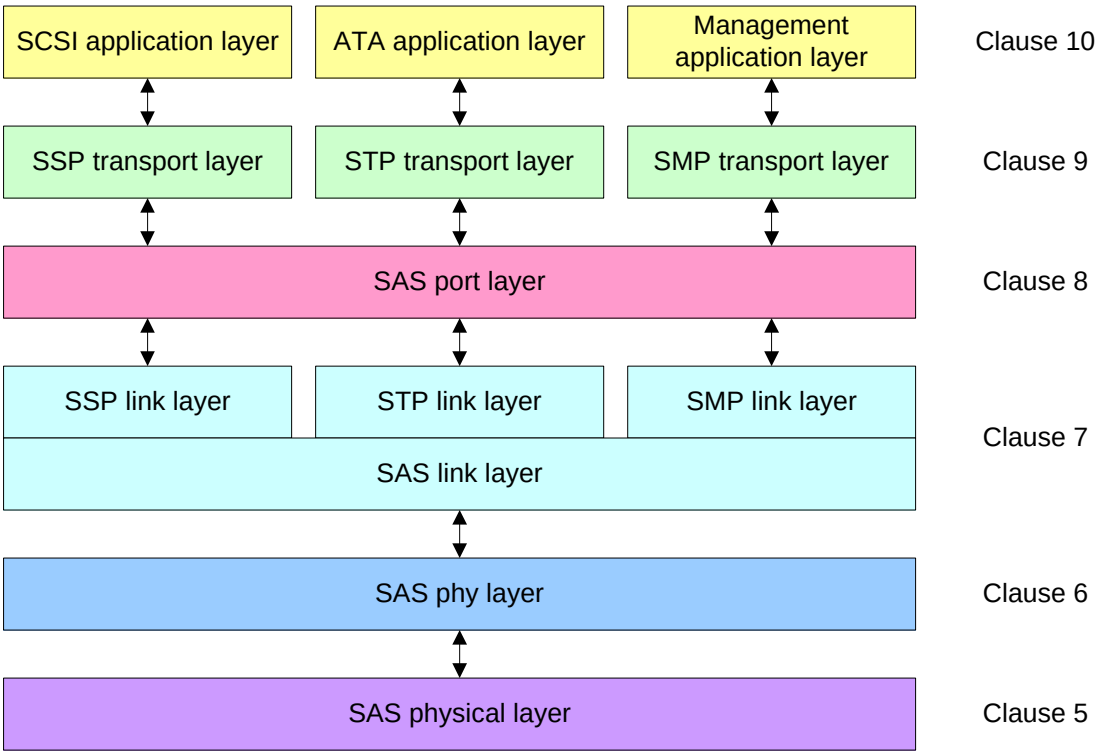
Organisation of this standard

Clause 1 (Scope) describes the relationship of this standard to the SCSI and ATA families of standards.
Clause 2 (References) provides references to other standards and documents.
Clause 3 (Definitions, symbols, abbreviations, keywords, and conventions) defines terms and conventions used throughout this standard.
Clause 4 (General) describes architecture, names and identifiers, state machines, resets, I_T nexus loss, and provides an expander device model.
Clause 5 (Physical layer) describes the physical layer. It describes passive interconnect components (connectors, cables, and backplanes) and defines the transmitter and receiver electrical characteristics.
Clause 6 (Phy layer) describes the phy layer. It describes 8b10b encoding, bit order, out of band (OOB) signals, phy reset sequences, phy layer state machines, and spin-up.
Clause 7 (Link layer) describes the link layer. It describes primitives, clock skew management, idle physical links, CRC, scrambling, address frames, the identification sequence and its state machine, power management, SAS domain changes, connections, rate matching, and SSP, STP, and SMP connection rules and link layer state machines.
Clause 8 (Port layer) describes the port layer, which sits between one or more link layers and one or more transport layers. It includes port layer state machines.
Clause 9 (Transport layer) describes the transport layer. It includes SSP, STP, and SMP frame definitions and transport layer state machines.
Clause 10 (Application layer) describes the application layer. It describes SCSI protocol services, mode parameters, log parameters, and power conditions, ATA specifics, and SMP functions.

Normative Annex A (Compliant jitter test pattern (CJTPAT)) describes the jitter test patterns.
Informative Annex B (SAS to SAS phy reset sequence examples) provides additional phy reset sequence examples.
Informative Annex C (CRC) provides information and example implementations of the CRC algorithm.
Informative Annex D (SAS address hashing) provides information and example implementations of the hashing algorithm.
Informative Annex E (Scrambling) provides information and example implementations of the scrambling algorithm.
Informative Annex F (ATA architectural notes) describes ATA architectural differences from Serial ATA and Serial ATA II.
Informative Annex G (Expander device handling of connections) describes expander device behavior in a variety of connection examples.
Informative Annex H (Primitive encoding) lists the primitive encodings available for future versions of this standard.
Informative Annex I (Messages between state machines) contains a list of messages between state machines.
Informative Annex J (Discover process example implementation) provides an example implementation of the discover process.
Informative Annex K (SAS icon) defines the SAS logo.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The following diagram shows the organization of the layers of this standard.



Organization of this standard

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Information Technology - Small Computer System Interface (SCSI) - Part 150: Serial Attached SCSI (SAS)

1 Scope

This part of ISO/IEC 14776 defines the rules for exchanging information between SCSI devices using the Serial Attached SCSI (SAS) interconnect, which is compatible with the Serial ATA (SATA) physical interconnect. It also specifies three transport protocols, one to transport SCSI commands, another to transport Serial ATA commands to multiple SATA devices, and a third one to support interface management.

2 References

2.1 Normative references

At the time of publication, none of the referenced standards were published.

2.2 References under development

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

ISO/IEC xxx (under consideration), A*T Attachment with Packet Interface-7 volume 1 (ATA/ATAPI-7 V1) standard (T13/1532-D)*

ISO/IEC xxx (under consideration), *AT Attachment with Packet Interface-7 volume 2 (ATA/ATAPI-7 V2) standard (T13/1532-D)*

ISO/IEC xxx (under consideration), *AT Attachment with Packet Interface-7 volume 3 (ATA/ATAPI-7 V3) standard (T13/1532-D)*

ISO/IEC 14776-114 (under consideration), *Information technology - Small computer system interface (SCSI) - Part 114: (SPI-4)*

ISO/IEC 14776-222 (under consideration), *Information technology - Small computer system interface (SCSI) - Part 222: Fibre channel protocol for SCSI 2 (FCP-2)*

ISO/IEC 14776-322 (under consideration), *Information technology - Small computer system interface (SCSI) - Part 322: (SBC-2)*

ISO/IEC 14776-332 (under consideration), *Information technology - Small computer system interface (SCSI) - Part 332: (SSC-2)*

ISO/IEC 14776-363 (under consideration), I*nformation technology - Small computer system interface (SCSI) - Part 363: (MMC-3)*

ISO/IEC 14776-413 (under consideration), *Information technology - Small computer system interface (SCSI) - Part 413: Architecture Model-3 (SAM-3)*

ISO/IEC 14776-453 (under consideration), *Information technology - Small computer system interface (SCSI) - Part 453: Primary Commands-3 (SPC-3)*

NOTE 1 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.

2.3 Bibliography

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

ISO/IEC 14165 (all parts), *Information technology - Fibre channel*

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Serial ATA II: Extensions to Serial ATA 1.0. Revision 1.0. 16 October 2002

NOTE 2 For more information on the current status of the Serial ATA documents, contact the Serial ATA International Organisation at http://www.serialata.org.

SFF-8223, 2.5" *Drive w/Serial Attachment Connector*

SFF-8323, 3.5" *Drive w/Serial Attachment Connector*

SFF-8410, *High Speed Serial Testing for Copper Links*

SFF-8460, *HSS Backplane Design Guidelines*

SFF-8470, *Multi Lane Copper Connector*

SFF-8482, *Unshielded Dual Port Serial Attachment Connector*

SFF-8523, 5,25" *Drive w/Serial Attachment Connector*

NOTE 3 For more information on the current status of the SFF documents, contact the SFF Committee at 408-867-6630 (phone), or 408-867-2115 (fax). To obtain copies of these documents, contact the SFF Committee at 14426 Black Walnut Court, Saratoga, CA 95070 at 408-867-6630 (phone) or 408-741-1600 (fax) or see http://www.sffcommittee.org.

OMG Unified Modeling Language (UML) Specification. Version 1.4, September 2001.

NOTE 4 For more information on the UML specification, contact the Object Modeling Group at http://www.omg.org.

Common Information Model (CIM) Specification. Version 2.2, 14 June 1999.

NOTE 5 For more information on the CIM specification, contact the Desktop Management Task Force, Inc. at http://www.dmtf.org.

3 Definitions, symbols, abbreviations, keywords and conventions

3.1 Definitions

3.1.1 8b10b coding: A coding scheme that represents an 8-bit data byte as a 10-bit character. See 6.2.

3.1.2 affiliation: STP target port state of limiting acceptance of connection requests to those from a single STP initiator port. See 7.17.3.

3.1.3 application client: An object that is the source of SCSI commands (see SAM-3), ATA commands (see ATA/ATAPI-7 V1), or management function requests.

3.1.4 AT Attachment (ATA): A standard for the internal attachment of storage devices to hosts. See ATA/ATAPI-7 V1.

3.1.5 ATA device: A storage peripheral (analogous to a SCSI target device). See ATA/ATAPI-7 V1.

3.1.6 ATA domain: An I/O system consisting of a set of ATA hosts and ATA devices that communicate with one another by means of a service delivery subsystem. See 4.1.1.

3.1.7 ATA host: A host device that originates requests to be processed by an ATA device (analogous to a SCSI initiator device). See ATA/ATAPI-7 V1.

3.1.8 big-endian: A 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).

3.1.9 broadcast primitive processor (BPP): An object within an expander function that manages broadcast primitives. See 4.6.5.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.10 burst time: The part of an OOB signal (see 3.1.82) where ALIGN primitives (see 3.1.93) are being transmitted. See 6.5.

3.1.11 byte: A sequence of eight contiguous bits considered as a unit.

3.1.12 character: A sequence of ten contiguous bits considered as a unit. A byte is encoded as a character using 8b10b coding (see 6.2).

3.1.13 command descriptor block (CDB): A structure used to communicate a command from a SCSI application client to a SCSI device server. See SAM-3.

3.1.14 compliant jitter test pattern (CJTPAT): A test pattern for jitter testing. See 5.3.8.

3.1.15 configurable expander device: An expander device that contains an expander route table that is configured with expander route entries. See 4.1.5.

3.1.16 confirmation: A message passed from a lower layer state machine to a higher layer state machine, usually responding to a request (see 3.1.102) from that higher layer state machine, and sometimes relaying a response (see 3.1.104) from a peer higher layer state machine.

3.1.17 connection: A temporary association between a SAS initiator port and a SAS target port. See 7.12.

3.1.18 connection rate: The effective rate of dwords through the pathway between a SAS initiator phy and a SAS target phy, established through the connection request.

3.1.19 control character (Kxx.y): A character that does not represent a byte of data. See 6.2.

3.1.20 cyclic redundancy check (CRC): An error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum. See 7.5.

3.1.21 D.C. idle: A differential signal level that is nominally 0 V(P-P). See 5.3.4.

3.1.22 data character (Dxx.y): A character representing a byte of data. See 6.2.

3.1.23 data dword: A dword that starts with a Dxx.y (data character).

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

3.1.25 deterministic jitter: Jitter with a non-Gaussian probability density function. Deterministic jitter is always bounded in amplitude and has specific causes. Four kinds of deterministic jitter are identified: duty cycle distortion, data dependent, sinusoidal, and uncorrelated (to the data) bounded. Deterministic jitter is characterized by its bounded, peak-to-peak value.

3.1.26 device server: An object within a SAS target device that processes SCSI tasks (see SAM-3), ATA commands (see ATA/ATAPI-7 V1), or management functions.

3.1.27 direct current (D.C.): The non-A.C. component of a signal. In this standard, all frequency components below 100 kHz.

3.1.28 direct routing attribute: The attribute of an expander phy that indicates it may be used by the expander connection manager to route a connection request to an end device. See 4.6.7.1.

3.1.29 direct routing method: The method the expander connection manager uses to establish a connection with an end device. See 4.6.7.1.

3.1.30 discover process: The algorithm used by a management application client to configure the SAS domain. See 4.6.7.4.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.31 disparity: The difference between the number of ones and zeros in a character (see 6.2).

3.1.32 domain: A SAS domain, a SCSI domain, or an ATA domain.

3.1.33 dword: A sequence of four contiguous bytes or four contiguous characters considered as a unit.

3.1.34 dword synchronization: Detection of an incoming stream of dwords from a physical link by a phy. See 6.8.

3.1.35 edge expander device: An expander device that is part of a single edge expander device set.

3.1.36 edge expander device set: A group of one or more edge expander devices that may be attached to no more than one other edge expander device set or one fanout expander device. See 4.1.8.2.

3.1.37 end device: A SAS device that is not contained within an expander device.

3.1.38 event notification: A message passed from the transport layer to the application layer notifying the application layer that a SCSI event has occurred. See SAM-3.

3.1.39 expander connection manager (ECM): An object within an expander function that manages routing. See 4.6.4.

3.1.40 expander connection router (ECR): The portion of an expander function that routes messages between expander phys. See 4.6.4.

3.1.41 expander device: A device that is part of the service delivery subsystem and facilitates communication between SAS devices. See 4.1.5.

3.1.42 expander function: An object within an expander device that contains an expander connection manager, expander connection router, and broadcast primitive processor. See 4.6.1.

3.1.43 expander phy: A phy in an expander device that interfaces to a service delivery subsystem.

3.1.44 expander port: An expander device object that interfaces to the service delivery subsystem and to SAS ports in other devices.

3.1.45 expander route entry: A routed SAS address and an enable/disable bit in an expander route table (see 4.6.7.3).

3.1.46 expander route index: A value used in combination with a phy identifier to select an expander route entry in an expander route table (see 4.6.7.3).

3.1.47 expander route table: A table of expander route entries within an expander device. The table is used by the expander function to resolve connection requests. See 4.6.7.3.

3.1.48 fanout expander device: An expander device that is capable of being attached to two or more edge expander device sets.

3.1.49 field: A group of one or more contiguous bits.

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

3.1.51 frame information structure (FIS): The SATA frame format. See ATA/ATAPI-7 V3.

3.1.52 hard reset: A SAS device or expander device action in response to a reset event in which the device performs the operations described in 4.4.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.53 hard reset sequence: A sequence that causes a hard reset (see 4.4).

3.1.54 hardware maximum physical link rate: The maximum physical link rate capability of a phy.

3.1.55 hardware minimum physical link rate: The minimum physical link rate capability of a phy.

3.1.56 hash function: A 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.

3.1.57 I_T nexus: A nexus that exists between a SCSI initiator port and a SCSI target port.

3.1.58 I_T nexus loss: A condition where a SAS port determines that another SAS port is no longer available. See 4.5.

3.1.59 I_T_L nexus: A nexus that exists between a SCSI initiator port, a SCSI target port, and a logical unit. This relationship extends the prior I_T nexus.

3.1.60 I_T_L_Q nexus: A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a tagged task. This relationship extends the prior I_T nexus or I_T_L nexus.

3.1.61 identification sequence: A sequence where phys exchange IDENTIFY address frames. See 4.4.

3.1.62 idle dword: A vendor-specific data dword that is scrambled and is transmitted outside a frame. See 7.4.

3.1.63 idle time: The part of an OOB signal (see 3.1.82) where D.C. idle (see 5.3.4) is being transmitted. See 6.5.

3.1.64 indication: A message passed from a lower layer state machine to a higher layer state machine, usually relaying a request (see 3.1.102) from a peer higher layer state machine.

3.1.65 information unit (IU): The portion of an SSP frame that carries command, task management function, data, response, or transfer ready information. See 9.2.2.

3.1.66 invalid dword: A dword with an illegal character, with a control character in other than the first character position, with a control character other than K28.5 or K28.3 in the first character position, or with a running disparity error.

3.1.67 jitter: Abrupt and unwanted variations in the interval between successive pulses.

3.1.68 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).

3.1.69 link reset: Performing the link reset sequence (see 3.1.70).

3.1.70 link reset sequence: For SATA, a phy reset sequence (see 3.1.87). For SAS, a phy reset sequence followed by an identification sequence (see 3.1.61), or a phy reset sequence followed by a hard reset sequence (see 3.1.53), another phy reset sequence, and an identification sequence. See 4.4.

3.1.71 little-endian: A 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).

3.1.72 livelock: A 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.73 media: Particular elements comprising the interconnect including copper cables, printed circuit boards, and other transmission line materials.

3.1.74 message: Information sent between state machines.

3.1.75 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).

3.1.76 narrow link: A physical link that attaches a narrow port to another narrow port. See 4.1.3.

3.1.77 narrow port: A port that contains exactly one phy. See 4.1.3.

3.1.78 negotiated physical link rate: The current operational physical link rate established after speed negotiation between two phys.

3.1.79 nexus: A relationship between a SCSI initiator port and a SCSI target port that may extend to a logical unit and a tagged task. See SAM-3.

3.1.80 object: An architectural abstraction or container that encapsulates data types, services, or other objects that are related in some way.

3.1.81 OOB sequence: A sequence where two phys exchange OOB signals. See 4.4.

3.1.82 OOB signal: Pattern of ALIGNs and idle time used during the link reset sequence. See 6.5.

3.1.83 partial pathway: The set of physical links participating in a connection request which has not reached a SAS endpoint (i.e., the connection request has been transmitted by the source device and confirmed as received by at least one expander device with AIP). See 4.1.9.

3.1.84 pathway: A set of physical links between a SAS initiator phy and a SAS target phy being used by a connection. See 4.1.9.

3.1.85 pathway blocked count: The number of times the port has retried this connection request due to receiving OPEN_REJECT (PATHWAY BLOCKED).

3.1.86 phy: A device object that is used to interface to other devices. See 4.1.2.

3.1.87 phy reset sequence: An OOB sequence (see 3.1.81) followed by a speed negotiation sequence (see 3.1.148). See 4.4.

3.1.88 physical link: Two differential signal pairs, one pair in each direction, that connect two physical phys. See 4.1.2.

3.1.89 physical phy: A phy (see 3.1.86) that contains a transceiver and electrically interfaces to a physical link to communicate with another physical phy. See 4.1.2.

3.1.90 port: A SAS port or an expander port. Each port contains one or more phys. See 4.1.3.

3.1.91 potential pathway: A set of physical links between a SAS initiator phy and a SAS target phy. See 4.1.9.

3.1.92 power on: Power being applied.

3.1.93 primitive: A dword starting with K28.5 or K28.3 followed by three data characters. See 7.2.

3.1.94 primitive sequence: A set of primitives treated as a single entity. See 7.2.4.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.95 programmed maximum physical link rate: The maximum operational physical link rate of a phy (e.g., as programmed via the SMP PHY CONTROL function (see 10.4.3.10) or the Phy Control and Discover subpage (see 10.2.6.2.3)).

3.1.96 programmed minimum physical link rate: The minimum operational physical link rate of a phy (e.g., as programmed via the SMP PHY CONTROL function (see 10.4.3.10) or the Phy Control and Discover subpage (see 10.2.6.2.3)).

3.1.97 random jitter: Jitter that is assumed to have a Gaussian distribution.

3.1.98 rate: Data transfer rate of a physical link (e.g., 1,5 Gbps or 3,0 Gbps).

3.1.99 rate change delay time (RCDT): The time between rates during the speed negotiation sequence (see 6.6.4.2).

3.1.100 receiver: The recipient of a signal.

3.1.101 reflection coefficient (ρ): The reflection coefficient of the transmission media (i.e., the ratio of the reflected voltage divided by the voltage applied to the transmission media).

3.1.102 request: A message passed from a higher layer state machine to a lower layer state machine, usually to initiate some action.

3.1.103 reset event: An event that triggers a hard reset (see 4.4.2) from a SAS device.

3.1.104 response: A message passed from a higher layer state machine to a lower layer state machine, usually in response to an indication (see 3.1.64).

3.1.105 running disparity: A binary value indicating the cumulative encoded signal imbalance between the one and zero signal state of all characters since dword synchronization has been achieved (see 6.2).

3.1.106 SAS address: A worldwide unique name assigned to a SAS initiator port, SAS target port, expander device, SAS initiator device, or SAS target device. See 4.2.2.

3.1.107 SAS device: A SAS initiator device, SAS target device, or SAS target/initiator device.

3.1.108 SAS domain: The I/O system defined by this standard that may serve as an ATA domain and/or a SCSI domain. See 4.1.7.

3.1.109 SAS initiator device: A device containing SSP, STP, and/or SMP initiator ports in a SAS domain. See 4.1.4.

3.1.110 SAS initiator phy: A phy in a SAS initiator device.

3.1.111 SAS initiator port: An SSP initiator port, STP initiator port, and/or SMP initiator port in a SAS domain.

3.1.112 SAS phy: A phy in a SAS device that interfaces to a service delivery subsystem.

3.1.113 SAS port: A SAS initiator port, SAS target port, or SAS target/initiator port.

3.1.114 SAS target device: A device containing SSP, STP, and/or SMP target ports in a SAS domain. See 4.1.4.

3.1.115 SAS target phy: A phy in a SAS target device.

3.1.116 SAS target port: An SSP target port, STP target port, and/or SMP target port in a SAS domain.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.117 SAS target/initiator device: A device that has all the characteristics of a SAS target device and a SAS initiator device.

3.1.118 SAS target/initiator port: A port that has all the characteristics of a SAS target port and a SAS initiator port in a SAS domain.

3.1.119 SATA device: A storage device that contains a SATA device port in an ATA domain (analogous to a SCSI target device).

3.1.120 SATA device port: A storage device object in an ATA domain that interfaces to the service delivery subsystem with SATA (analogous to a SCSI target port).

3.1.121 SATA host: A host device containing a SATA host port in an ATA domain (analogous to a SCSI initiator device).

3.1.122 SATA host port: A host device object in an ATA domain that interfaces to the service delivery subsystem with SATA (analogous to a SCSI initiator port).

3.1.123 SATA phy: A phy in a SATA device that interfaces to a service delivery subsystem (analogous to a SAS phy).

3.1.124 scrambling: Modifying data by XORing each bit with a pattern generated by a linear feedback shift register to minimize repetitive character patterns. See 7.6.

3.1.125 SCSI device: A device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol. See SAM-3.

3.1.126 SCSI domain: An I/O system consisting of a set of SCSI devices that communicate with one another by means of a service delivery subsystem. See SAM-3.

3.1.127 SCSI initiator device: A SCSI device containing 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-3.

3.1.128 SCSI initiator port: A SCSI initiator device object that acts as the connection between application clients and the service delivery subsystem through which indications and responses are routed. See SAM-3.

3.1.129 SCSI port: A SCSI initiator port or a SCSI target port. See SAM-3.

3.1.130 SCSI target device: A 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-3.

3.1.131 SCSI target port: A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and the service delivery subsystem through which requests and confirmations are routed. See SAM-3.

3.1.132 SCSI target/initiator device: A device that has all the characteristics of a SCSI target device and a SCSI initiator device. See SAM-3.

3.1.133 SCSI target/initiator port: A SCSI target/initiator device object that has all the characteristics of a SCSI target port and a SCSI initiator port. See SAM-3.

3.1.134 Serial ATA (SATA): The protocol defined by ATA/ATAPI-7 V3 (see 2.3).

3.1.135 Serial ATA Tunneled Protocol (STP): The protocol defined in this standard used by STP initiator ports to communicate with STP target ports in a SAS domain.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.136 Serial Attached SCSI (SAS): The set of protocols and the interconnect defined by this standard.

3.1.137 Serial Management Protocol (SMP): The protocol defined in this standard used by SAS devices to communicate management information with other SAS devices in a SAS domain.

3.1.138 Serial SCSI Protocol (SSP): The protocol defined in this standard used by SCSI initiator ports to communicate with SCSI target ports in a SAS domain.

3.1.139 service delivery subsystem: The 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.

3.1.140 SMP initiator phy: A SAS initiator phy in an SMP initiator port.

3.1.141 SMP initiator port: A SAS initiator device object in a SAS domain that interfaces to the service delivery subsystem with SMP.

3.1.142 SMP phy: A SAS phy in an SMP port.

3.1.143 SMP port: An SMP initiator port, SMP target port, or SMP target/initiator port.

3.1.144 SMP target phy: A SAS target phy in an SMP target port.

3.1.145 SMP target port: A SAS target device object in a SAS domain that interfaces to the service delivery subsystem with SMP.

3.1.146 SMP target/initiator port: A port that has all the characteristics of an SMP initiator port and an SMP target port.

3.1.147 speed negotiation lock time (SNLT): The maximum time during a speed negotiation window for a transmitter to reply with ALIGN (1) (see 6.6.4.2).

3.1.148 speed negotiation sequence: A sequence in which two phys negotiate the operational physical link rate. See 4.4.

3.1.149 speed negotiation transmit time (SNTT): The time during which ALIGN (0) or ALIGN (1) is transmitted during the speed negotiation sequence (see 6.6.4.2).

3.1.150 spread spectrum clocking: The technique of modulating the operating frequency of a transmitted signal to reduce the measured peak amplitude of radiated emissions.

3.1.151 SSP initiator phy: A SAS initiator phy in an SSP initiator port.

3.1.152 SSP initiator port: A SCSI initiator port in a SAS domain that implements SSP.

3.1.153 SSP phy: A SAS phy in an SSP port.

3.1.154 SSP port: An SSP initiator port, SSP target port, or SSP target/initiator port.

3.1.155 SSP target phy: A SAS target phy in an SSP target port.

3.1.156 SSP target port: A SCSI target port in a SAS domain that implements SSP.

3.1.157 SSP target/initiator port: A port that has all the characteristics of an SSP initiator port and an SSP target port.

3.1.158 STP initiator phy: A SAS initiator phy in an STP initiator port.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.1.159 STP initiator port: A SAS initiator device object in a SAS domain that interfaces to the service delivery subsystem with STP.

3.1.160 STP phy: A SAS phy in an STP port.

3.1.161 STP port: An STP initiator port, STP target port, or STP target/initiator port.

3.1.162 STP target phy: A SAS target phy in an STP target port.

3.1.163 STP target port: A SAS target device object in a SAS domain that interfaces to the service delivery subsystem with STP.

3.1.164 STP target/initiator port: A port that has all the characteristics of an STP initiator port and an STP target port.

3.1.165 STP/SATA bridge: An 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.

3.1.166 subtractive routing attribute: The attribute of an edge expander phy that indicates it may be used by the expander connection manager to route connection requests not resolved using the direct routing method or table routing method. See 4.6.7.1.

3.1.167 subtractive routing method: The method the expander connection manager uses to route connection requests not resolved using the direct routing method or table routing method to an expander device. See 4.6.7.1.

3.1.168 table routing attribute: The attribute of an expander phy that indicates it may be used by the expander connection manager to route connection requests using an expander route table. See 4.6.7.1.

3.1.169 table routing method: The method the expander connection manager uses to route connection requests to an expander device using an expander route table. See 4.6.7.1.

3.1.170 task: An object within the logical unit representing the work associated with a command or group of linked commands.

3.1.171 task management function: A task manager service capable of being requested by an application client to affect the processing of one or more tasks. See SAM-3.

3.1.172 task manager: An agent within the device server that processes task management functions. See SAM-3.

3.1.173 total jitter: Measured jitter including deterministic jitter and random jitter.

3.1.174 transceiver: An object that contains both transmitter and receiver objects.

3.1.175 transmitter: The source or generator of a signal.

3.1.176 transmitter compliance transfer function (TCTF): The mathematical statement of the transfer function through which the transmitter shall be capable of producing acceptable signals as defined by a receive mask. See 5.3.11.

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

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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

3.1.179 transport protocol service request: A message 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.

3.1.180 transport protocol service response: A message passed from the application layer to the transport layer (i.e., from the SCSI device server to the SSP target port) that completes the SCSI transport protocol service.

3.1.181 unit interval (UI): The time required to transmit one bit on a physical link (e.g., 666,$\overline{6}$ ps at 1,5 Gbps and 333,$\overline{3}$ ps at 3,0 Gbps).

3.1.182 valid dword: A dword that is not an invalid dword.

3.1.183 virtual phy: A phy (see 3.1.86) that interfaces to another virtual phy inside the same device. See 4.1.2.

3.1.184 wide link: A group of physical links that attaches a wide port to another wide port. See 4.1.3.

3.1.185 wide port: A port that contains more than one phy. See 4.1.3.

3.2 Symbols and abbreviations

Units and abbreviations used in this standard:

Abbreviation	Meaning
AA	ATA application layer (see 10.3)
A.C.	alternating current
ACK	acknowledge primitive (see 7.2.6.1)
AIP	arbitration in progress primitive (see 7.2.5.1)
ATA	AT attachment (see 3.1.4)
ATAPI	AT attachment packet interface
ATA/ATAPI-7	AT Attachment with Packet Interface - 7 standard (see 2.3)
AWG	American wire gauge
AWT	arbitration wait time
BCH	Bose, Chaudhuri and Hocquenghem code (see 4.2.3)
BER	bit error rate
BIST	built in self test
BPP	broadcast primitive processor (see 3.1.9)
CDB	command descriptor block (see 3.1.13)
CJTPAT	compliant jitter test pattern (see 3.1.14)
CRC	cyclic redundancy check (see 3.1.20)
dB	decibel
D.C.	direct current (see 3.1.27)
Dxx.y	data character (see 3.1.22)
ECM	expander connection manager (see 3.1.39)
ECR	expander connection router (see 3.1.40)
EMI	electromagnetic interference

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Abbreviation	Meaning
EOAF	end of address frame primitive (see 7.2.5.6)
EOF	end of frame primitive (see 7.2.6.4)
FIS	frame information structure (see 3.1.51)
G1	generation 1 physical link rate (1,5 Gbps)
G2	generation 2 physical link rate (3,0 Gbps)
G3	generation 3 physical link rate (defined in a future version of this standard)
Gbps	gigabits per second (10^9 bits per second)
GHz	gigaherz (10^9 transitions per second)
IU	information unit (see 3.1.65)
kHz	kilohertz (10^3 bits per second)
Kxx.y	control character (see 3.1.19)
LED	light-emitting diode
LSB	least significant bit (see 3.1.68)
LUN	logical unit number
μA	microampere (10^{-6} amperes)
μs	microsecond (10^{-6} seconds)
MA	management application layer (see 10.4)
Mbaud	megabaud (10^6 transitions per second)
MBps	megabytes per second (10^6 bytes per second)
MHz	megahertz (10^6 bits per second)
MSB	most significant bit (see 3.1.75)
ms	millisecond (10^{-3} seconds)
MT	SMP transport layer state machines (see 9.4)
mV	millivolt (10^{-3} volts)
N/A	not applicable
NAA	name address authority
NAK	negative acknowledge primitive (see 7.2.6.5)
nF	nanofarad (10^{-9} Farads)
ns	nanosecond (10^{-9} seconds)
OOB	out-of-band
PL	port layer state machines (see 8.2)
PLL	phase lock loop
P-P	peak-to-peak
ppm	parts per million (10^{-6})
ps	picosecond (10^{-12} seconds)
ρ	reflection coefficient (rho)
RCDT	rate change delay time (see 3.1.99)
RRDY	receiver ready primitive (see 7.2.6.6)
Rx	receive
SA	SCSI application layer (see 10.2)
SAM-3	SCSI Architecture Model - 3 standard (see 2.2)
SAS	Serial Attached SCSI (see 3.1.136)

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Abbreviation	Meaning
SATA	Serial ATA (see 3.1.134)
SBC-2	SCSI Block Commands - 2 standard (see 2.2)
SCSI	Small Computer System Interface family of standards
SL	link layer for SAS phys state machines (see 7.14)
SL_IR	link layer identification and hard reset state machines (see 7.9.5)
SMP	Serial Management Protocol (see 3.1.137), or link layer for SMP phys state machines (see 7.18.4)
SNLT	speed negotiation lock time (see 3.1.147)
SNTT	speed negotiation transmit time (see 3.1.149)
SOAF	start of address frame primitive (see 7.2.5.12)
SOF	start of frame primitive (see 7.2.6.7)
SP	phy layer state machine (see 6.7)
SP_DWS	phy layer dword synchronization state machine (see 6.8)
SPC-3	SCSI Primary Commands - 3 standard (see 2.2)
SSP	Serial SCSI Protocol (see 3.1.138), or link layer for SSP phys state machines (see 7.16.7)
ST	SSP transport layer state machines (see 9.2)
STP	Serial ATA Tunneled Protocol (see 3.1.135), or link layer for STP phys state machines (see 7.17.7)
s	second (unit of time)
TCTF	transmitter compliance transfer function
TT	STP transport layer state machines (see 9.3)
Tx	transmit
UI	unit interval (see 3.1.181)
V	volt
VPD	vital product data (see 10.2.9)
XL	link layer for expander phys state machine (see 7.15)
XOR	exclusive logical OR
^	exclusive logical OR
x	multiplication
/	division

3.3 Keywords

3.3.1 ignored: A keyword used to describe an unused bit, byte, word, field or code value. The contents or value of an ignored bit, byte, word, field or code value shall not be examined by the receiving SCSI device and may be set to any value by the transmitting SCSI device.

3.3.2 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.

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

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

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.3.5 may not: Keywords that indicate flexibility of choice with no implied preference (equivalent to “may or may not”).

3.3.6 need not: Keywords indicating a feature that is not required to be implemented (equivalent to “is not required to”).

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

3.3.8 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

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

3.3.10 restricted: A keyword referring to bits, bytes, words, and fields that are set aside for use in other standards or for other data structures in this standard. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field for the purposes of the requirements defined in this standard.

3.3.11 shall: A keyword indicating a mandatory requirement (equivalent to “is required to”). Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.3.12 should: A keyword indicating flexibility of choice with a strongly preferred alternative (equivalent to “is strongly recommended”).

3.3.13 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 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The ISO convention of numbering is used (i.e., the thousands and higher multiples are separated by a space and a comma is used as the decimal point). Table 1 shows a comparison of the ISO and American numbering conventions.

Table 1 — ISO and American numbering conventions

ISO	American
0,6	0.6
3,141 592 65	3.14159265
1 000	1,000
1 323 462,95	1,323,462.95

Numbers that are not immediately followed by lower-case b or h are decimal values (e.g., 25).

Numbers immediately followed by lower-case b (e.g., 0101b) are binary values. Underscores may be included in binary values to increase readability or delineate field boundaries (e.g., 0101_1010b).

Numbers immediately followed by lower-case h (e.g., 15h) are hexadecimal values. A sequence of numbers and/or upper case letters 'A' through 'F' immediately followed by lower-case h (e.g., FA23h) are hexadecimal values. Underscores may be included in hexadecimal values to increase readability or delineate field boundaries (e.g., FD8C_FA23h).

Numbers after a decimal point with overlines represent infinitely repeating digits (e.g., 666,$\overline{6}$ means 666,666666... or 666 2/3).

Lists sequenced by letters (e.g., a) red, b) blue, c) green) show no ordering relationship between the listed items. Numbered lists (e.g., 1) red, 2) blue, 3) green) show an ordering between the listed items.

In the event of conflicting information the precedence for requirements defined in this standard is:

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

Notes do not constitute any requirements for implementers.

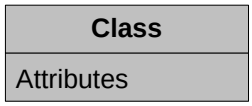
3.5 Object and class diagram conventions

The notation in figure 3 and figure 4 is based on the Unified Modeling Language (UML) specification and the Common Information Model (CIM) specification.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 3 shows how objects and classes of objects are described.

Class - a descriptor for a set of objects with similar structure, behavior, and relationships. A class has a class name and may have attributes. An object is a particular instance of a class.



Instantiation - shows an object with a specific combination of attributes from the indicated class.

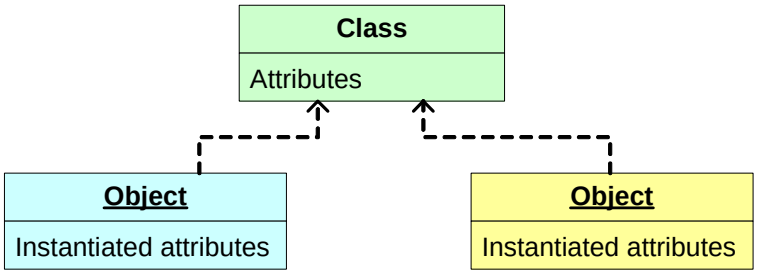
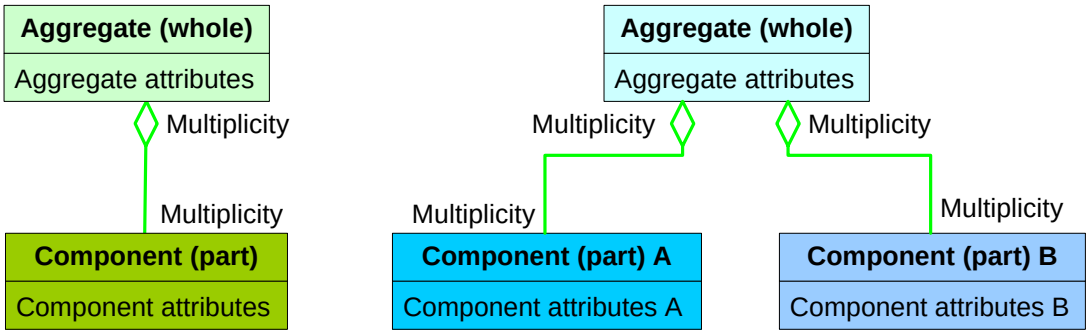


Figure 3 — Object and class diagram conventions

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 4 shows how classes of objects are related by aggregation (i.e., containment) and generalization (i.e., inheritance).

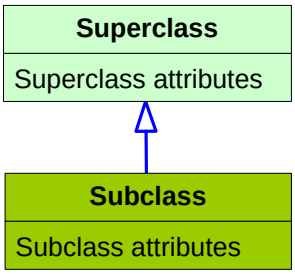
Aggregation - the aggregate (whole) class contains the component (part) classes(s). Component classes have access to all aggregate class attributes.



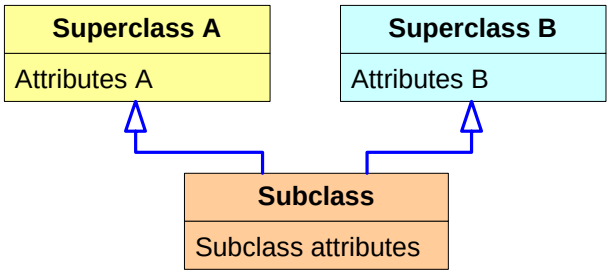
Multiplicity is a comma separated sequence of integer intervals in the form:
lower-bound .. upper-bound
where * may be used as an unlimited upper-bound.

Generalization - All members of a subclass are members of its superclass(es).
Each member of a subclass inherits all the attributes of its superclass(es).

Single superclass:



Subclass with multiple superclasses:



Multiple subclasses with one superclass:

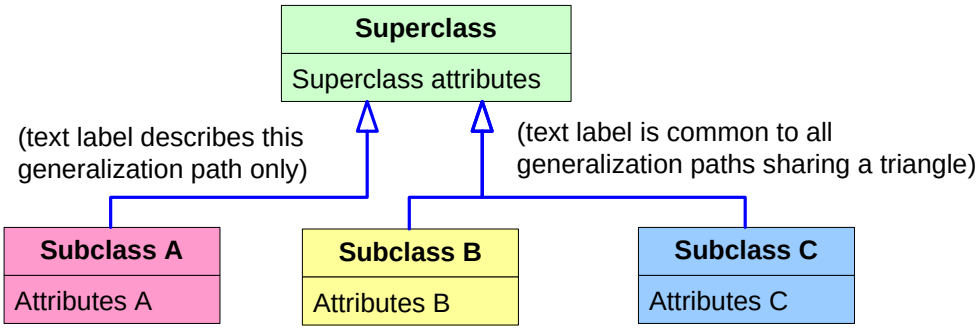


Figure 4 — Class diagram conventions for aggregation and generalization

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

3.6 State machine conventions

3.6.1 State machine conventions overview

Figure 5 shows how state machines are described. See 4.3 for a summary of the state machines in this standard.

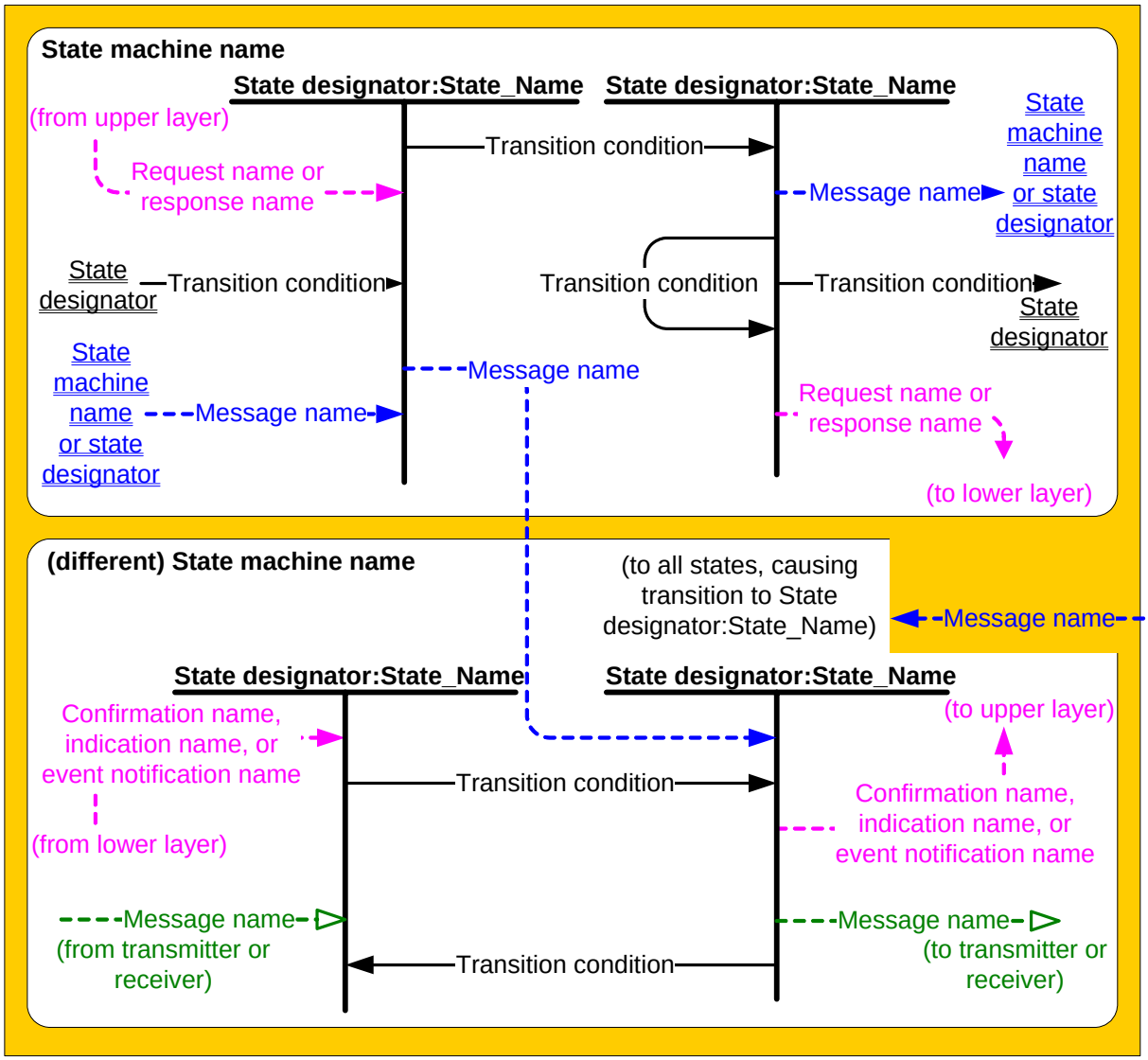


Figure 5 — State machine conventions

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

Each state is identified by a state designator and a state name. The state designator (e.g., SL1) is unique among all state machines in this 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, a brief description of the event or condition that causes the transition to occur.

If the state transition leaves the page, the transition label goes to or from a state designator label with double underlines rather than to or from a state.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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 to the top or bottom of the figure. Each request, indication, confirmation, response, and event notification is labeled. The meaning of each request, indication, confirmation, response, and event notification is described in the state description text.

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

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 1 - 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 with only two rows, with the non-sequentially increasing byte number indicating the presence of additional bytes.

A data dword consists of 32 bits. Table 2 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 2 — 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

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 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 3 — 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 enclosing optional or conditional parameters and arguments.

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

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

where:

Found = Flag
If set, indicates that a matching item was located.

Input Arguments:

Pattern = … /* Definition of Pattern object */
Object containing the search pattern.
Item List = Item<NN> /* Definition of Item List as an array of NN Item objects*/
This list contains the items to be searched for a match.

Output Arguments:

Item Found = Item … /* Item located by the search procedure */
This object is only returned if the search succeeds.

--`,,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4 General

4.1 Architecture

4.1.1 Architecture overview

A SAS domain (see 4.1.7) contains one or more SAS devices and a service delivery subsystem. A SAS domain may be a SCSI domain (see SAM-3).

A SAS device (see 4.1.4) contains one or more SAS ports (see 4.1.3). A SAS device may be a SCSI device (see SAM-3).

A SAS port (see 4.1.3) contains one or more phys (see 4.1.2). A SAS port may be a SCSI port (see SAM-3).

The service delivery subsystem (see 4.1.6) in a SAS domain may contain expander devices (see 4.1.5).

Expander devices contain expander ports (see 4.1.3) and one SMP port.

An expander port contains one or more phys (see 4.1.2).

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

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 6 shows the SAS object model, 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 objects. Relationships to ATA objects (e.g., ATA domain) are not shown in figure 6.

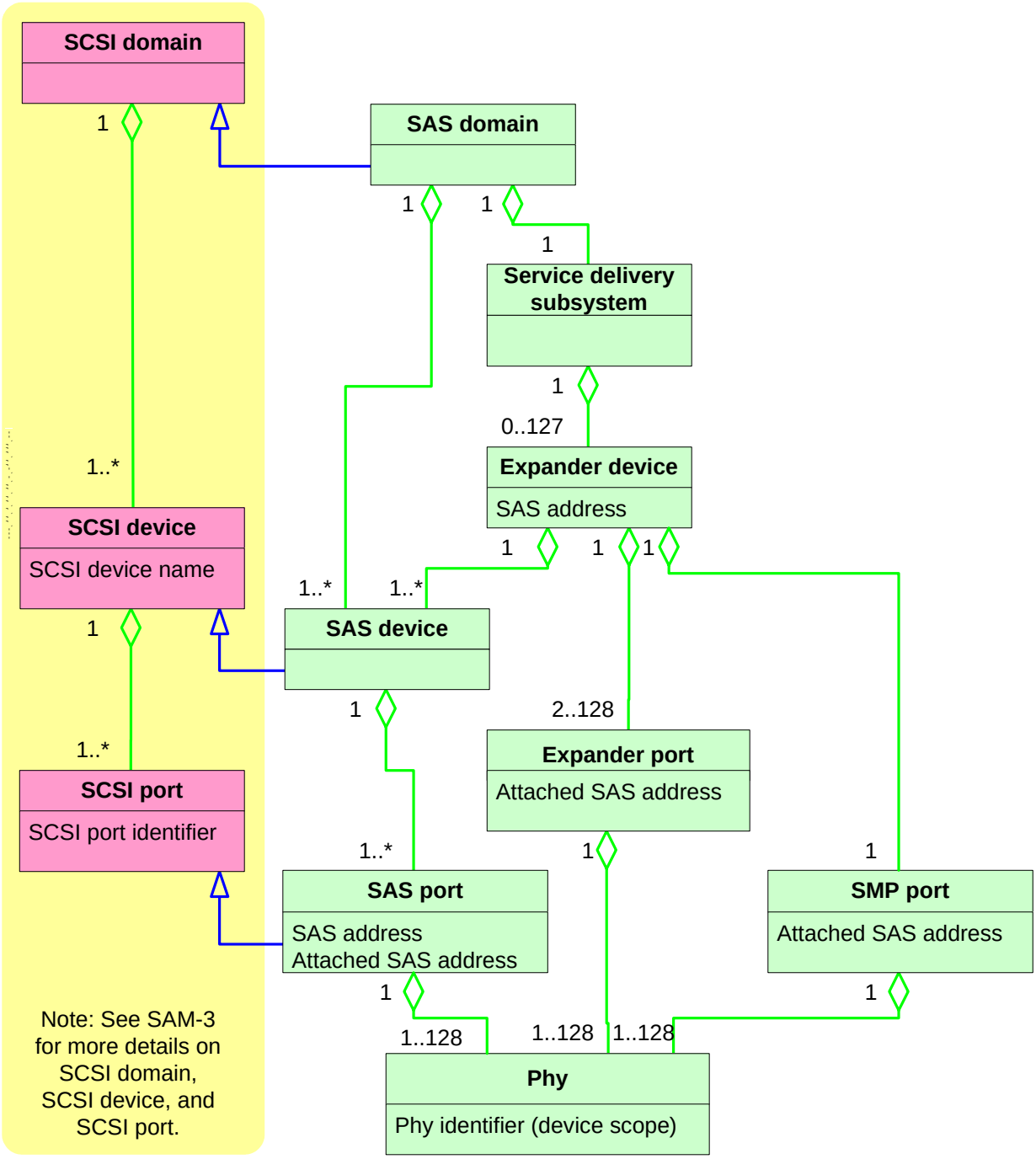


Figure 6 — SAS object model

4.1.2 Physical links and phys

A physical link is a set of four wires used as two differential signal pairs. One differential signal transmits in one direction while the other differential signal transmits in the opposite direction. Data may be transmitted in both directions simultaneously.

A physical phy contains a transceiver which electrically interfaces to a physical link, which attaches to another physical phy. A virtual phy contains a vendor-specific interface to another virtual phy.

Phys are contained in ports (see 4.1.3). Phys interface to the service delivery subsystem (see 4.1.6).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 7 shows two phys attached with a physical link.

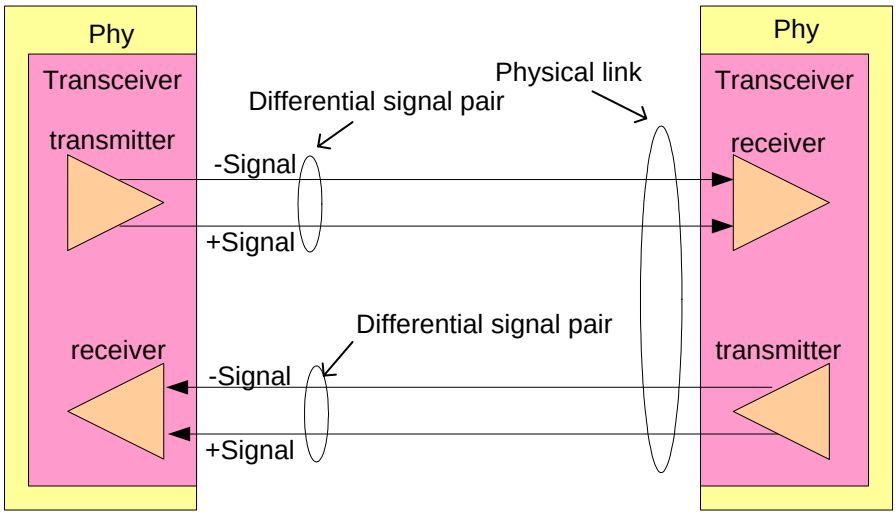


Figure 7 — Physical links and phys

An attached phy is the phy to which a phy is attached over a physical link.

A device may contain one or more phys. Each phy has a phy identifier (see 4.2.7) which is unique within the device.

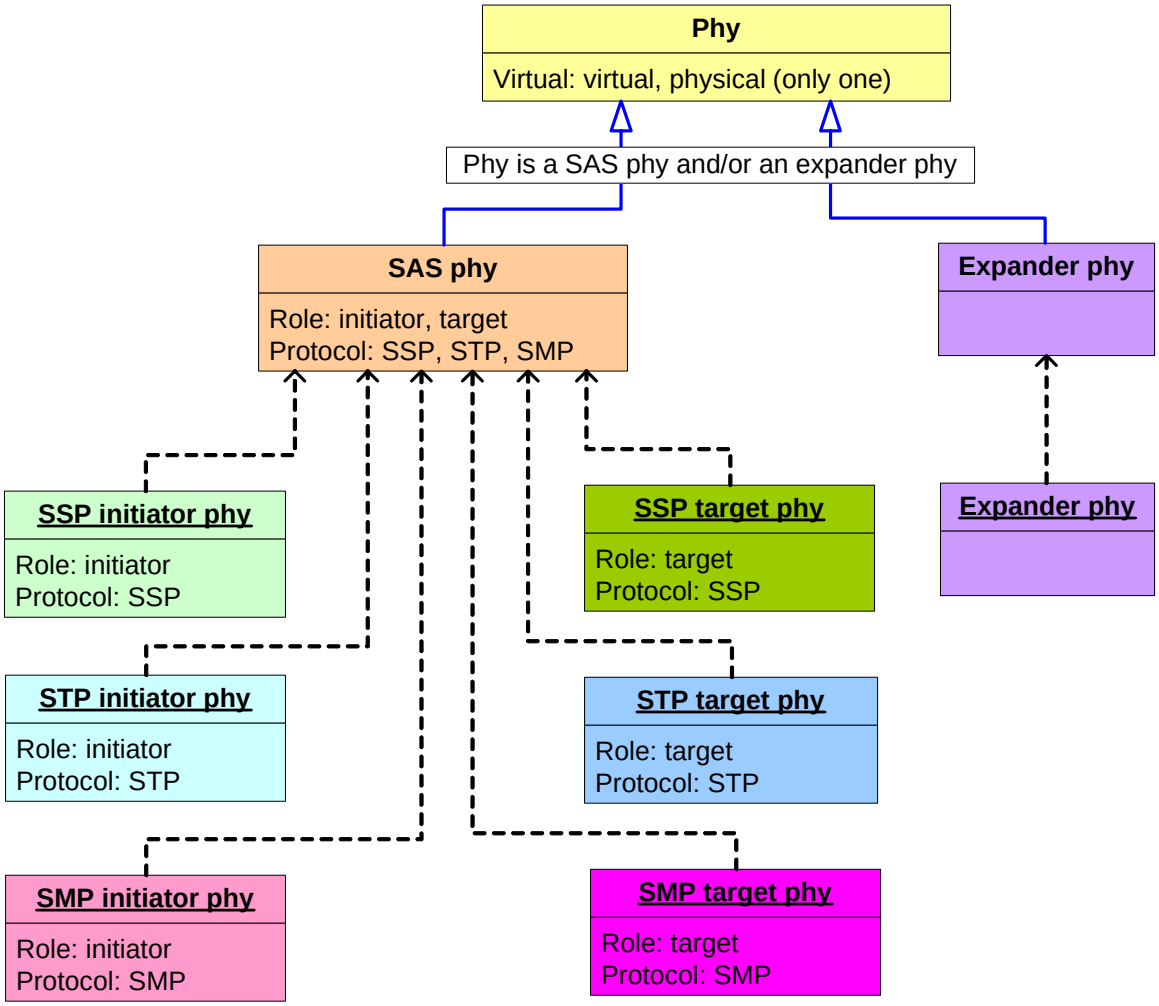
The transceiver follows the electrical specifications defined in 5.3. Phys transmit and receive bits at physical link rates defined in 5.3. The physical link rates supported by a phy are specified or indicated by the NEGOTIATED PHYSICAL LINK RATE field, HARDWARE MINIMUM PHYSICAL LINK RATE field, the HARDWARE MAXIMUM PHYSICAL LINK RATE field, the PROGRAMMED MINIMUM PHYSICAL LINK RATE field, and the PROGRAMMED MAXIMUM PHYSICAL LINK RATE field in the SMP DISCOVER function (see 10.4.3.5), SMP PHY CONTROL function (see 10.4.3.10), and Phy Control and Discover subpage (see 10.2.6.2.3). The bits are part of dwords (see 6.2.1), each of which has been encoded using 8b10b coding into four 10-bit characters (see 6.2).

Figure 8 shows the phy object classes, showing the relationships between phy, virtual phy, physical phy, SAS phy, expander phy, SAS initiator phy, SAS target phy, SSP phy, STP phy, SMP phy, SSP initiator phy, SSP

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

target phy, STP initiator phy, STP target phy, SMP initiator phy, and SMP target phy object classes. SATA phys are also referenced in this standard but are defined by SATA (see ATA/ATAPI-7 V3).



A SAS phy acts as 1 to 6 more-specific types of phys based on combinations of role/ protocol.

A SAS phy uses exactly one role and one protocol during a connection. It may use different roles and/or protocols in different connections.

Figure 8 — Phy object classes

4.1.3 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 7.9). Ports are associated with virtual phys based on the design of the device.

A port is created from a set of physical phys if one or more physical phys contained within a device:

a) transmit the same SAS address (see 4.2) during the identification sequence (see 7.9); and
b) receive the same SAS address during the identification sequence (i.e., the corresponding attached phy or phys transmit the same SAS address).

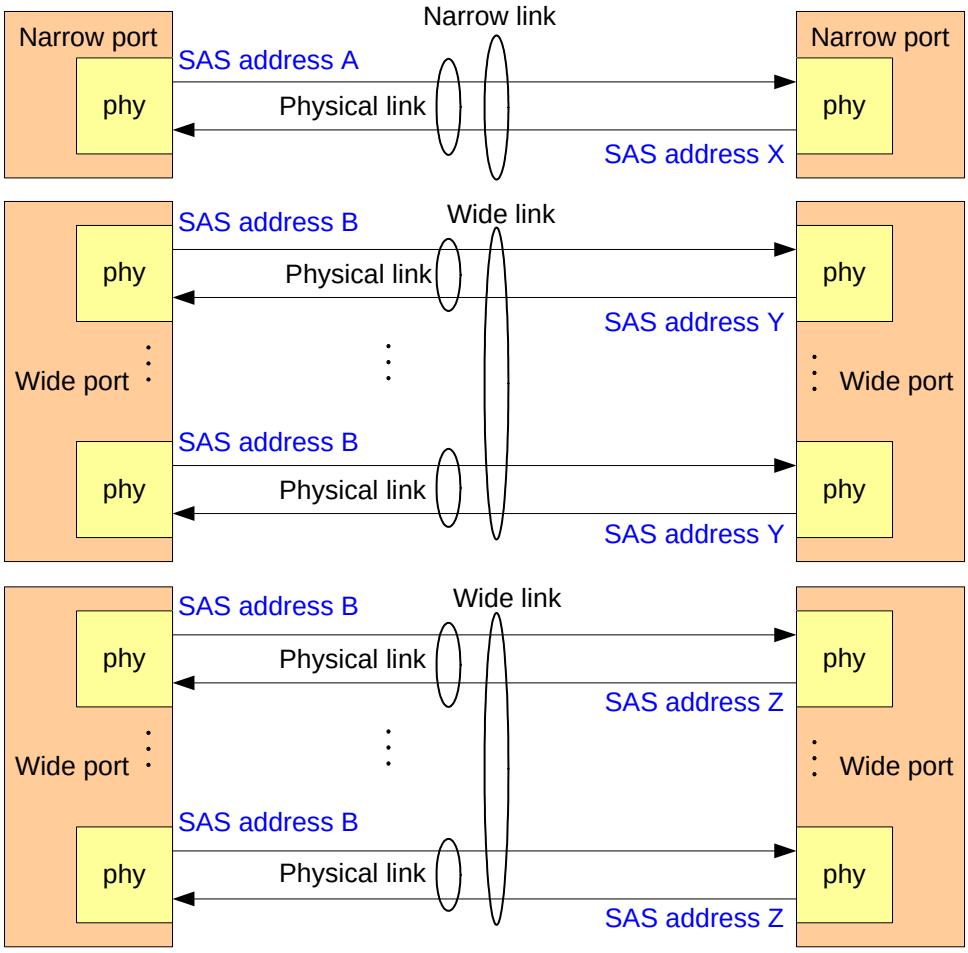
A wide port is created if there is more than one phy in the port. A narrow port is a port with only one phy.

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

Attaching phys within a wide port to other phys in the same wide port is outside the scope of this standard.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 9 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

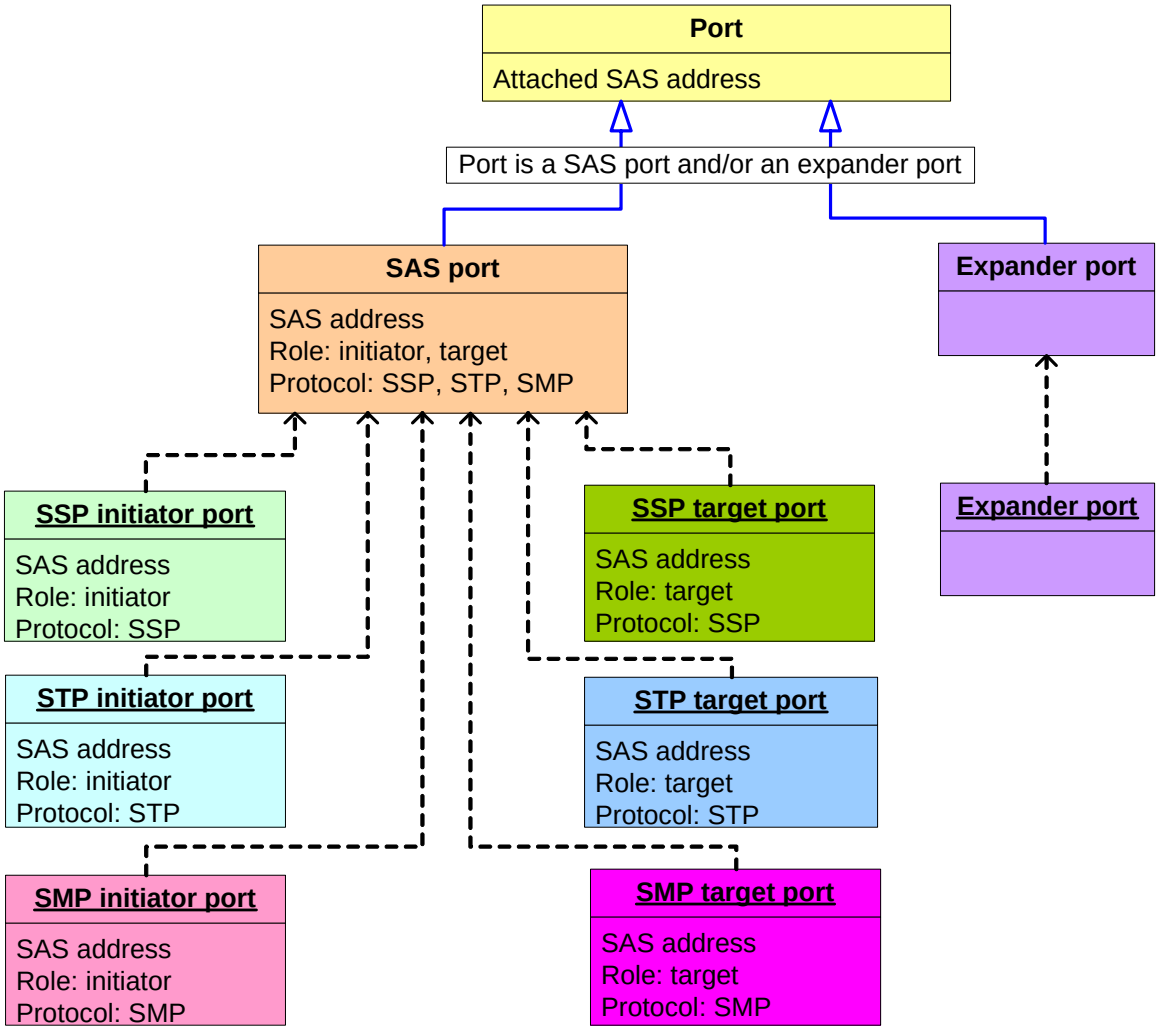
Figure 9 — Ports (narrow ports and wide ports)

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.

--`,,,`,,,--`--`,,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 10 shows the port object classes, showing the relationships between port, expander port, SAS port, SAS initiator port, SAS target port, SSP port, STP port, SMP port, SSP initiator port, STP initiator port, SMP initiator port, SSP target port, STP target port, and SMP target port object classes.



A SAS port acts as 1 to 6 more-specific types of ports based on combinations of role/ protocol.

A SAS port in an expander device (e.g., an SMP target port) inherits its SAS address from the expander device. An expander port inherits its SAS address from the expander device. SAS ports in SAS devices have their own SAS addresses.

Figure 10 — Port object classes

4.1.4 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).

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 11 shows examples of SAS devices with different port and phy configurations.

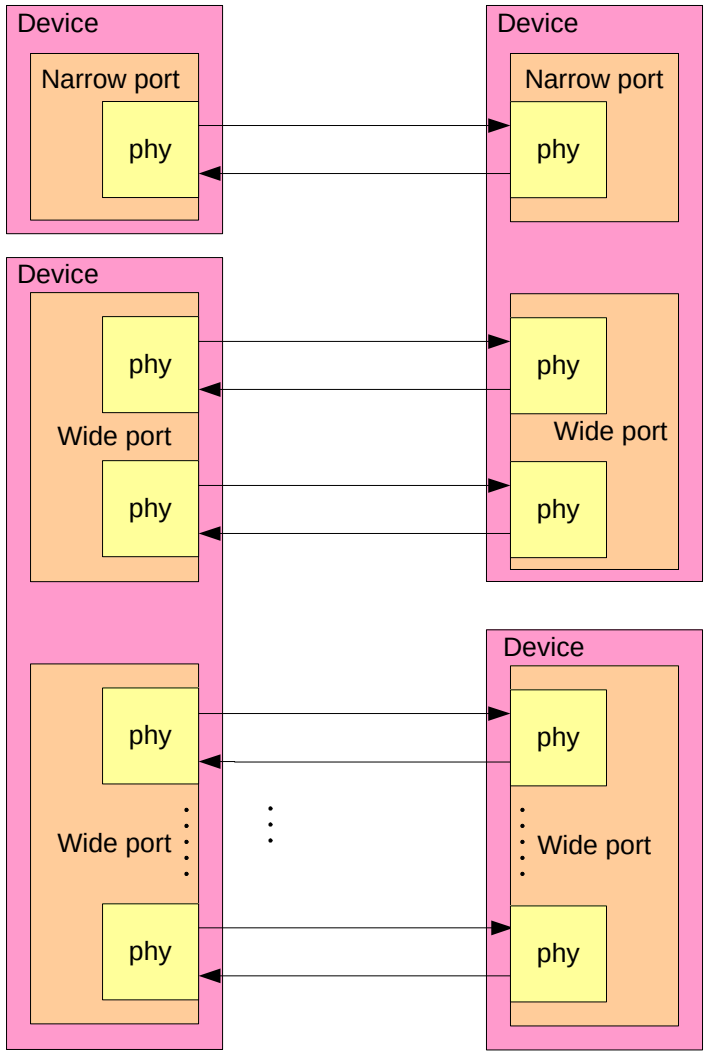


Figure 11 — SAS devices

4.1.5 Expander devices (edge expander devices and fanout expander devices)

Expander devices are part of the service delivery subsystem and facilitate communication between multiple SAS devices. Expander devices contain two or more external expander ports. Each expander device contains at least one SMP target port for management and may contain SAS devices (e.g., an expander device may include an SSP target port for access to a SCSI enclosure services logical unit).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 12 shows an expander device.

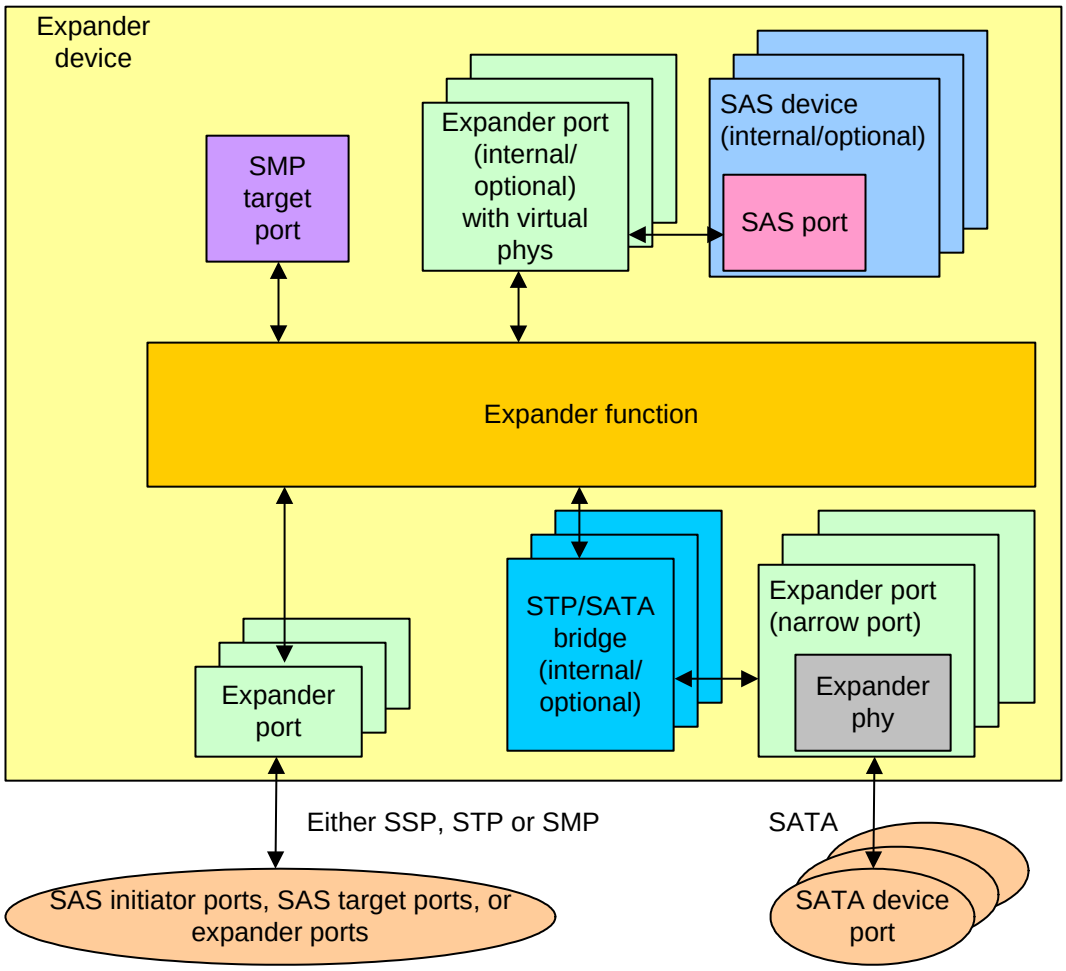


Figure 12 — Expander device

There are two types of expander devices:

a) edge expander devices; and
b) fanout expander devices.

Figure 13 shows the expander device object classes.

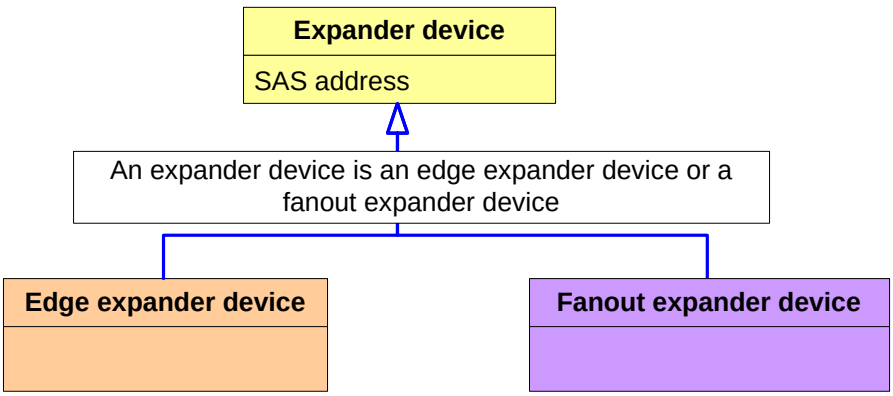


Figure 13 — Expander device object classes

An edge expander device is part of an edge expander device set (see 4.1.8.2). A fanout expander device is an expander device capable of being attached to two or more edge expander device sets.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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.

Edge expander devices may contain phys with the subtractive routing attribute. A fanout expander device shall not contain any phys with the subtractive routing attribute. Edge expander devices and fanout expander devices may also contain phys with direct routing and table routing attributes.

Expander devices with expander phys with the table routing attribute contain an expander route table. The expander route table may be configurable. An expander device with a configurable route table depends on a management application client within the SAS domain to use the discover process (see 4.6.7.4) to configure the expander route table. An expander device with expander phys with the table routing attribute that does not have a configurable route table shall be self-configuring, and shall contain a management application client and SMP initiator port to perform the discover process to configure its own expander route table.

4.1.6 Service delivery subsystem

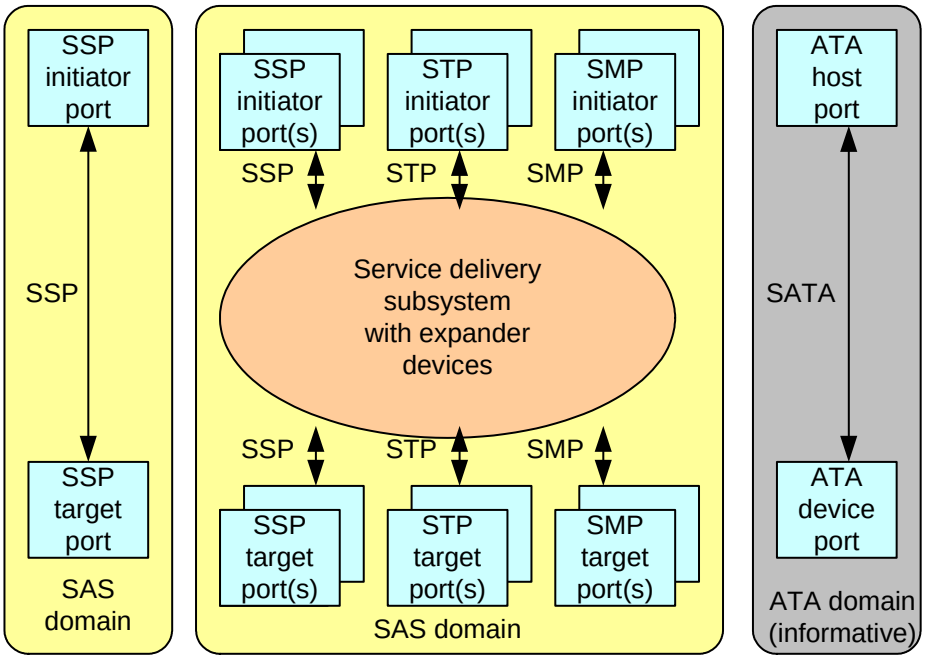
The service delivery subsystem is either:

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

4.1.7 Domains

Figure 14 shows examples of SAS domains (defined by this standard) and an ATA domain (defined by SATA).



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

Figure 14 — Domains

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 15 shows a SAS domain bridging to one or more ATA domains.

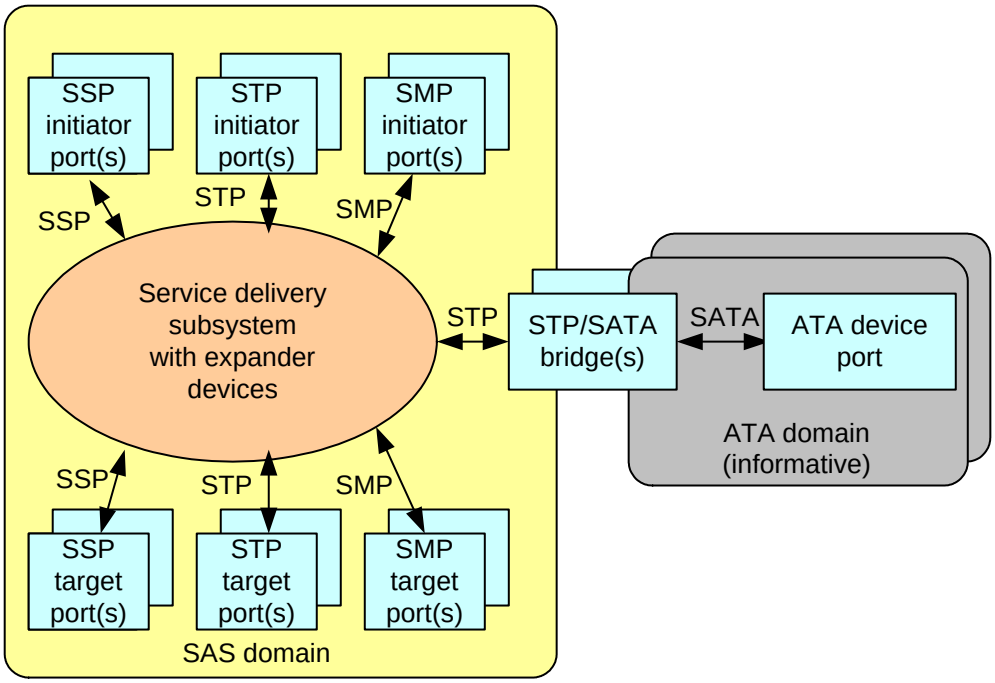


Figure 15 — SAS domain bridging to ATA domains

Figure 16 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 different SAS domains, the ports may have the same SAS address (see 4.2); when its ports are in the same SAS domain, they shall have different SAS addresses.

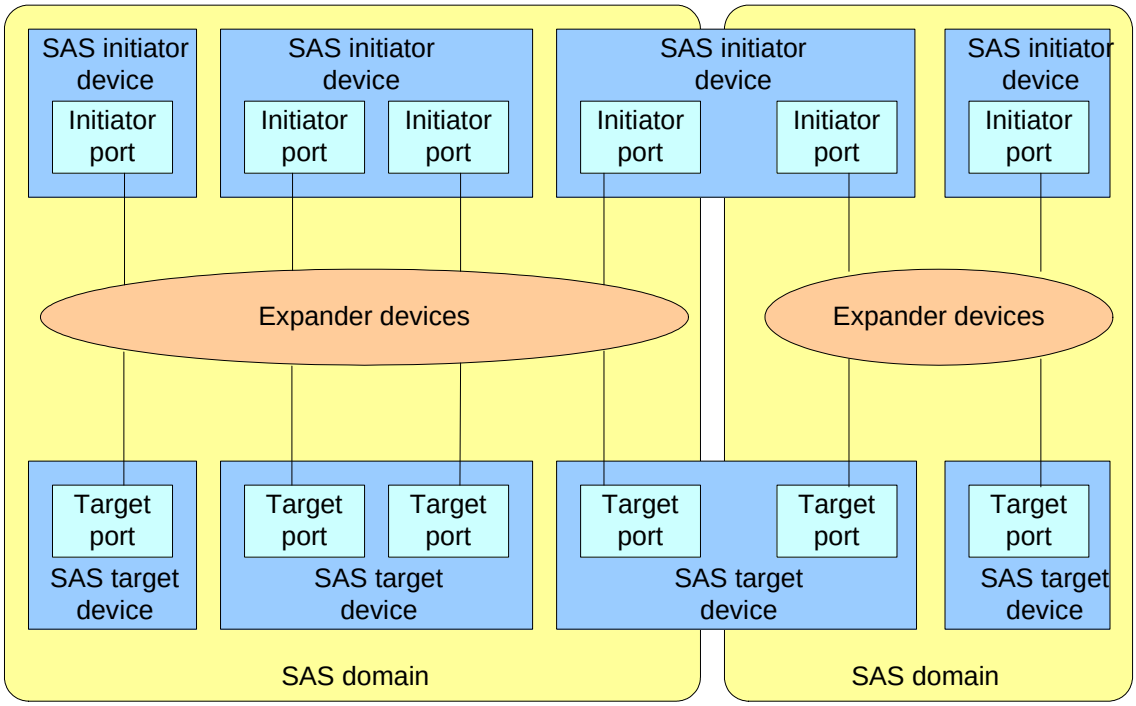


Figure 16 — Devices spanning SAS domains

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.1.8 Expander device topologies

4.1.8.1 Expander device topology overview

More than one expander device may be part of a service delivery subsystem. Some of them may be grouped into edge expander device sets.

4.1.8.2 Edge expander device set

One or more edge expander devices may be grouped into an edge expander device set. The number of edge expander devices and the routing attributes of the expander phys in the edge expander devices within an edge expander device set shall be established when the edge expander device set is constructed. The method used to construct edge expander sets is outside the scope of this standard.

An edge expander device set is bounded by expander phys with the direct routing attribute that are either:

a) attached to end devices; or
b) not attached to any device,

and by expander phys with the subtractive routing attribute that are:

a) attached to expander phys of a fanout device;
b) attached to expander phys with subtractive routing attributes on another edge expander device set;
c) attached to an end device; or
d) not attached to any device.

Phys with table routing attributes within an edge expander device set are used to provide attachments between edge expander devices in the edge expander device set.

The number of SAS addresses used by an edge expander device set shall not exceed 128. This includes SAS addresses for:

a) the edge expander devices within the edge expander device set;
b) SAS devices contained in the expander devices;
c) STP target ports in STP/SATA bridges contained in the expander devices; and
d) the end devices potentially attached to the edge expander device set.

To avoid an overflow of an edge expander route index during the discover process (see 4.6.7.4), an edge expander device set shall be constructed such that the number of edge expander route indexes available per phy identifier is greater than or equal to the sum of all SAS addresses addressable through the edge expander phy.

An edge expander device set shall not be attached to more than one fanout expander device.

An edge expander device set may be attached to one other edge expander device set if that is the only other edge expander device set in the SAS domain, they are attached using expander phys with subtractive routing attributes, and there are no fanout expander devices in the SAS domain.

--`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 17 shows an edge expander device set.

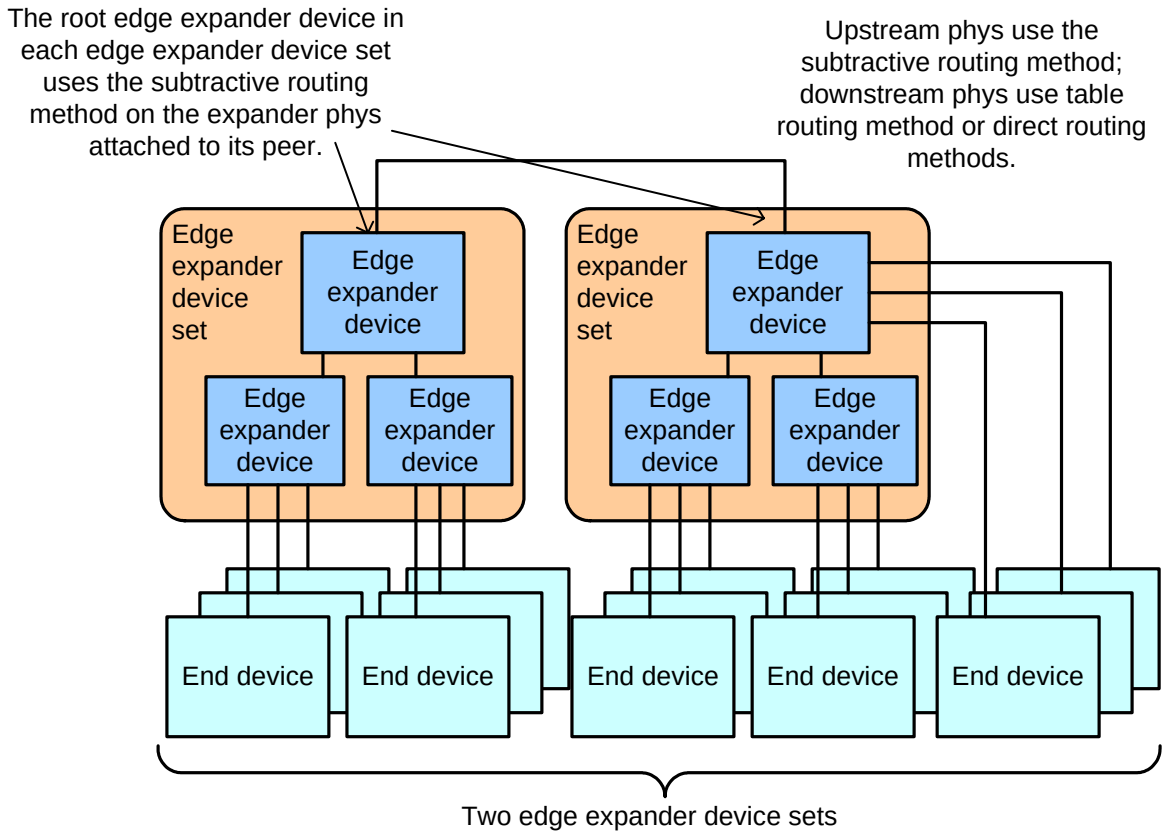


Figure 17 — Edge expander device set

4.1.8.3 Expander device topologies

No more than one fanout expander device shall be included in a SAS domain. The fanout expander device may be attached to up to 128 edge expander device sets or SAS ports.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 18 shows a SAS domain with the maximum number of expander device sets.

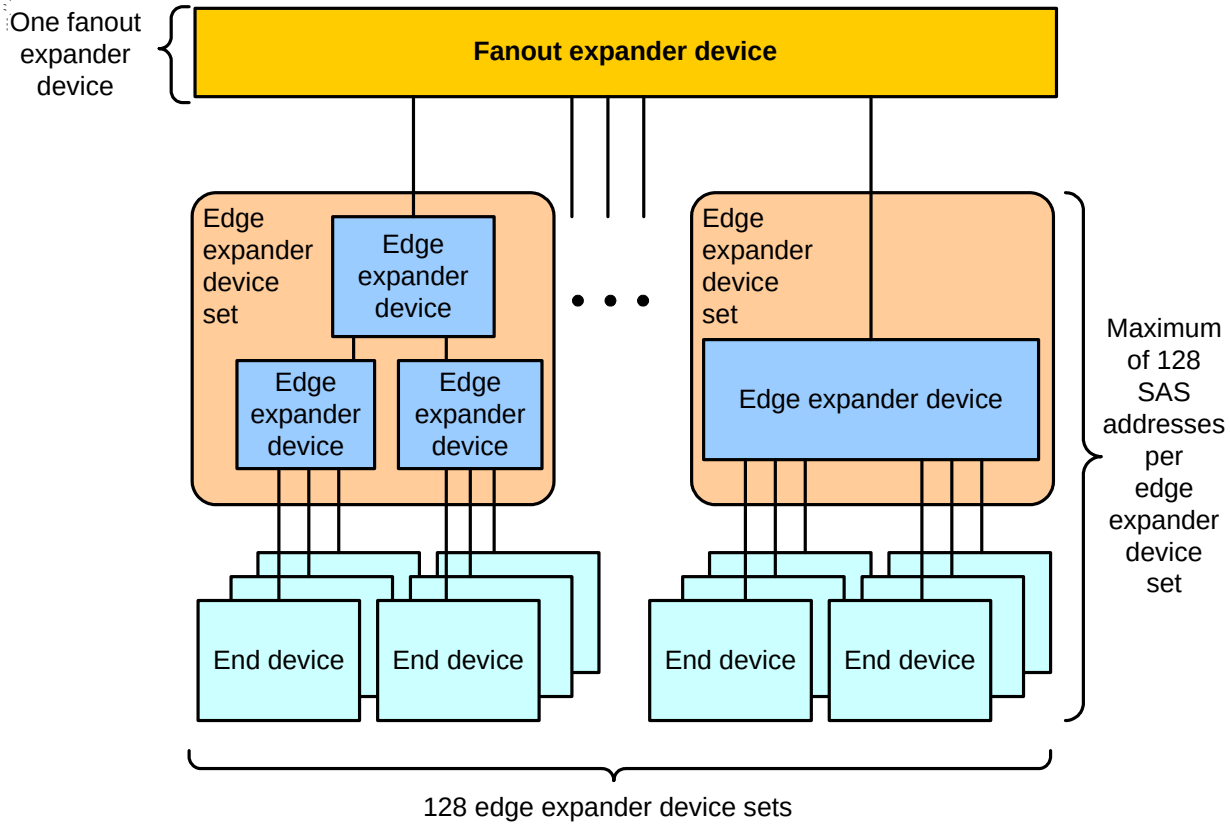


Figure 18 — Maximum expander device set topology

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

End devices may be attached directly to the fanout expander device, as shown in figure 19.

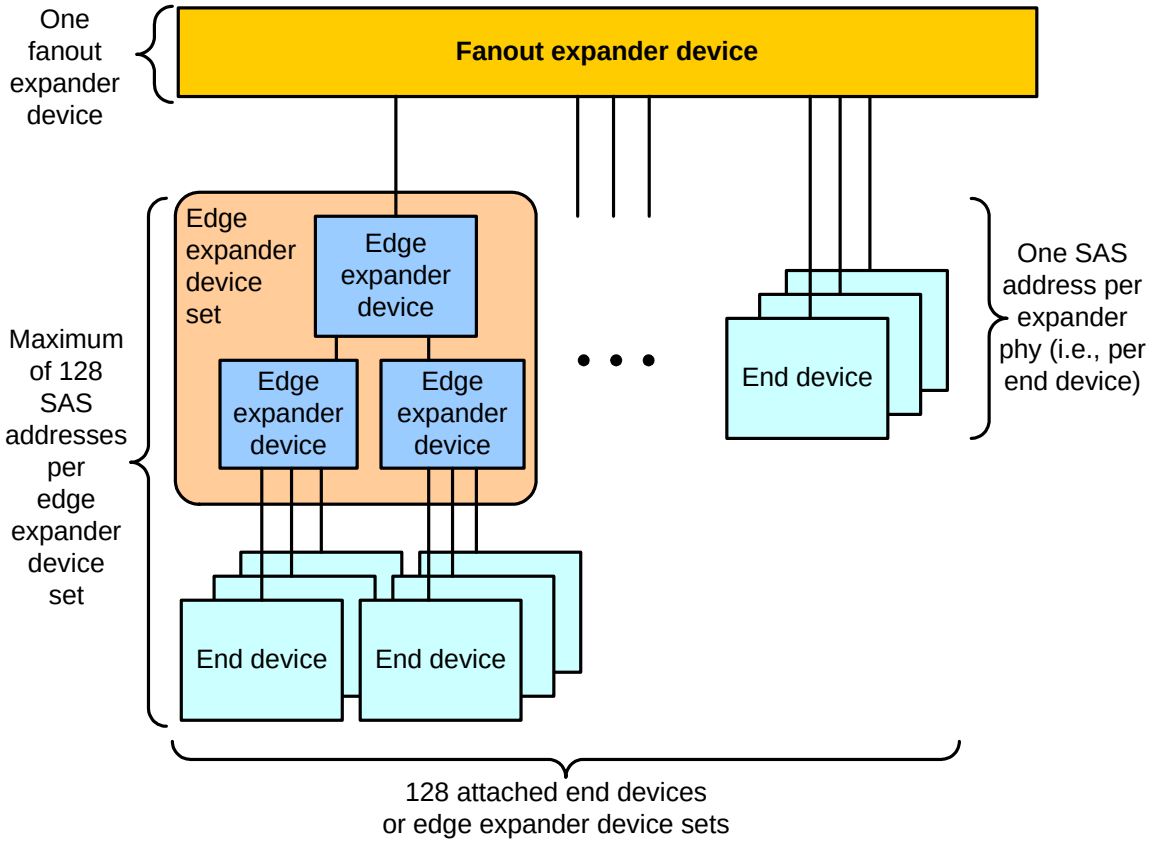


Figure 19 — Fanout expander device topology

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SAS initiator devices and SAS target devices may be attached to any of the edge expander devices within an edge expander device set and at most, two edge expander device sets may be attached together without a fanout expander, as shown in figure 20.

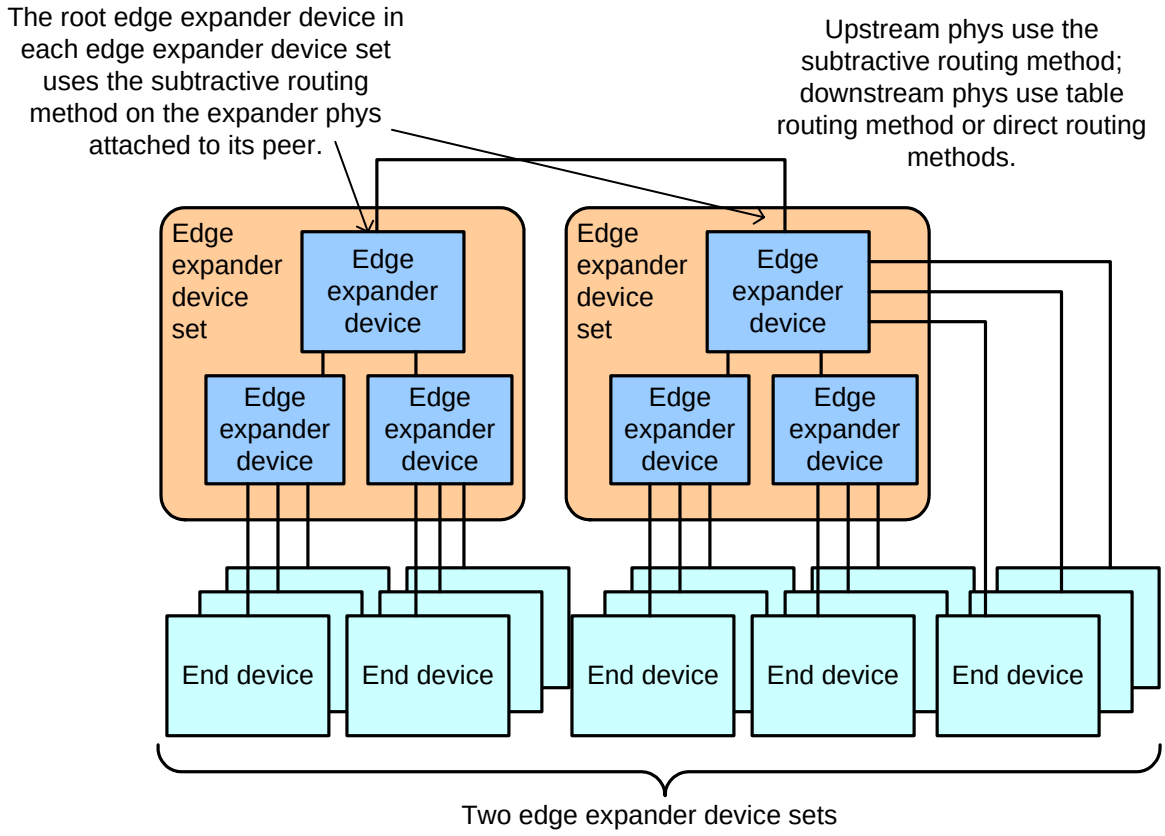


Figure 20 — Edge expander device set to edge expander device set topology

4.1.9 Pathways

A potential pathway is a set of physical links between a SAS initiator phy and a SAS target phy. When a SAS initiator phy is directly attached to a SAS target phy, there is one potential pathway. When there are expander devices between a SAS initiator phy and a SAS target phy, there are multiple potential pathways, each consisting of a set of physical 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 physical links between a SAS initiator phy and a SAS target phy being used by a connection (see 4.1.10).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 21 shows examples of potential pathways.

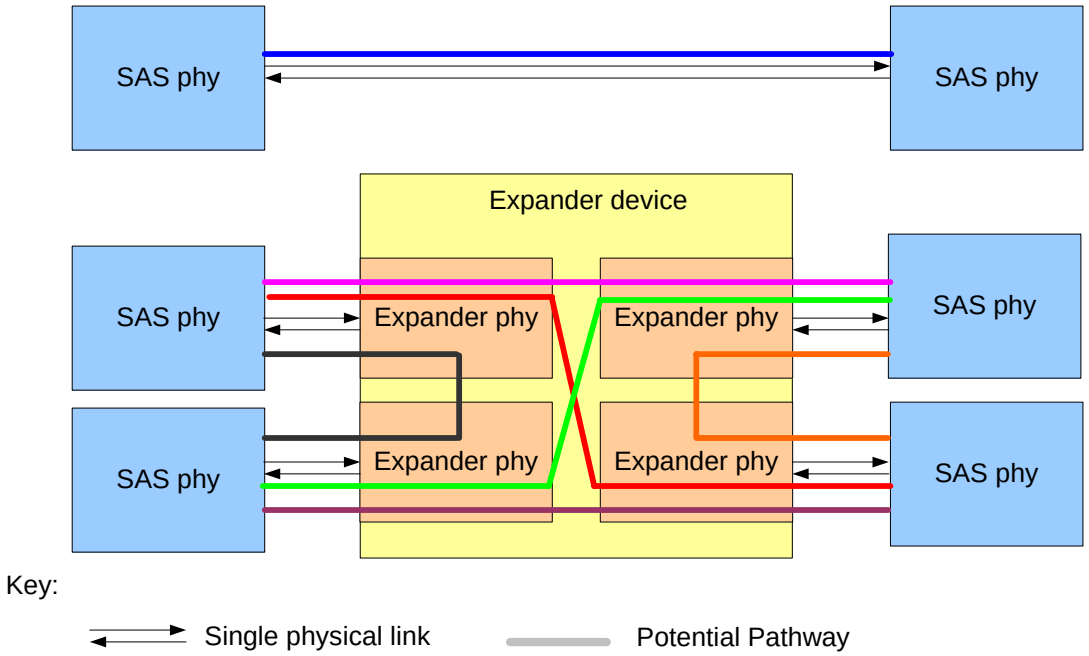


Figure 21 — Potential pathways

A partial pathway is the set of physical links participating in a connection request that has not reached the destination phy (e.g., the OPEN address frame has been transmitted by the source phy but the OPEN address frame has not yet reached the destination phy)(see 7.12).

A partial pathway is blocked when path resources it requires are held by either another connection or another partial pathway (see 7.12).

4.1.10 Connections

A connection is a temporary association between a SAS initiator port and a SAS target port. During a connection all dwords from the SAS initiator port are forwarded to the SAS target port, and all dwords from the SAS target port are forwarded to the SAS initiator port.

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 an OPEN_ACCEPT is returned to the source 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 7.12).

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 phy shall support a 1,5 Gbps connection rate regardless of its physical link rate.

One connection may be active on a physical 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, SATA_SYNCs, SATA_CONTs, or vendor-specific scrambled data dwords (after a SATA_CONT) are transmitted. If there is no connection on a physical link then idle dwords are transmitted.

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

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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:

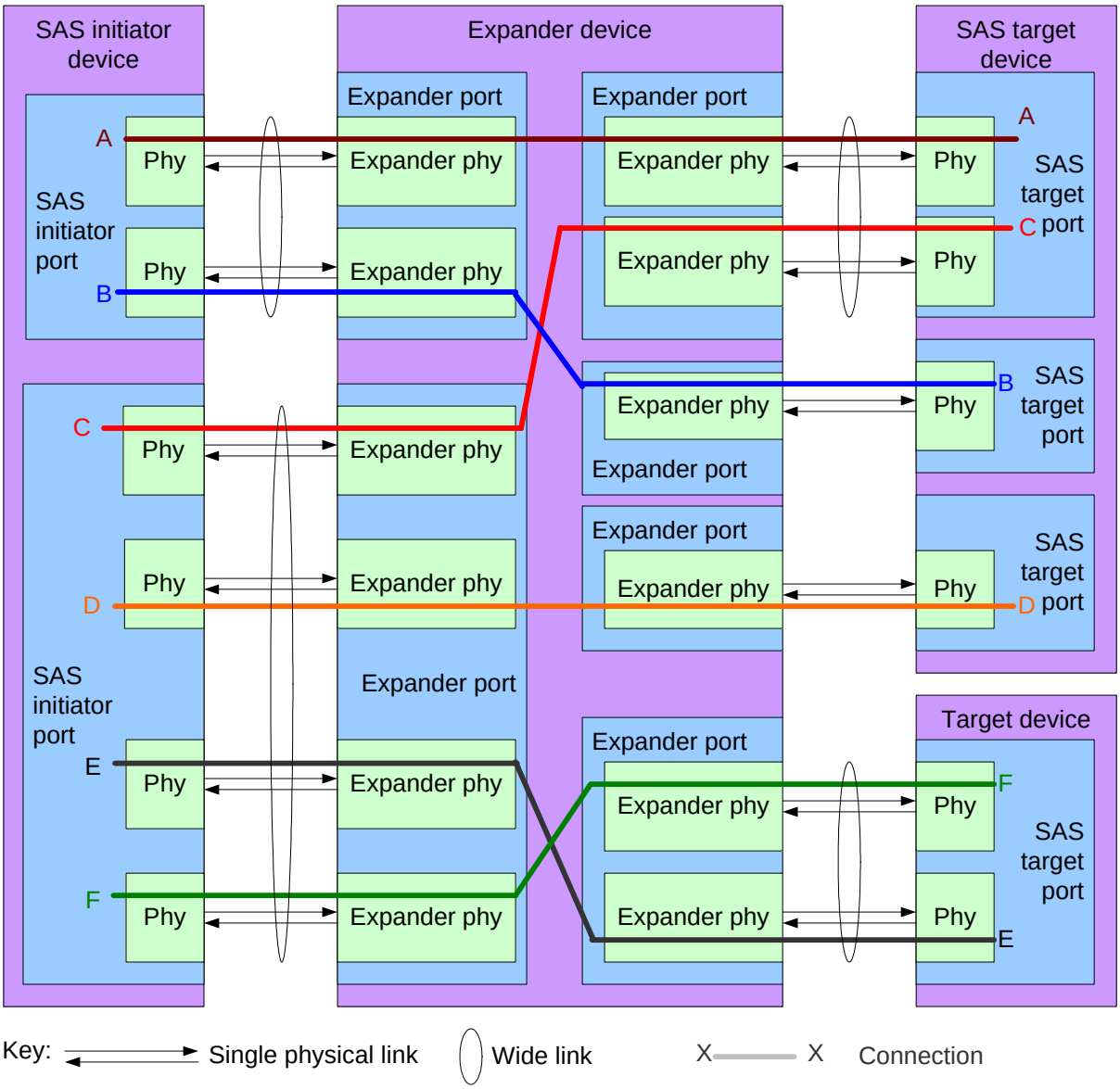
a) one SAS initiator port to multiple SAS target ports;
b) one SAS target port to multiple SAS initiator ports; or
c) one SAS initiator port to one SAS target port.

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

Figure 22 shows examples of connections between wide and narrow ports. All the connections shown may occur simultaneously. Additionally:

a) the connections labeled A and B are an example of one SAS initiator port with connections to multiple SAS target ports;
b) the connections labeled A and C are an example of one SAS target port with connections to multiple SAS initiator ports;
c) the connections labeled E and F are an example of multiple connections between one SAS initiator port and one SAS target port; and
d) the connections 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale



Notes: The expander device has a unique SAS address. Each SAS initiator port and SAS target port has a unique SAS address. Connections E and F represent a wide SAS initiator port with two simultaneous connections to a wide SAS target port.

Figure 22 — Multiple connections on wide ports

4.2 Names and identifiers

4.2.1 Names and identifiers overview

Device names are worldwide unique names for devices within a transport protocol (see SAM-3). 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, and are used as SAS addresses. Phy identifiers are unique within a device.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 4 describes the definitions of names and identifiers for SAS.

Table 4 — Names and identifiers

Object	SAS implementation
Port identifier	SAS address
Port name	Not defined
Device name	SAS address
Phy identifier	Phy identifier

Table 5 describes how various SAM-3 objects are implemented in SSP.

Table 5 — SAM-3 object mapping

SAM-3 object	SSP implementation
Initiator port identifier	SAS address of SSP initiator port
Initiator port name	Not defined
Target port identifier	SAS address of SSP target port
Target port name	Not defined
Device name	SAS address of SCSI device

4.2.2 SAS addresses

Table 6 defines the SAS address format. SAS addresses shall be compatible with the NAA (Name Address Authority) IEEE Registered format identification descriptor defined in SPC-3.

Table 6 — SAS address format

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

The NAA field contains 5h.

The IEEE COMPANY ID field contains a 24-bit canonical form company identifier assigned by the IEEE. Information about IEEE company identifiers may be obtained from the http://standards.ieee.org/regauth/oui web site.

The VENDOR-SPECIFIC IDENTIFIER contains a 36-bit numeric value that is uniquely assigned by the organization associated with the company identifier in the IEEE COMPANY ID field.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The SAS address shall be worldwide unique. A SAS address of 00000000_00000000h indicates an invalid SAS address.

4.2.3 Hashed SAS address

SSP frames include a hashed version of the SAS address 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 7 lists the parameters for the code.

Table 7 — Hashed SAS address code parameter

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 information on SAS address hashing.

4.2.4 Device names

Each expander device, SAS initiator device, SAS target device, and SAS target/initiator device shall include a SAS address (see 4.2.2) as its device name. The selected SAS address shall be used for no other name or identifier.

Expander devices report their device names in the IDENTIFY address frame (see 7.8.2).

Logical units accessed through SSP target ports report SAS target device names through SCSI vital product data (see 10.2.9).

NOTE 2 - There is no way to retrieve SAS initiator device names defined in this standard.

4.2.5 Port names

Port names as defined by SCSI are not defined in SAS.

NOTE 3 - The SAS address used by SSP target ports in different SAS domains may be the same (e.g., when a set of phys transmit the same SAS address in the identification sequence but receive different SAS addresses, indicating they are attached to two separate SAS domains) so it serves as a port identifier rather than a port name.

4.2.6 Port identifiers

Each SAS initiator port, SAS target port, and SAS target/initiator port shall include a SAS address (see 4.2.2) as its port identifier. The selected SAS address shall be used for no other name or identifier.

SAS ports report their port identifiers in the IDENTIFY address frame (see 7.8.2). Port identifiers are used as source and destination addresses in the OPEN address frame (see 7.8.3).

Logical units accessed through SSP target ports report SAS target port identifiers through SCSI vital product data (see 10.2.9).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4.2.7 Phy identifiers

Each SAS phy and expander phy shall be assigned an identifier that is unique within the SAS device and/or expander device. The phy identifier is used for management functions (see 10.4).

Phy identifiers shall be greater than or equal to 00h and less than 80h.

4.3 State machines

4.3.1 State machine overview

Figure 23 shows the state machines for SAS devices, their relationships to each other and to the SAS device, SAS port, and SAS phy objects.

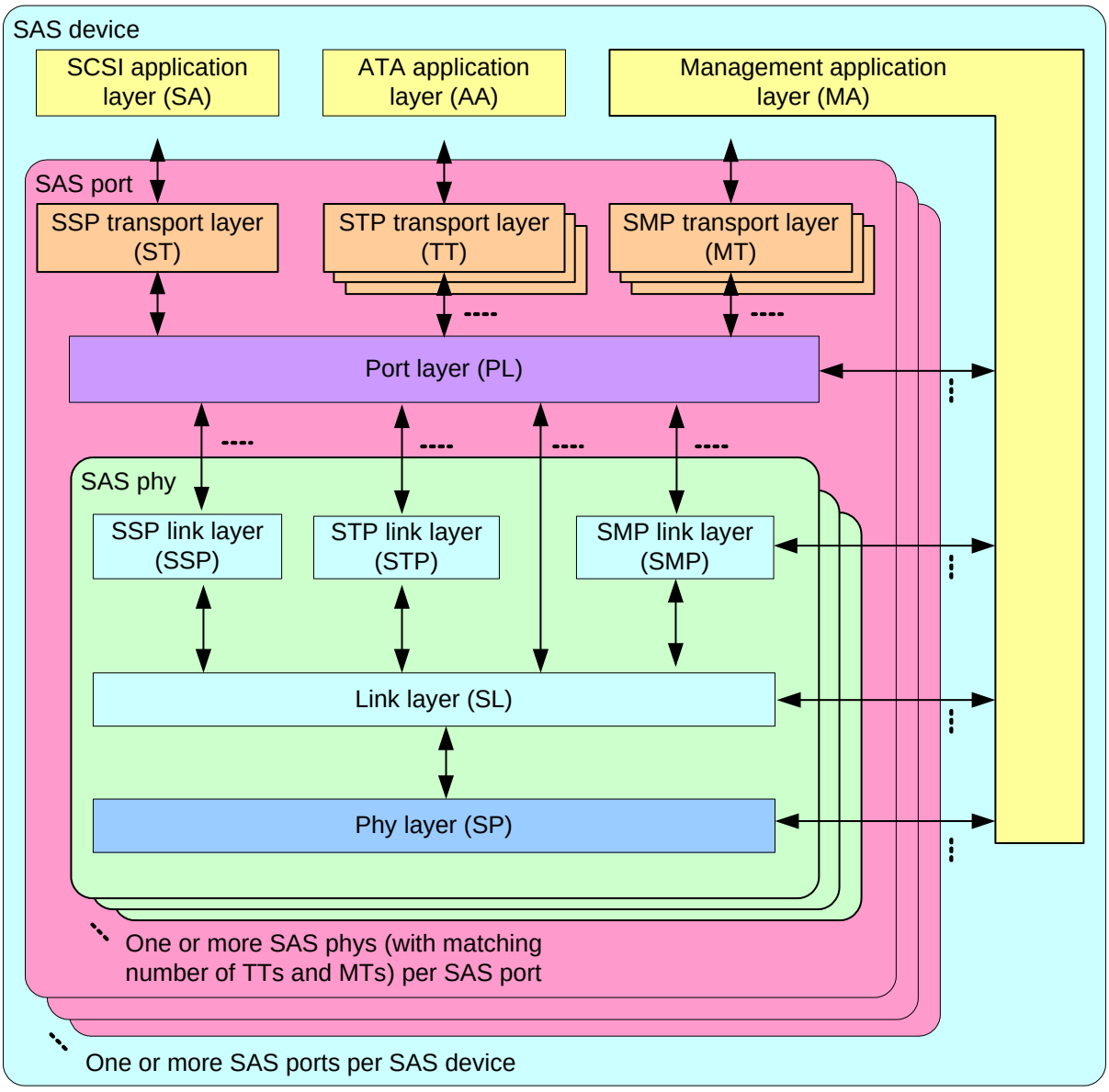


Figure 23 — State machines for SAS devices

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 24 shows the state machines for expander devices, their relationships to each other and to the expander device, expander port, and expander phy objects. Expander function state machines are not defined in this standard, but the interface to the expander function is defined in 4.6.6.

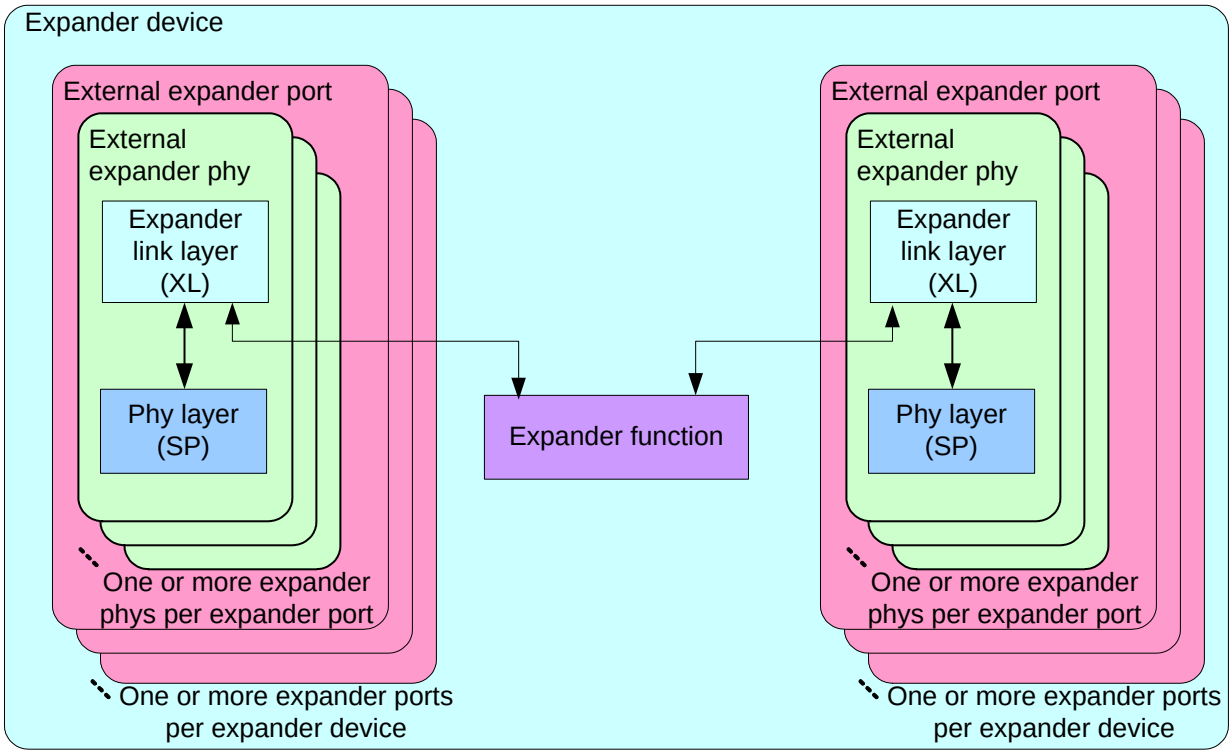


Figure 24 — State machines for expander devices

Annex I contains a list of messages between state machines.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.3.2 Transmit data path

Figure 25 shows the transmit data path in a SAS phy.

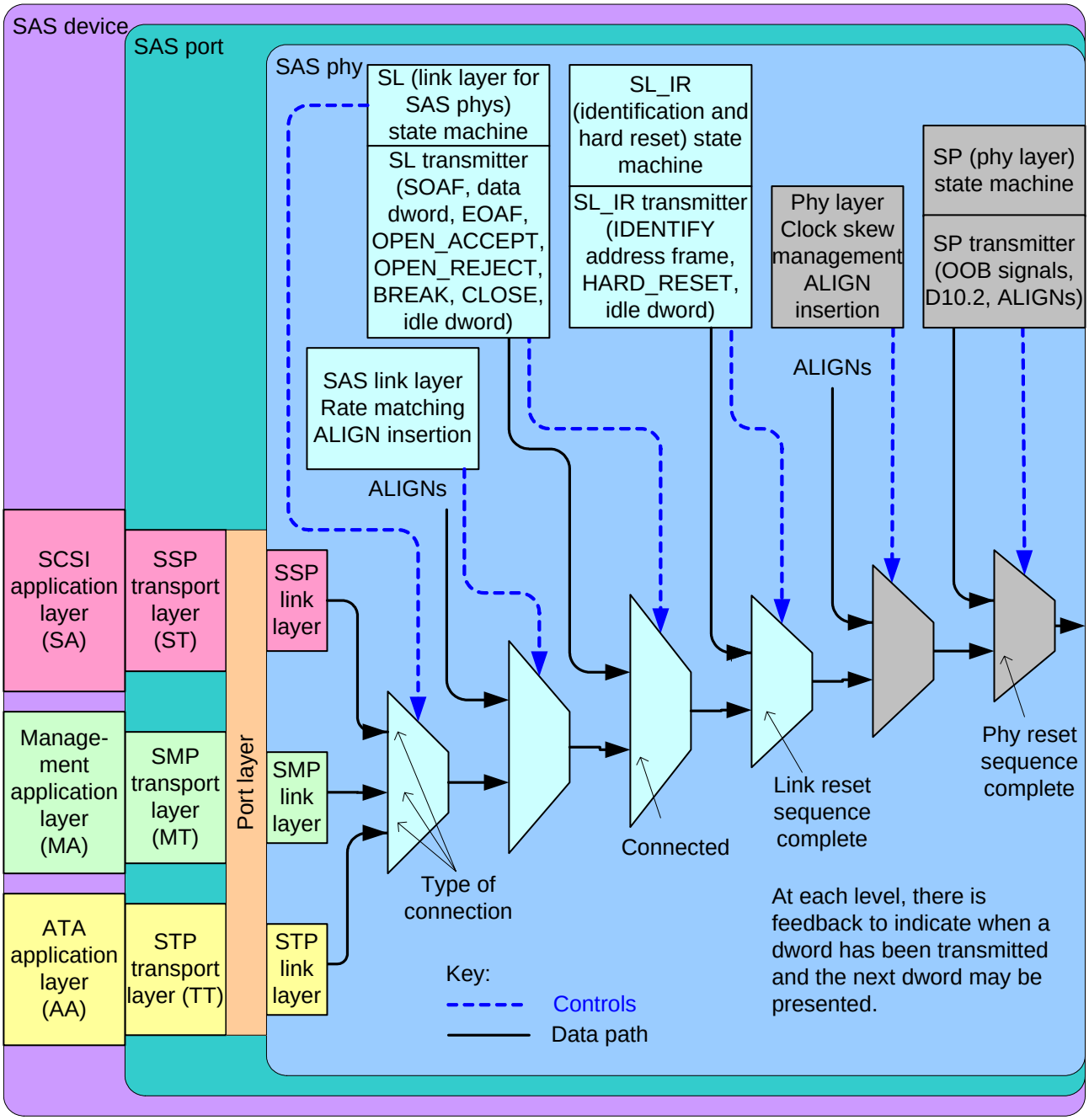


Figure 25 — Transmit data path in a SAS phy

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 26 shows the transmit data path for the SSP link layer 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.

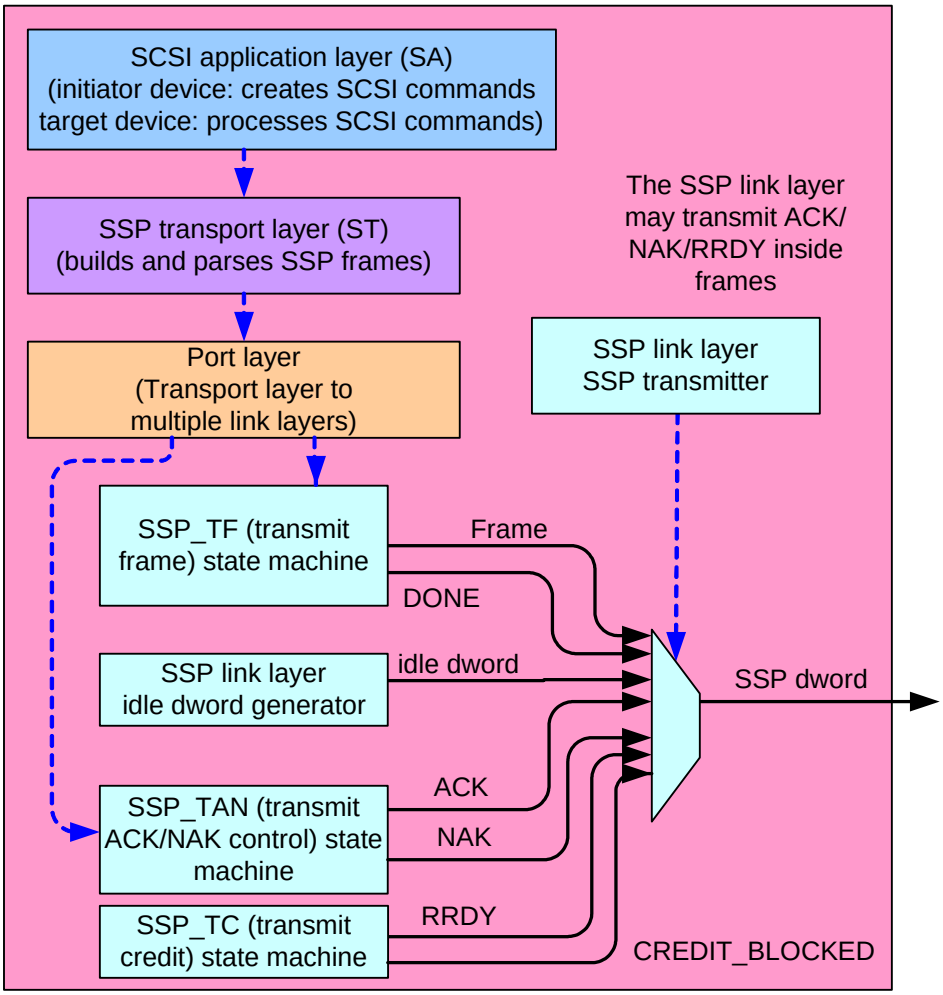


Figure 26 — SSP link, port, SSP transport, and SCSI application layer state machines

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 27 shows the transmit data path for the SMP link layer 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.

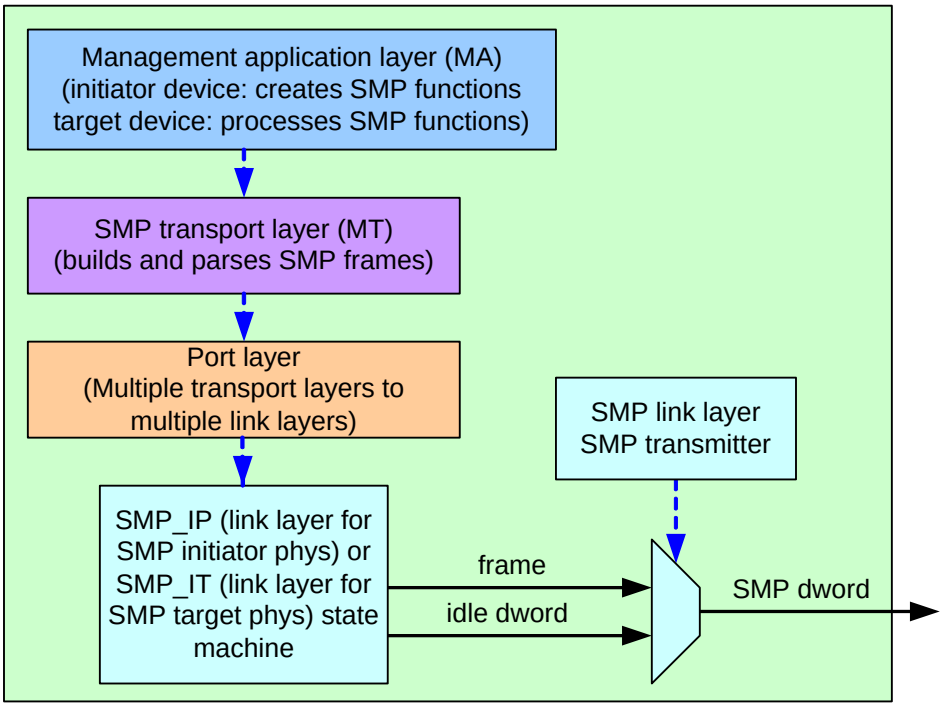


Figure 27 — SMP link, port, SMP transport, and management application layer state machines

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 28 shows the transmit data path for the STP link layer 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.

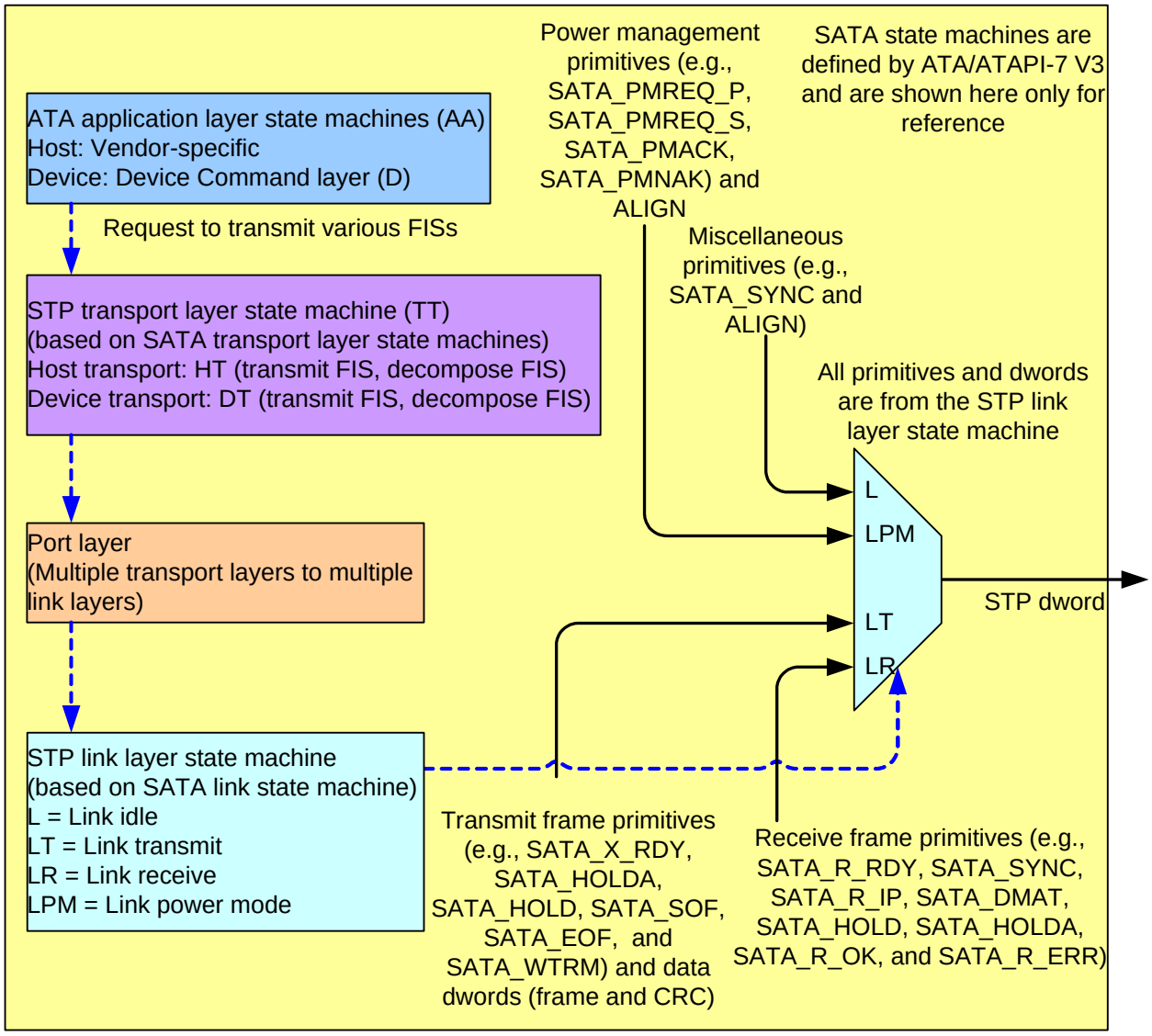


Figure 28 — STP link, port, STP transport, and ATA application layer state machines

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 29 shows the transmit data path in an expander phy.

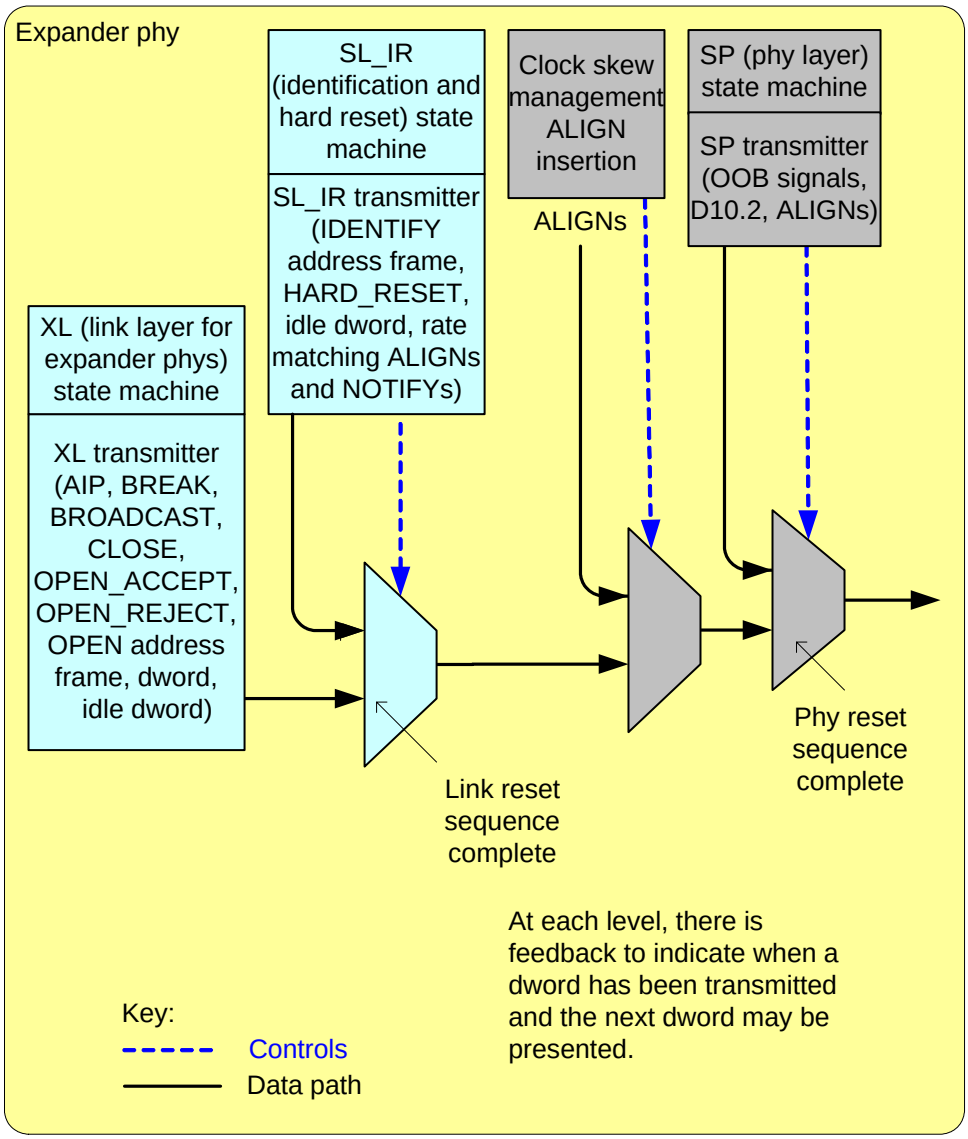


Figure 29 — Transmit data path and state machines in an expander phy

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.3.3 State machines and SAS device, SAS port, and SAS phy objects

Figure 30 shows which state machines are contained within the SAS device, SAS port, and SAS phy objects.

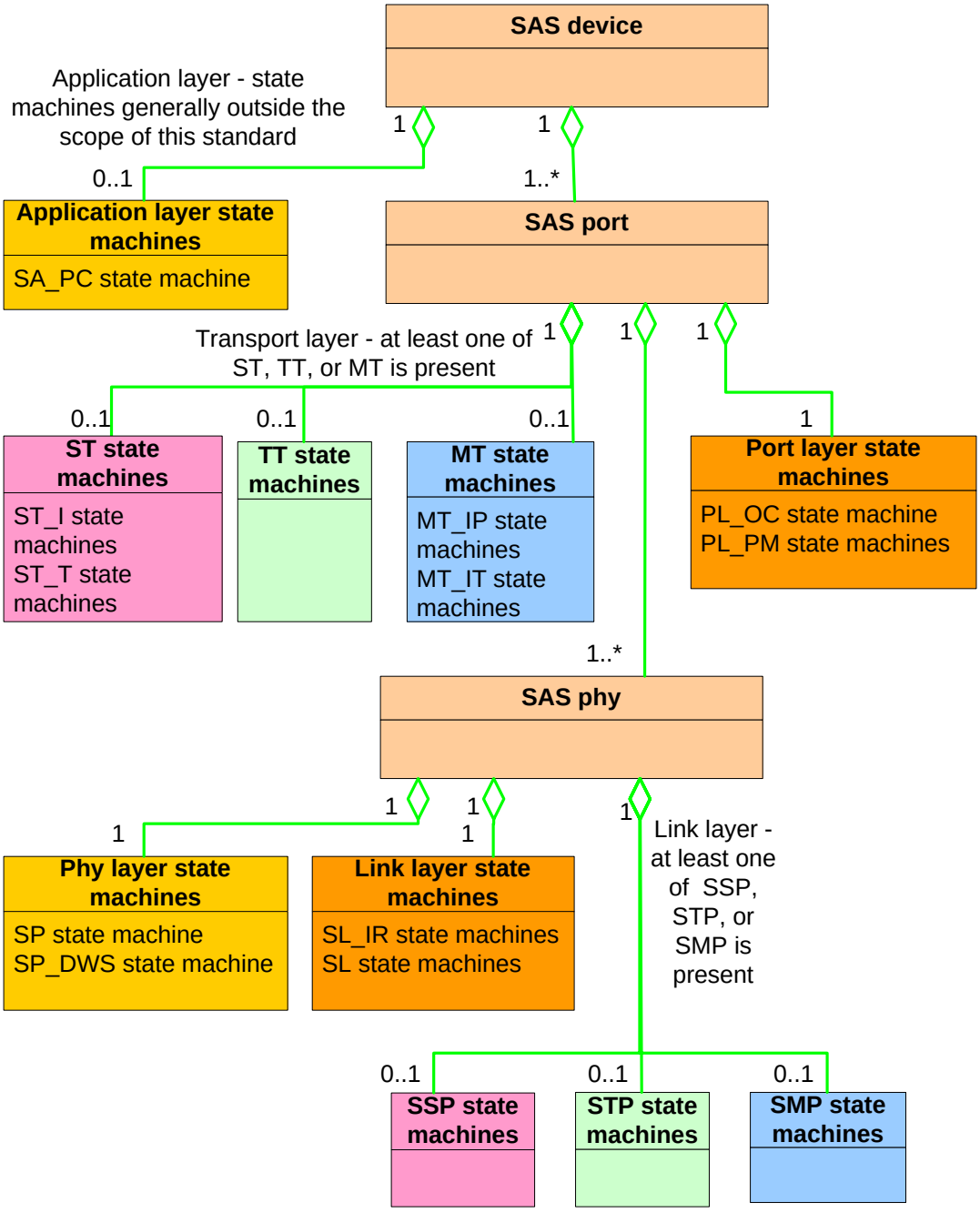


Figure 30 — State machine and SAS device, SAS port, and SAS phy objects

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 31 shows which state machines are contained within the expander device, expander port, and expander phy objects.

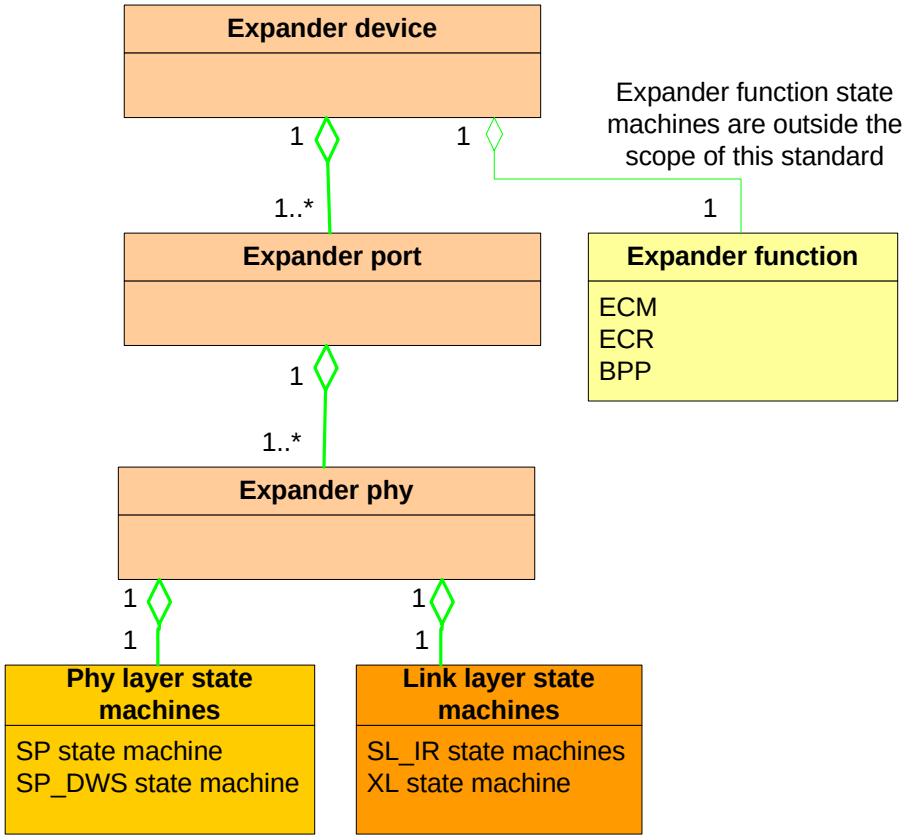


Figure 31 — State machine and expander device, expander port, and expander phy objects

4.4 Resets

4.4.1 Reset overview

Figure 32 describes the reset terminology used in this standard:

a) link reset sequence;
b) phy reset sequence (see 6.6);
c) SATA OOB sequence (see 6.6.2.1);
d) SATA speed negotiation sequence (see 6.6.2.2);
e) SAS OOB sequence (see 6.6.4.1);
f) SAS speed negotiation sequence (see 6.6.4.2);
g) hard reset sequence (see 7.9); and
h) identification sequence (see 7.9).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

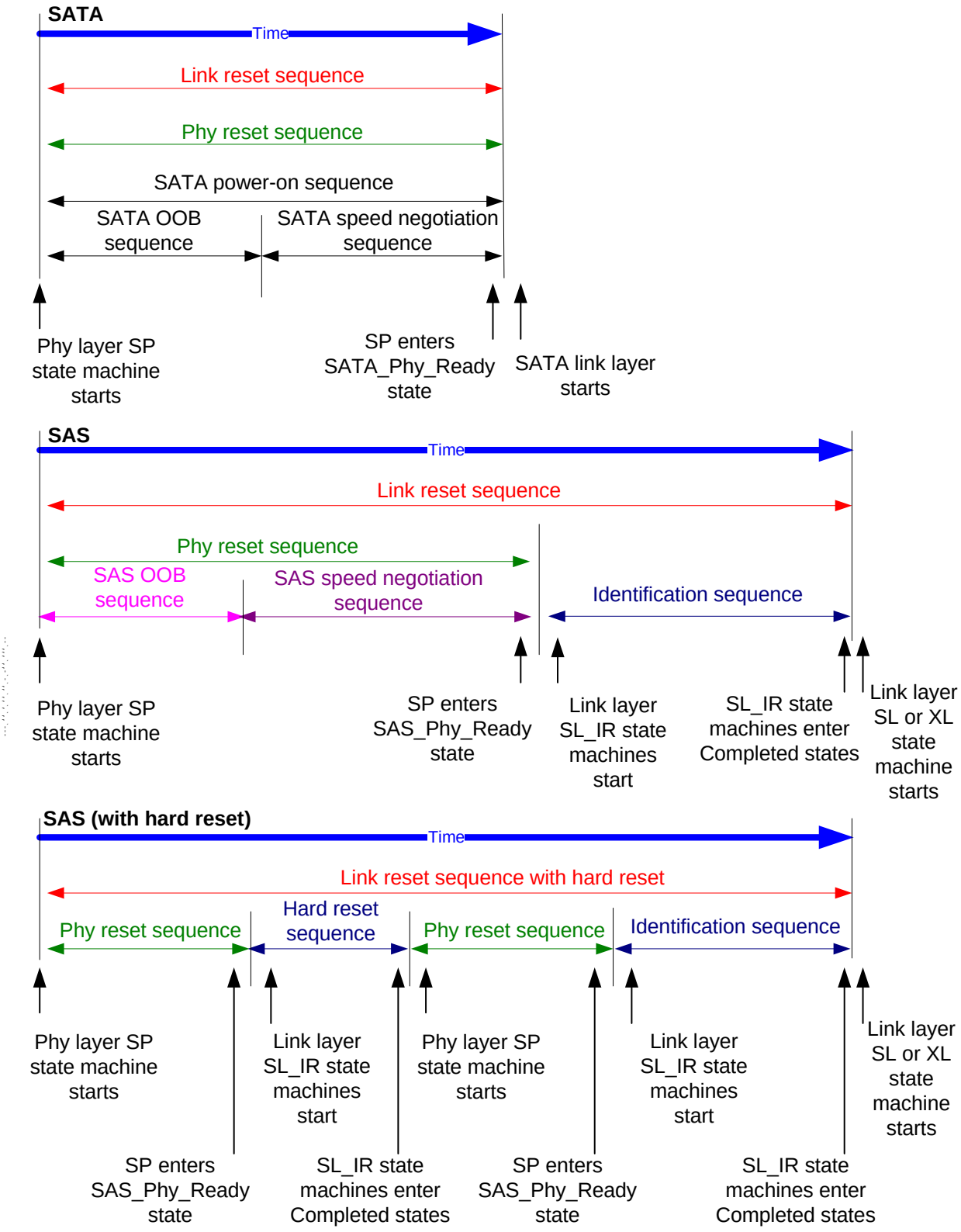


Figure 32 — Reset terminology

The phy reset sequences, including the OOB sequence and speed negotiation sequences, are implemented by the SP state machine and are described in 6.6 and 6.8. The hard reset sequence and identification sequence are implemented by the SL_IR state machine and are described in 7.9.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The link reset sequence has no effect on the transport layer and application layer. HARD_RESET may be used during the identification sequence to initiate a hard reset. The link reset sequence serves as a hard reset for SATA devices.

4.4.2 Hard reset

After the phy reset sequence, if a SAS phy receives a HARD_RESET, it shall be considered a reset event and initiate a hard reset of the port containing that phy.

When a port detects a hard reset, it shall stop transmitting valid dwords on each of the phys contained in that port. Each phy may then participate in new phy reset sequences and start transmitting. The hard reset shall not affect any other ports in the device (see SAM-3). Expander devices shall not forward HARD_RESETs to other phys.

If the port is an SSP port, a hard reset causes a Transport Reset event notification to the SCSI application layer (see 10.2.4); the SCSI device shall perform the actions defined for hard reset in SAM-3.

If the port is an STP port, the ATA device shall perform the actions defined for power-on or hardware reset in ATA.

If the port is an expander port, 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.

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 10.4.3.10).

After processing a hard reset, a phy shall originate a link reset sequence.

After the link reset sequence completes, each logical unit to which an SSP target port has access shall create a unit attention condition for all SSP initiator ports. The sense key shall be set to UNIT ATTENTION with the additional sense code set to SCSI BUS RESET OCCURRED (see SPC-3 for additional sense code assignments).

4.5 I_T nexus loss

When a SAS port receives OPEN_REJECT (NO DESTINATION), OPEN_REJECT (PATHWAY BLOCKED), or an open connection timeout occurs in response to a connection request, it shall retry the connection request until:

a) the connection is established;
b) for SSP target ports, the time indicated by the I_T NEXUS LOSS field in the Protocol-Specific Port Control mode page (see 10.2.6.2) expires; or
c) for STP or SMP connection requests, a vendor-specific I_T nexus loss time expires.

An SSP initiator port should retry the connection request for the time indicated by the I_T NEXUS LOSS field in the Protocol-Specific Port Control mode page (see 10.2.6.2) for the SSP target port to which it is trying to establish a connection.

If the time expires, then the port shall send a Nexus Lost event notification to the SCSI application layer (see 10.2.4); the SCSI device shall perform the actions defined for I_T nexus loss in SAM-3.

I_T nexus loss is handled by the port layer state machine (see 8.2.2.3).

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) expander connection manager (ECM);
   B) expander connection router (ECR); and
   C) broadcast primitive processor (BPP);

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) two or more physical expander phys. For the maximum number of phys, see 4.1.5;
c) an expander port available per phy; and
d) an SMP target port.

An expander device may contain SAS devices with SSP ports, STP ports, and/or SMP ports.

Figure 33 shows a model of an expander device showing the state machines in each expander port. The internal SMP port is not shown.

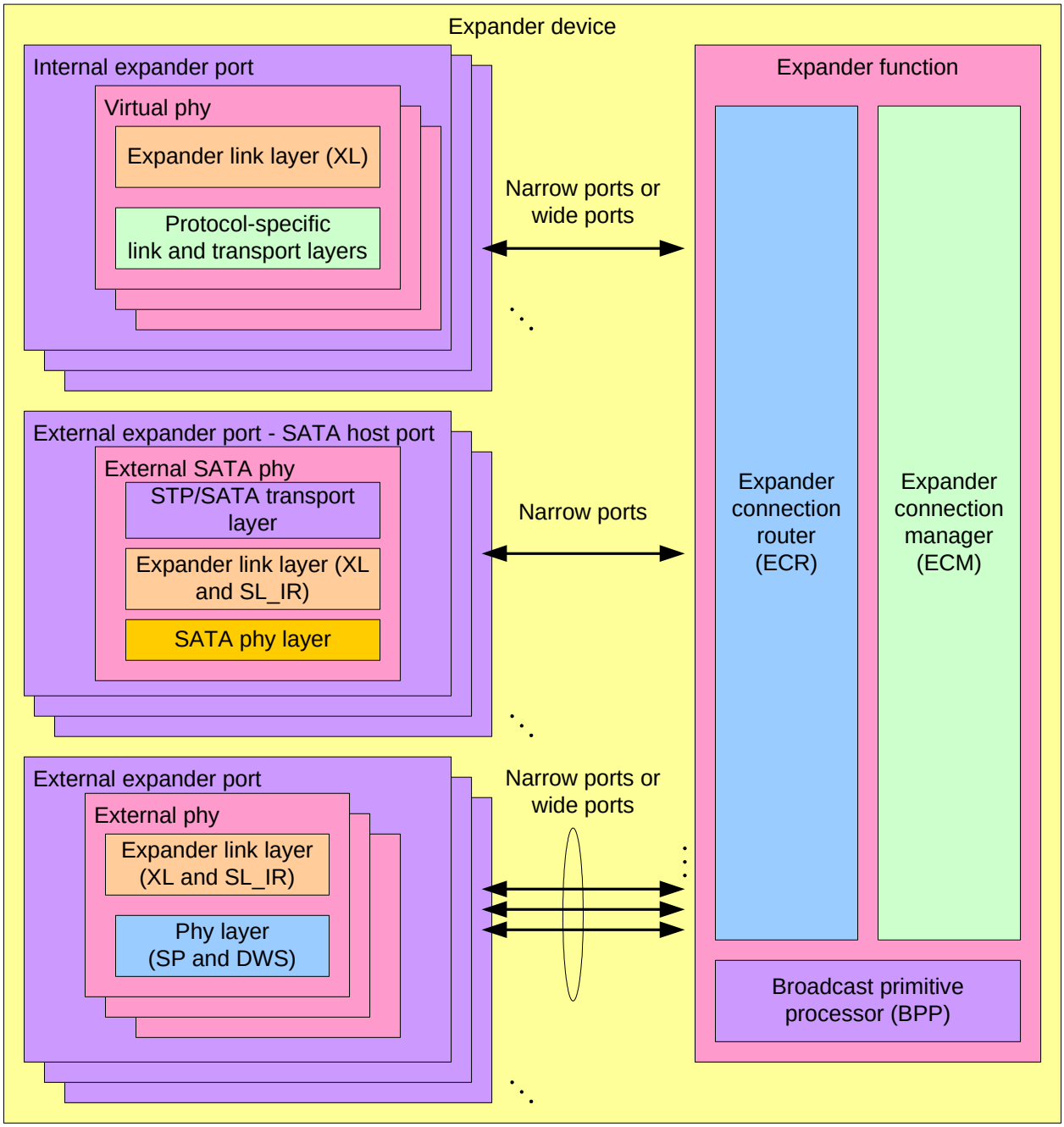


Figure 33 — Expander device model

4.6.2 Expander ports

An external expander port contains one or more physical phys (see 4.1.2). Each expander phy contains an expander link layer with an XL state machine and an SL_IR state machine. The expander link layers within an expander port request and respond to connection requests independently.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

An internal expander port contains a virtual phy with an expander link layer and a protocol-specific transport layer (e.g., to provide access to a SCSI enclosure services device as an SSP target).

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.

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 routed addressing methods;
b) arbitrates and assigns or denies path resources for connection requests following SAS arbitration and pathway recovery rules; and
c) configures the ECR.

4.6.4 Expander connection router (ECR)

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

4.6.5 Broadcast primitive processor (BPP)

The BPP receives broadcast primitive requests from each expander phy and requests transmission of those requests on all expander ports except the expander port from which the broadcast primitive request was received.

4.6.6 Expander device interfaces

4.6.6.1 Expander device interface overview

The expander device arbitrates and routes between expander phys. All routing occurs between expander 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.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 34 describes the interfaces present within an expander device.

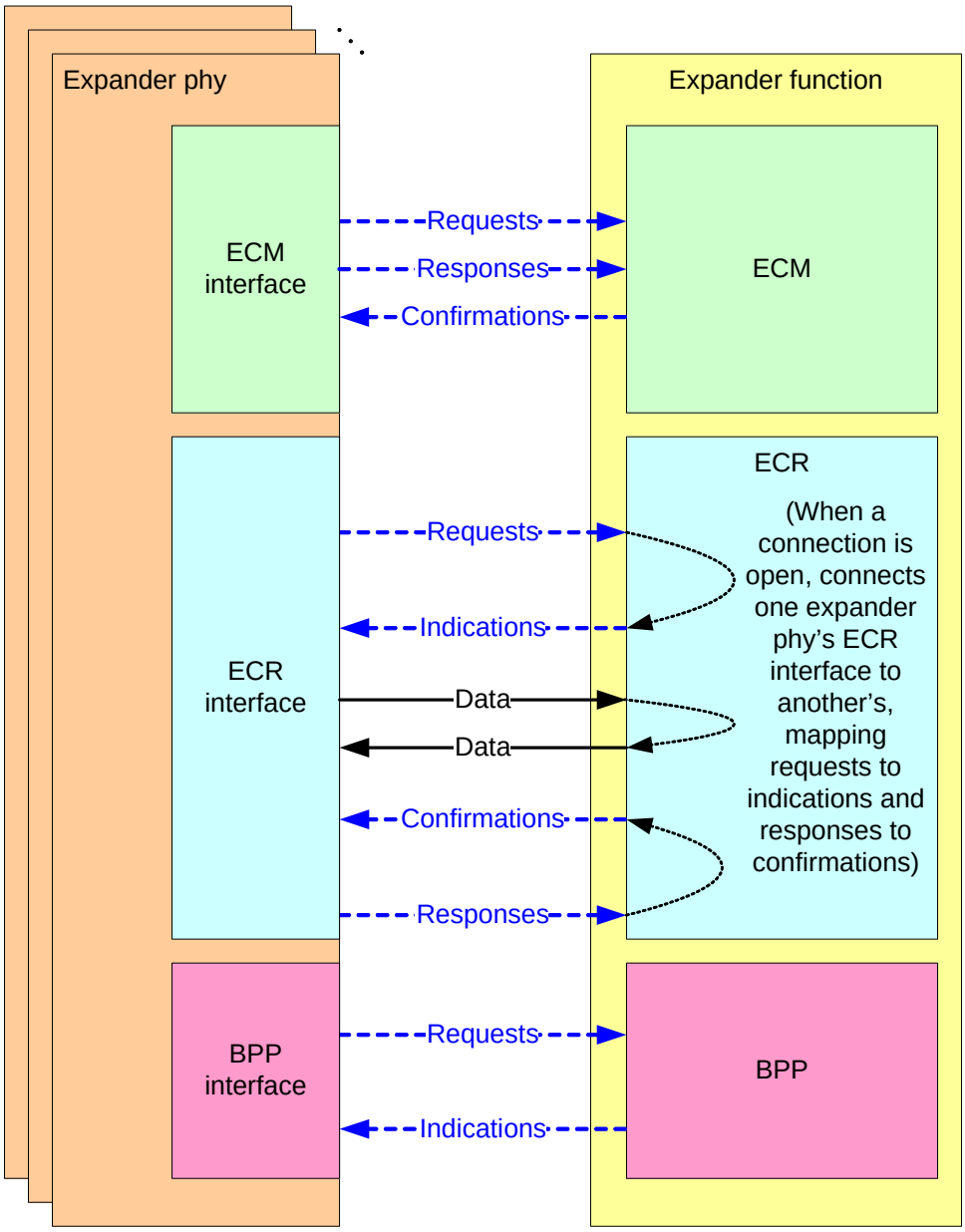


Figure 34 — Expander device interfaces

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.6.6.2 Expander device interfaces detail

Figure 35 shows the interface requests, confirmations, indications, and responses used by an expander device to manage connections.

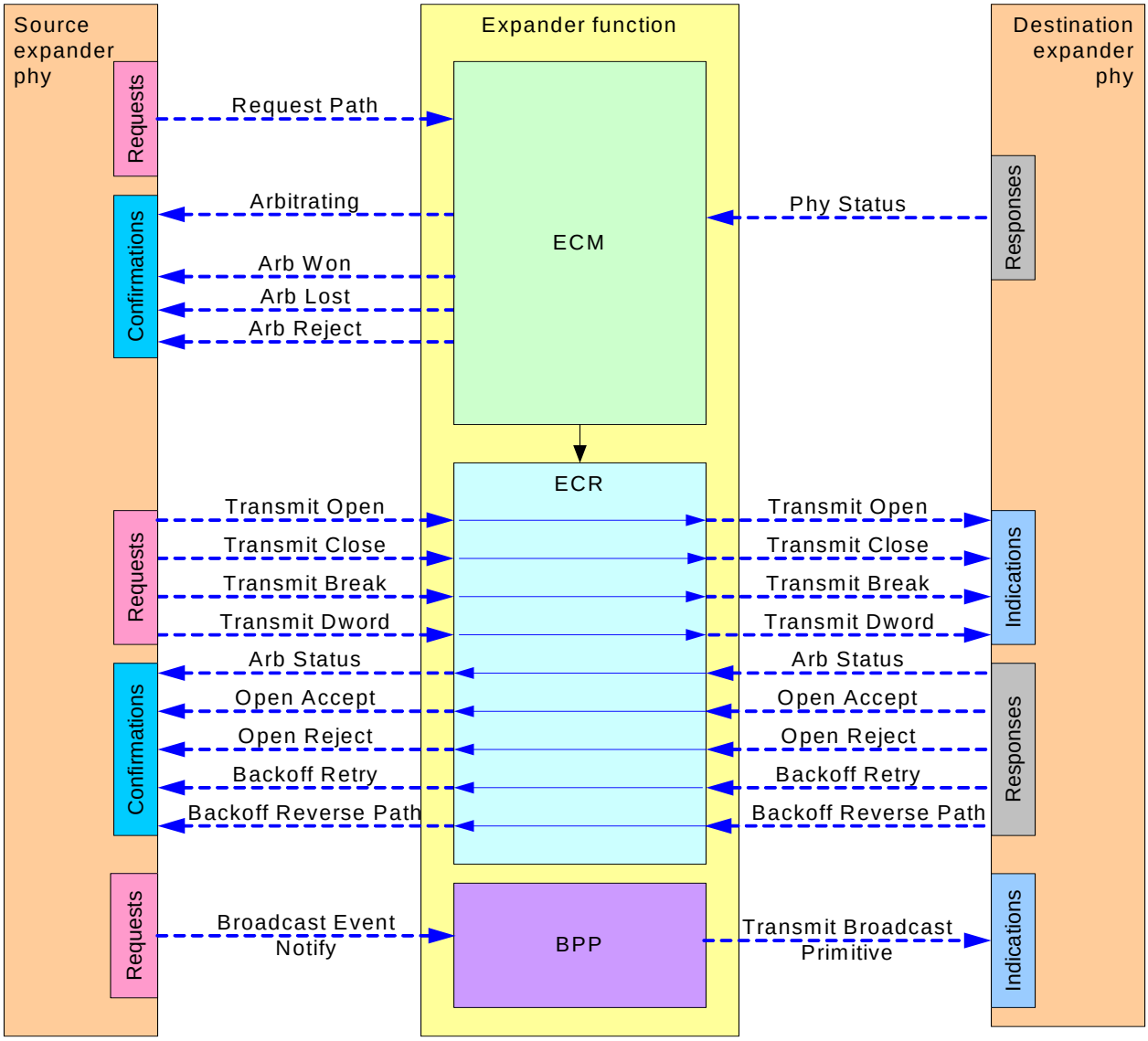


Figure 35 — Expander device interface detail

4.6.6.3 ECM interface

Table 8 describes the requests from an expander phy to the ECM.

Table 8 — Expander phy to ECM requests

Message	Description
Request Path (arguments)	Request for a connection.

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 8 describes the responses from an expander phy to the ECM.

Table 9 — Expander phy to ECM responses

Message	Description
Phy Status (Partial Pathway)	Response meaning that an expander phy contains a partial pathway.
Phy Status (Blocked Partial Pathway)	Response meaning that an expander phy contains a partial pathway and the most recent AIP received or indicated is AIP (WAITING ON PARTIAL).
Phy Status (Connected)	Response meaning that an expander phy is being used for a connection.

Table 10 describes the confirmations from the ECM to an expander phy.

Table 10 — ECM to expander phy confirmations

Message	Description
Arbitrating (Waiting On Partial)	Confirmation that the ECM has determined that: a) at least one destination phy matches the requested destination SAS address; and b) all of the phys within the destination port are returning a Phy Status (Partial Pathway) response.
Arbitrating (Blocked On Partial)	Confirmation that the ECM has determined that: a) at least one destination phy matches the requested destination SAS address; and b) all of the phys within the destination port are returning a Phy Status (Blocked Partial Pathway) response.
Arbitrating (Waiting On Connection)	Confirmation that the ECM has determined that: a) at least one destination phy matches the requested destination SAS address; b) no phys within the destination port are available; and c) at least one of the phys within the destination port are returning a Phy Status (Connected) response.
Arb Won	Confirmation that an expander phy has won path arbitration.
Arb Lost	Confirmation that an expander phy has lost path arbitration.
Arb Reject (No Destination)	Confirmation that the ECM did not find an operational expander phy configured to match the requested destination SAS address.
Arb Reject (Bad Destination)	Confirmation that the ECM has determined that the requested destination SAS address maps back to the requesting port.
Arb Reject (Bad Connection Rate)	Confirmation that the ECM has determined that at least one destination phy matches the requested destination SAS address but no phys within the destination port are configured to support the requested connection rate.
Arb Reject (Pathway Blocked)	Confirmation that the ECM has determined that the requesting expander phy shall back off according to SAS pathway recovery rules.

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4.6.6.4 ECR interface

Table 11 describes the requests from an expander phy to the ECR and the corresponding indications from the ECR to another expander phy.

Table 11 — Expander phy to ECR to expander phy requests and indications

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

Table 12 describes the responses from an expander phy to the ECR and the corresponding confirmations from the ECR to another expander phy.

Table 12 — Expander phy to ECR to expander phy responses and confirmations

Message	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 AIP (WAITING ON DEVICE) has been received.
Open Accept	Confirmation/response that OPEN_ACCEPT has been received.
Open Reject	Confirmation/response that OPEN_REJECT has been received.
Backoff Retry	Confirmation/response that: a) a higher priority OPEN address frame has been received (see 7.12.3); 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 7.12.3); 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4.6.6.5 BPP interface

Table 13 describes the requests from an expander phy to the BPP.

Table 13 — Expander phy to BPP requests

Message	Description
Broadcast Event Notify (Phy Not Ready)	Request to transmit a BROADCAST (CHANGE) on all other ports because an expander phy lost dword synchronization (see 6.7).
Broadcast Event Notify (SATA Spinup Hold)	Request to transmit a BROADCAST (CHANGE) on all other ports because the SATA spinup hold state has been reached (see 6.9).
Broadcast Event Notify (Identification Sequence Complete)	Request to transmit a BROADCAST (CHANGE) on all other ports because an expander phy has completed the identification sequence (see 7.9).
Broadcast Event Notify (CHANGE Received)	Request to transmit a BROADCAST (CHANGE) on all other ports because a BROADCAST (CHANGE) was received.
Broadcast Event Notify (RESERVED CHANGE Received)	Request to transmit a BROADCAST (RESERVED CHANGE) on all other ports because a BROADCAST (RESERVED CHANGE) was received.
Broadcast Event Notify (RESERVED 0 Received)	Request to transmit a BROADCAST (RESERVED 0) on all other ports because a BROADCAST (RESERVED 0) was received.
Broadcast Event Notify (RESERVED 1 Received)	Request to transmit a BROADCAST (RESERVED 1) on all other ports because a BROADCAST (RESERVED 1) was received.

Table 14 describes the indications from the BPP to an expander phy.

Table 14 — BPP to expander phy indications

Message	Description
Transmit Broadcast (type)	Indication to transmit a BROADCAST 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;
b) table routing attribute; or
c) subtractive routing attribute.

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

a) the table routing method routes connection requests to attached expander devices using an expander route table;
b) the subtractive routing method routes unresolved connection requests to an attached expander device; or
c) the direct routing method routes connection requests to attached end devices or SAS devices contained in the expander device.

An expander phy with the direct routing attribute allows the ECM to use the direct routing method.

An expander phy with the table routing attribute allows the ECM to use one of the following methods to route connection requests:

a) the table routing method and the direct routing method if attached to an expander device; or

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) the direct routing method if attached to an end device.

An expander device may have zero or more phys with the table routing attribute.

An expander phy with the subtractive routing attribute allows the ECM to use one of the following methods to route connection requests:

a) the subtractive routing method if attached to an expander device; or
b) the direct routing method if attached to an end device.

An edge expander device shall have at most one defined port containing phys with the subtractive routing attribute. Phys in a fanout expander device shall not have the subtractive routing attribute.

An edge expander device shall only use phys with the table routing attribute to attach to phys with the subtractive routing attribute in other edge expander devices within an edge expander device set.

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

If multiple phys within an expander device have subtractive routing attributes and are attached to expander devices that do not have identical SAS addresses, the application client that is performing the discover process (see 4.6.7.4) shall report an error in a vendor-specific manner.

4.6.7.2 Connection request routing

The ECM shall determine how to route a connection request from a source expander phy to a destination expander phy in a different expander port using the following precedence:

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

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

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

4.6.7.3 Expander route table

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 and expander phy identifiers. Each association represents an expander route entry.

An expander device reports the size of its expander route table and indicates if the expander route table is configurable in the SMP REPORT GENERAL function (see 10.4.3.3). Each expander route entry shall be disabled after power on.

A management application client may reference a specific expander route entry within an expander route table with the SMP REPORT ROUTE INFORMATION function (see 10.4.3.8) and the SMP CONFIGURE ROUTE INFORMATION function (see 10.4.3.9).

--`,,,`,,,-`-`,,,`,,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 36 shows a representation of an expander route table.

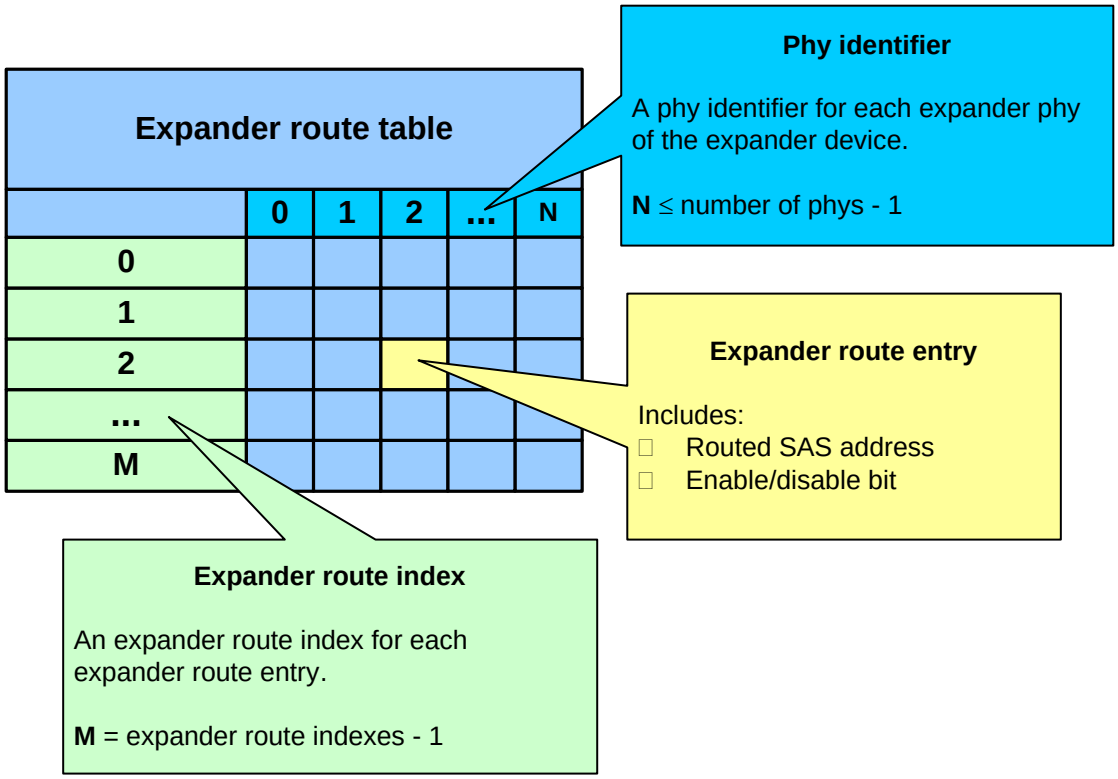


Figure 36 — Expander route table example

The number of end devices that may be attached to an edge expander device set is dependent on the number of expander route entries in the expander route table of the edge expander devices.

4.6.7.4 Discover process

The management application client performing the discover process shall perform a level-order (i.e., breadth-first) traversal of the SAS domain to identify the end devices and expander devices in the SAS domain. The order of traversal shall be to discover:

1) the device to which the device containing the management application client is attached;
2) if the attached device is an expander device, every device attached to that expander device; and
3) for each expander device found, every device attached to that expander device.

This order is repeated until all expander devices have been traversed.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 37 shows an example of level-order traversal.

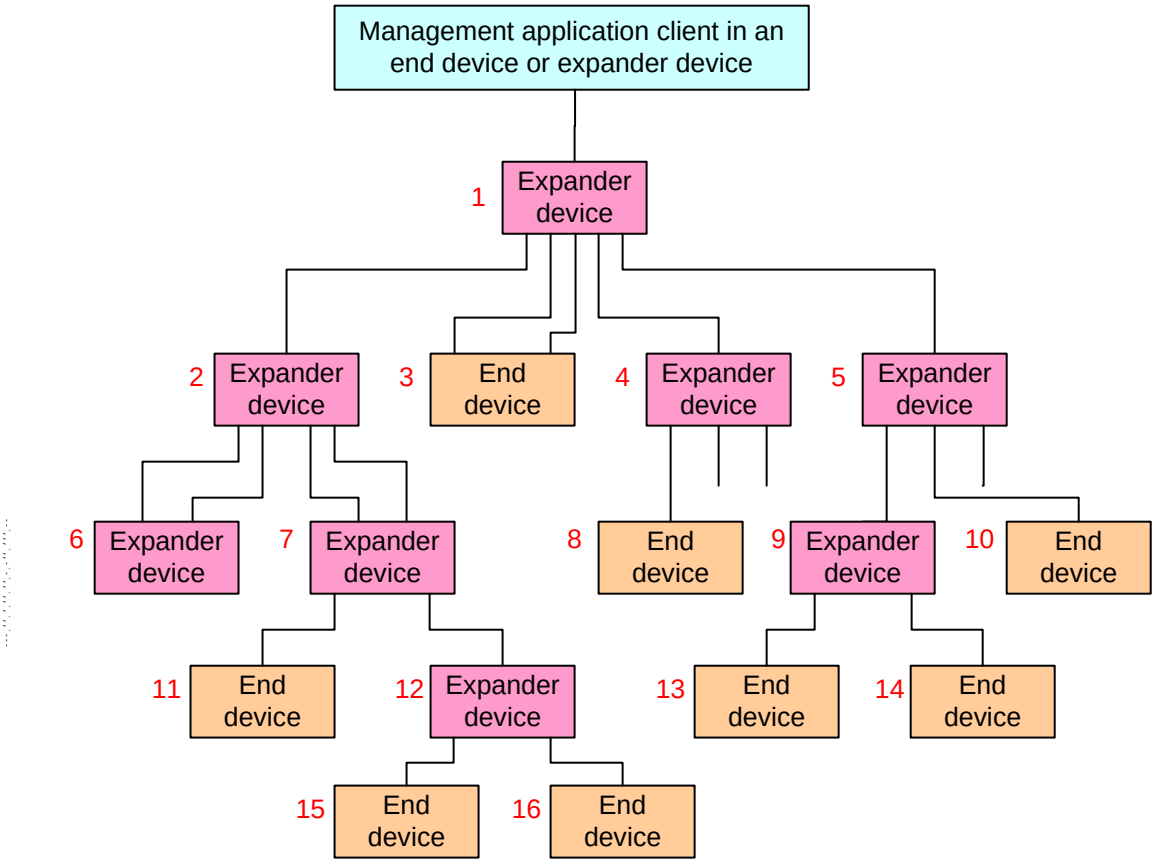


Figure 37 — Level-order traversal example

The discover process begins with the management application client determining that an expander device is attached.

If an expander device is detected, the management application client shall use the SMP REPORT GENERAL function (see 10.4.3.3) and SMP DISCOVER function (see 10.4.3.5) to determine what is attached to each expander phy.

If an expander device is detected and its CONFIGURABLE ROUTE TABLE bit is set to one in the SMP REPORT GENERAL function response, the management application client shall configure its expander route table as described in 4.6.7.5.

If an end device is detected, the management application client may attempt to open an SMP connection to determine if it contains an SMP target port, and use the SMP REPORT GENERAL function to determine additional information about the end device. End devices are not required to support SMP.

The result of the discover process is that the management application client has the necessary information to communicate with each SAS device and expander device in the SAS domain and each configurable expander device is configured with the expander route entries to allow routing of connection requests through the SAS domain.

The discover process may be aborted prior to completion if there is an indication that it may be based on incorrect information (e.g., arrival of a BROADCAST (CHANGE)).

The management application client shall allow the following attachments between expander phys:

a) edge expander phy with the subtractive routing attribute attached to an edge expander phy with the subtractive routing attribute;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

b) edge expander phy with the subtractive routing attribute attached to an edge expander phy with the table routing attribute;
c) edge expander phy with the subtractive routing attribute attached to a fanout expander phy with the table routing attribute.

If the management application client detects any other combination of expander phy attachment (e.g., expander phy with table routing attached to expander phy with table routing, or an expander phy with direct routing attached to an expander phy), it shall report an error in a vendor-specific manner.

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

If the management application client detects an expander route entry that references the SAS address of the expander device itself (i.e., self-reference), it shall disable the expander route entry by setting the DISABLE EXPANDER ROUTE ENTRY bit to one in the SMP CONFIGURE ROUTE INFORMATION function (see 10.4.3.9).

The management application client shall disable each expander route entry in the route table by setting the DISABLE EXPANDER ROUTE ENTRY bit to one in the SMP CONFIGURE ROUTE INFORMATION function (see 10.4.3.9) 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 with a SAS address it has already found attached to another expander device, it has found a routing loop and may break the loop by disabling all the expander phys attached to that SAS address except for the expander phy with the lowest phy identifier in the expander device with the lowest SAS address by using the SMP CONFIGURE ROUTE INFORMATION function (see 10.4.3.9).

Annex I contains an example algorithm used by an application client to perform the discover process.

4.6.7.5 Expander route index order

The expander route table shall be configured for each expander phy (in either a fanout expander device or an edge expander device) that:

a) has a table routing attribute; and
b) is attached to an edge expander device.

For purposes of configuring the expander route table, the edge 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 edge expander device is considered level 1;
3) devices attached to the level 1 edge expander device, except for the level 0 expander device, are considered level 2;
4) devices attached to level 2 edge expander devices, except for level 1 edge expander devices, are considered level 3; and
5) for each n greater than 3, devices attached to level n-1 edge expander devices, except for level n-2 edge 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 (i.e., all level 1 entries first, then all level 2 entries, then all level 3 entries, etc.) up to the value of the EXPANDER ROUTE INDEXES field reported by the SMP REPORT GENERAL function (see 10.4.3.3).

Assuming the level 1 edge expander device has N expander phys, the first N entries shall be used for the level 1 edge expander device's expander phy 0 through expander phy N. If an expander phy is attached to a device other than the level 0 expander device (i.e., attached to a level 2 device), the corresponding expander route entry shall contain the SAS address of the level 2 device. Otherwise (e.g., if the expander phy is not attached to any device, or the expander phy is attached to the level 0 expander device), the corresponding expander route entry shall be disabled.

For each of the level 2 devices that:

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

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

the next M entries shall be used for the level 2 edge expander device's expander phy 0 through expander phy M. If an expander phy is attached to a device other than the level 1 expander device (i.e., attached to a level 2 device), the corresponding expander route entry shall contain the SAS address of the level 3 device. Otherwise (e.g., if the expander phy is not attached to any device, or the expander phy is attached to the level 1 expander device), the corresponding expander route entry shall be disabled.

This process shall repeat for all levels of edge expander devices in the edge expander device set.

Figure 38 shows a portion of an edge expander device set, where phy A in the root edge expander device is being configured.

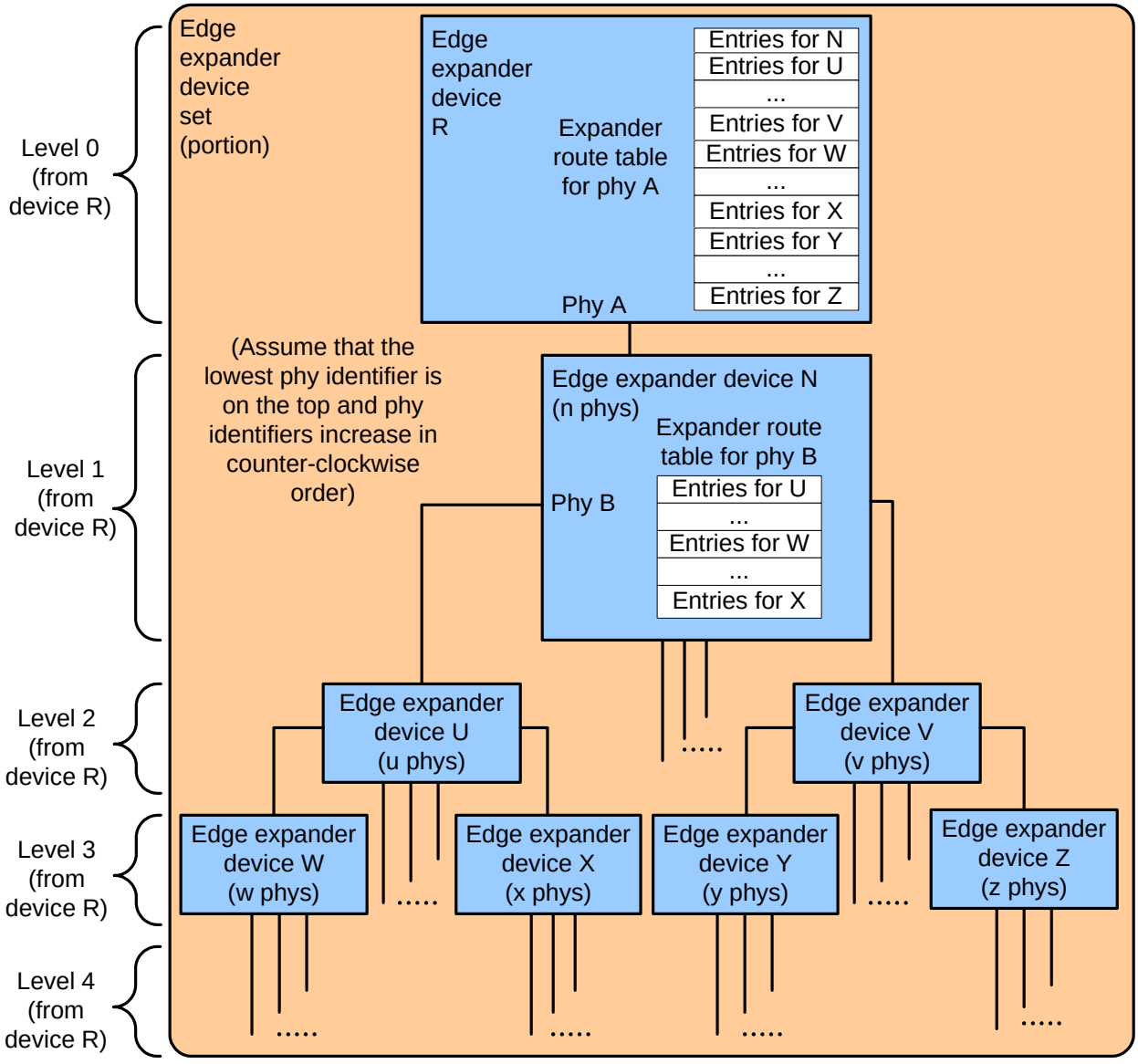


Figure 38 — Expander route index levels example

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 39 shows a fanout expander device and an edge expander device set, where phy A in the fanout expander device is being configured.

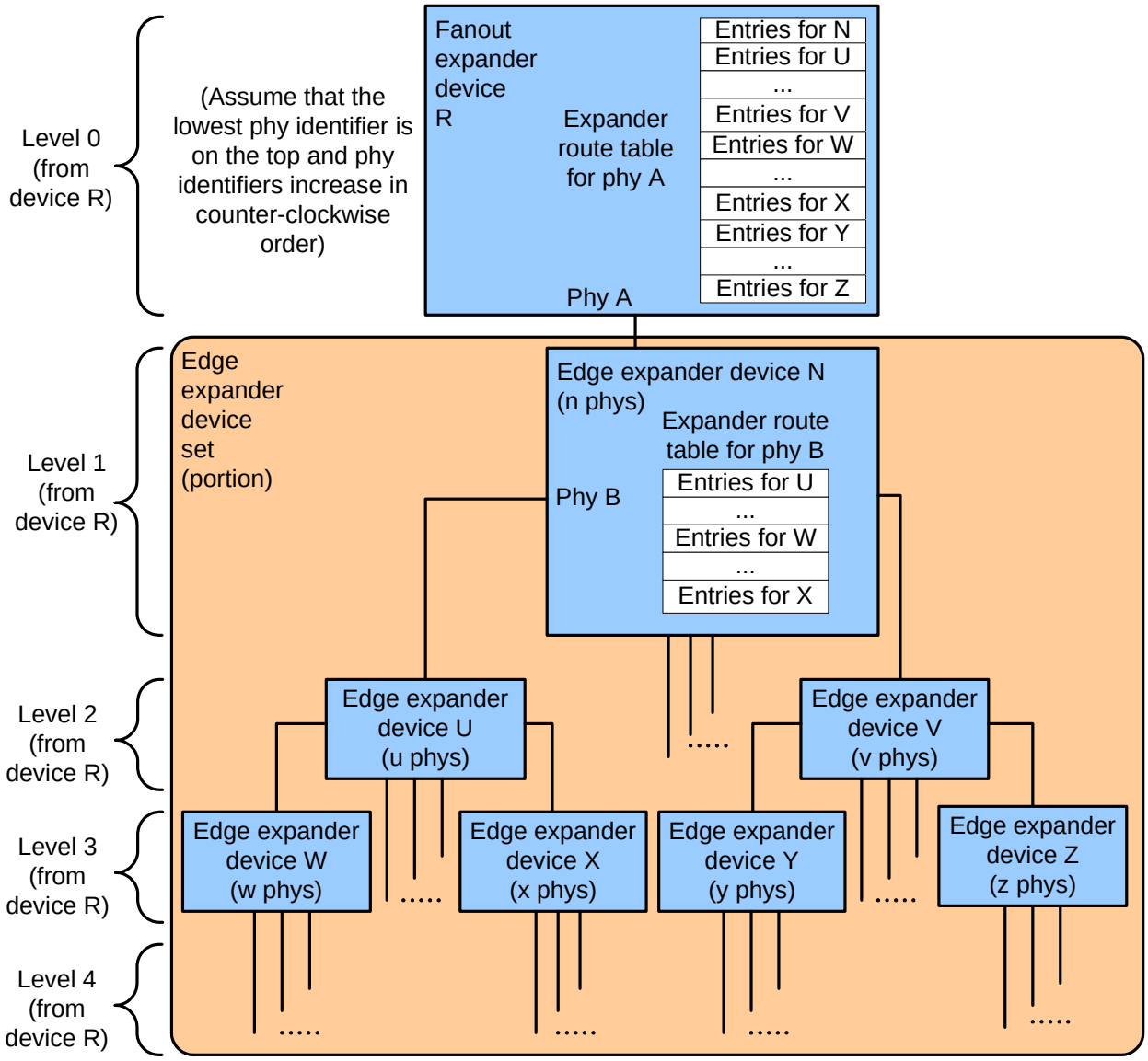


Figure 39 — Expander route index levels example with fanout expander device

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 15 shows how the expander route table is configured for edge expander device R phy A in figure 38 and the fanout expander device R phy A in figure 39.

Table 15 — Expander route table levels for edge expander device R or fanout expander device R

Expander route index	Expander route entry contents
Level 1 (from device R) entries	
0	SAS address of the port attached to edge expander device N phy 0
1	SAS address of the port attached to edge expander device N phy 1
...	...additional phys in edge expander device N...
n	SAS address of the port attached to edge expander device N phy n
Level 2 (from device R) entries	
n + 1	SAS address of the port attached to edge expander device U phy 0
...	...additional phys in edge expander device U...
n + 1 + u	SAS address of the port attached to edge expander device U phy u
...	...additional devices at level 2 (from device R)...
	SAS address of the port attached to edge expander device V phy 0
...	...additional phys in edge expander device V...
	SAS address of the port attached to edge expander device V phy v
Level 3 (from device R) entries	
	SAS address of the port attached to edge expander device W phy 0
...	...additional phys in edge expander device W...
	SAS address of the port attached to edge expander device W phy w
...	...additional devices at level 3 (from device R)...
	SAS address of the port attached to edge expander device Z phy 0
...	...additional phys in edge expander device Z...
	SAS address of the port attached to edge expander device Z phy z
Entries for additional levels	
...	...

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 16 shows how the expander route table is configured for edge expander device N phy B in figure 38 and figure 39.

Table 16 — Expander route table levels for edge expander device N

Expander route index	Expander route entry contents
Level 1 (from device N) entries	
0	SAS address of the port attached to edge expander device U phy 0
...	...additional phys in edge expander device U...
u	SAS address of the port attached to edge expander device U phy u
Level 2 (from device N) entries	
u + 1	SAS address of the port attached to edge expander device W phy 0
...	...additional phys in edge expander device W...
u + 1 + w	SAS address of the port attached to edge expander device W phy w
...	...additional devices at level 2 (from device N)...
	SAS address of the port attached to edge expander device X phy 0
...	...additional phys in edge expander device Z...
	SAS address of the port attached to edge expander device X phy z
Entries for additional levels	
...	...

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 40 shows an example topology with a fanout expander device.

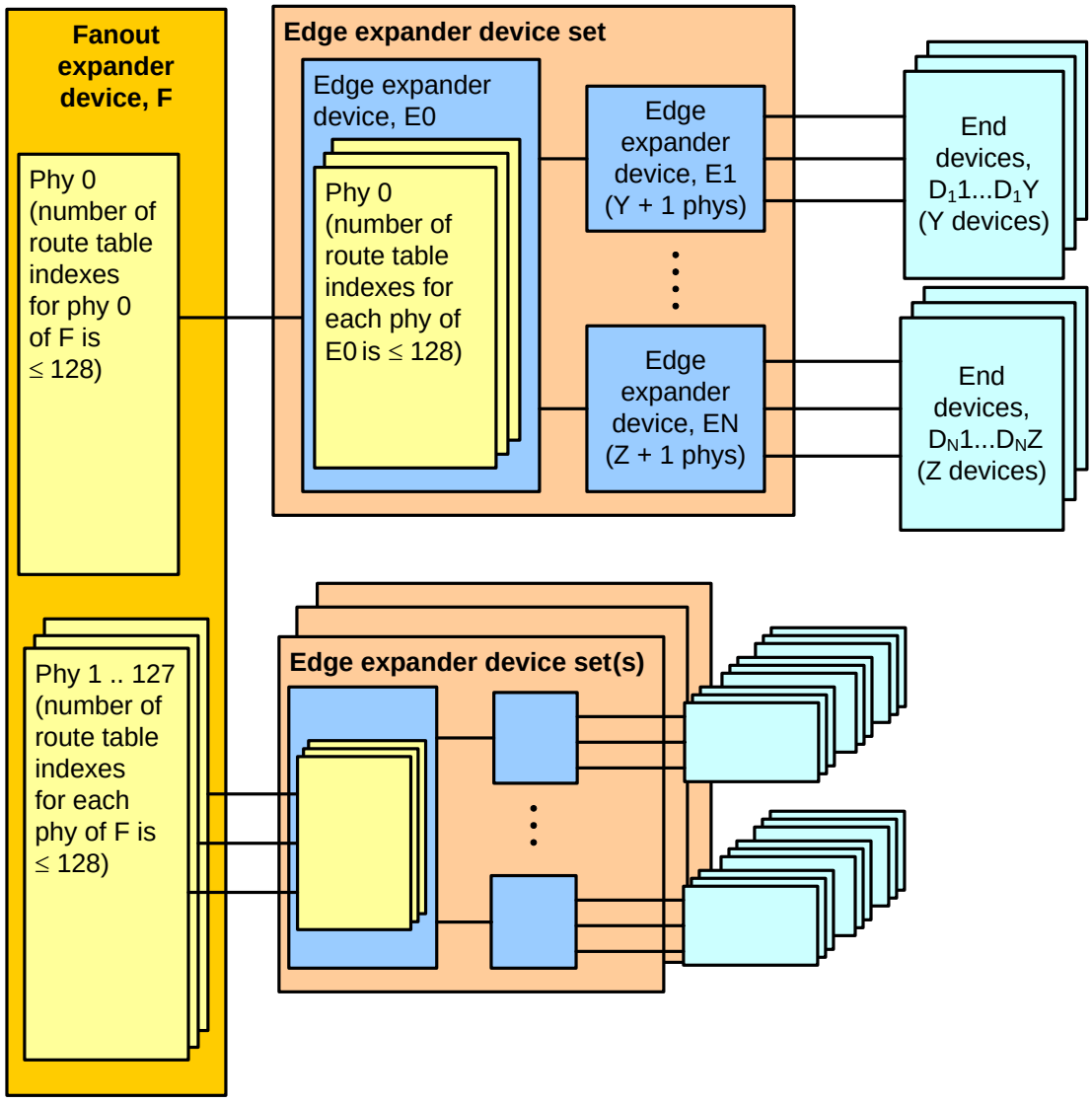


Figure 40 — Expander route index order example

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 17 shows the expander route index order for the edge expander E0 phy 0 in figure 40.

Table 17 — Expander route entries for edge expander E0 phy 0

Expander route index	Expander route entry contents
Level 1 entries	
0	SAS address (e.g., E0) of the device attached to phy 0 of edge expander device E1. This expander route entry is disabled because E0 is the expander device being configured.
1	SAS address (e.g., $D_1$1) of the device attached to phy 1 of edge expander device E1
2	SAS address (e.g., $D_1$2) of the device attached to phy 2 of edge expander device E1
...	...
Y	SAS address (e.g., D_1Y) of the device attached to phy N of the attached edge expander device E1
Level 2: no entries since all devices attached to E1 through EN, except for E0, are end devices	

Table 18 shows the expander route index order for the fanout expander F phy 0 in figure 40.

Table 18 — Expander route entries for fanout expander device F phy 0

Expander route index	Expander route entry contents
Level 1 entries	
0	SAS address (e.g., F) of the device attached to phy 0 of edge expander device E0. This expander route entry is disabled because F is the expander device being configured.
1	SAS address (e.g., E1) of the device attached to phy 1 of edge expander device E0
2	SAS address (e.g., E2) of the device attached to phy 2 of edge expander device E0
...	...
N	SAS address (e.g., EN) of the device attached to phy N of the attached edge expander device E0
Level 2 entries	
N + 1	SAS address (e.g., E0) of the device attached to phy 0 of the edge expander device E1. This expander route entry is disabled because E0 is attached to the expander device being configured.
...	...
N + 1 + Y	SAS address (e.g., D_1Y) of the device attached to phy Y of the edge expander device E1
...	...
	SAS address (e.g., E0) of the device attached to phy 0 of the edge expander device EN. This expander route entry is disabled because E0 is attached to the expander device being configured.
...	...
	SAS address (e.g., D_NZ) of the device attached to phy Z of the edge expander device EN
Level 3 entries: none since all devices attached to E1 through EN, except for E0, are end devices	

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5 Physical layer

5.1 Physical layer overview

The physical layer defines cables and connectors and transmitter and receiver electrical characteristics.

5.2 Passive interconnect

5.2.1 SATA cables and connectors

Figure 41 shows a schematic representation of the cables and connectors defined by SATA (see ATA/ATAPI-7 V3). A SATA host is analogous to a SAS initiator device; a SATA device is analogous to a SAS target device.

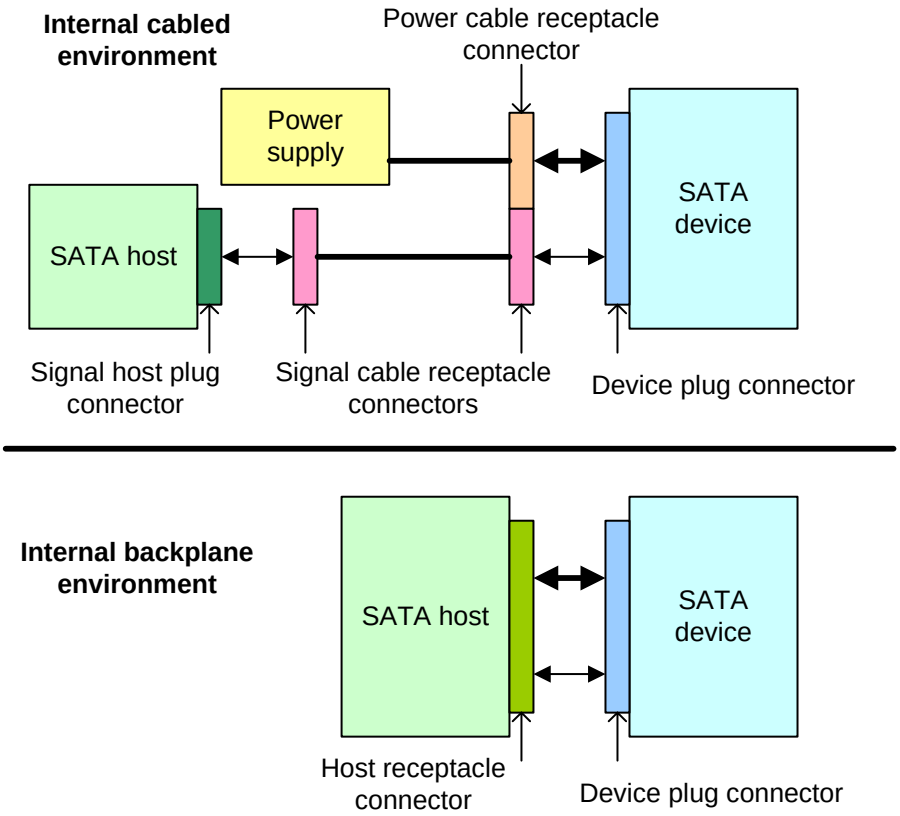


Figure 41 — SATA cables and connectors

5.2.2 SAS cables and connectors

This standard supports external cable, internal cable, and internal backplane environments.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 42 shows a schematic representation of the cables and connectors defined in this standard to support an external environment.

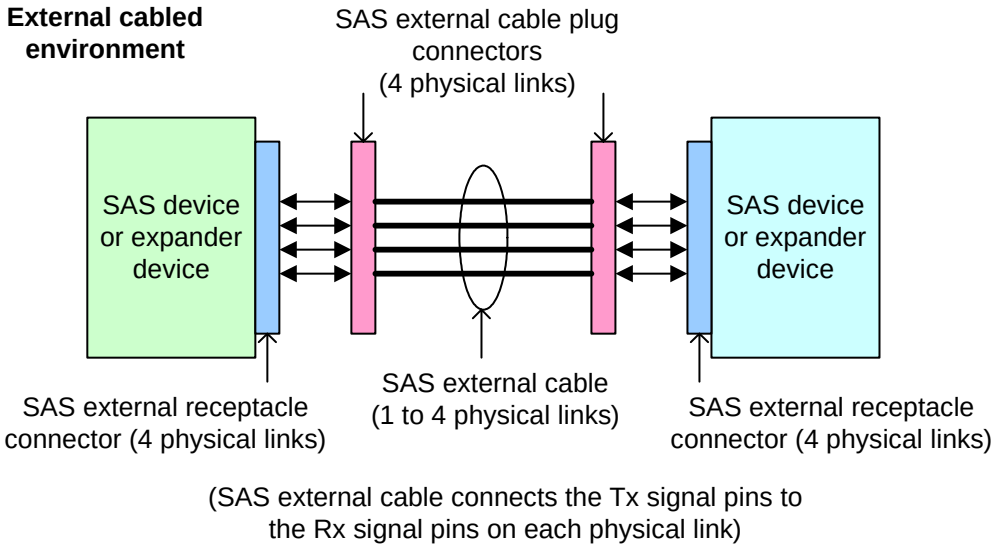


Figure 42 — SAS cables and connectors - external environment

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 43 shows a schematic representation of the cables and connectors defined in this standard for internal environments.

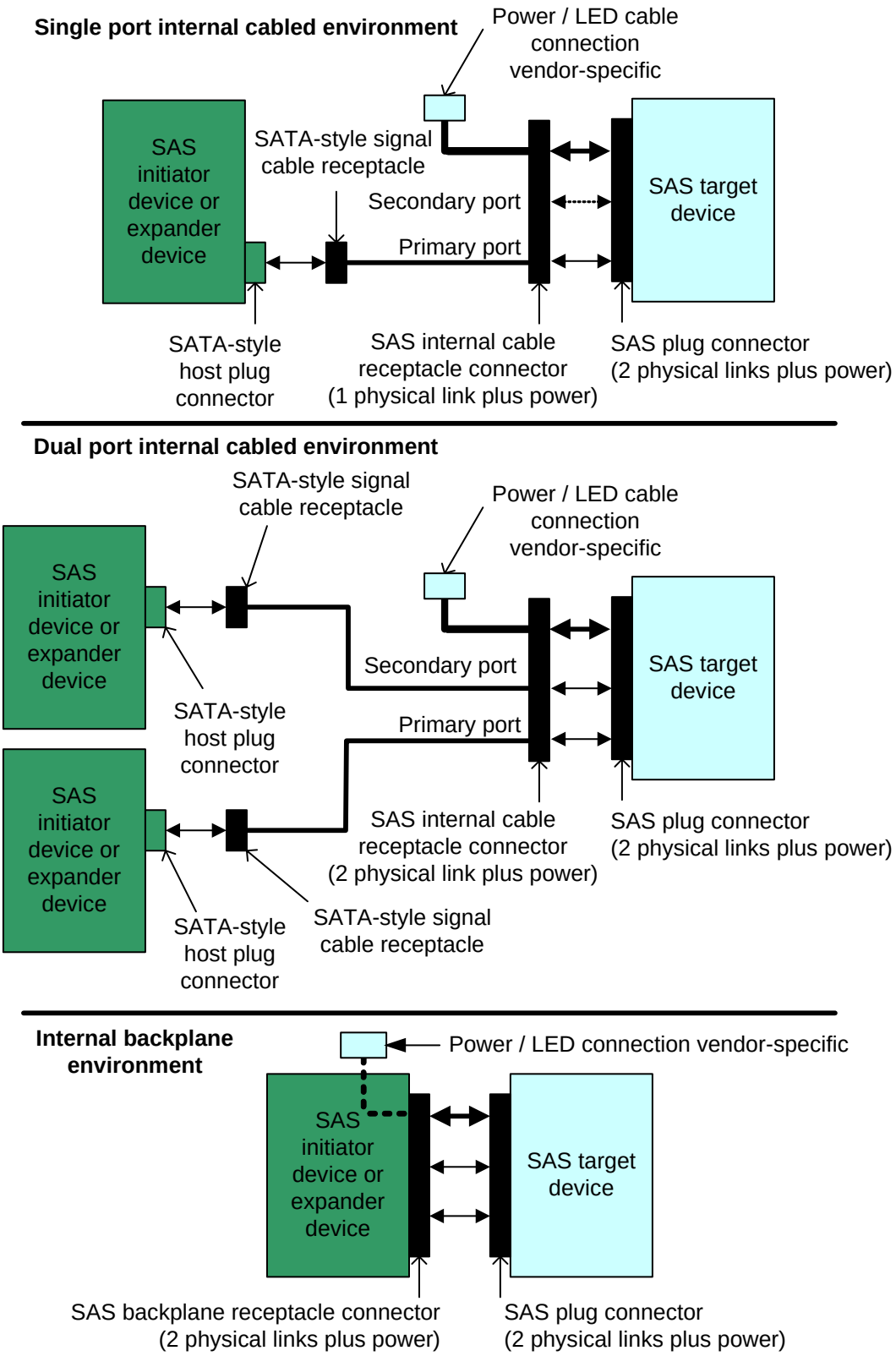


Figure 43 — SAS cables and connectors - internal environment

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 19 summarizes the connectors defined in this standard.

Table 19 — Connectors

Type of connector	Physical links	Reference	Attaches to	Physical links	Reference
SAS plug	2	5.2.3.2	SAS internal cable receptacle	1 or 2	5.2.3.3
			SAS backplane receptacle	2	5.2.3.4
SAS internal cable SATA-style signal cable receptacle	1	ATA/ATAPI -7 V3	SATA-style host plug	1	ATA/ATAPI -7 V3
SAS internal cable receptacle	1 or 2	5.2.3.3	SAS plug	2	5.2.3.2
			SATA device plug	1	SATA
SAS backplane receptacle	2	5.2.3.4	SAS plug	2	5.2.3.2
			SATA device plug	1	ATA/ATAPI -7 V3
SAS external cable plug	4	5.2.3.6	SAS external receptacle	4	5.2.3.7
SAS external receptacle	4	5.2.3.7	SAS external cable plug	4	5.2.3.6

The SATA device plug connector (e.g., used by a SATA device) may be attached to a SAS backplane receptacle connector or a SAS internal cable receptacle connector, connecting the primary signal pairs and leaving the secondary signal pairs unconnected.

See SFF-8223, SFF-8323, and SFF-8523 for the connector locations on common form factors.

5.2.3 Connectors

5.2.3.1 Connectors overview

SAS connectors should be marked with the SAS icon (see Annex K).

5.2.3.2 SAS plug connector

SAS target devices supporting internal environments shall use the SAS plug connector. The SAS plug connector is defined in SFF-8482. It attaches to a SAS internal cable receptacle connector or a SAS backplane receptacle connector.

Table 20 defines the pin assignments.

5.2.3.3 SAS internal cable receptacle connector

SAS internal cables shall use a SAS internal cable receptacle connector on the SAS target device end. The SAS internal cable receptacle connectors are defined in SFF-8482. The single-port version attaches to either:

a) a SAS plug connector, providing contact for the power pins and only the primary physical link; or
b) a SATA device plug connector, providing contact for the power pins and the primary physical link.

The dual-port version attaches to:

a) a SAS plug connector, providing contact for the power pins and only the primary physical link;
b) a SAS plug connector, providing contact for the power pins and both the primary and secondary physical links; or
c) a SATA device plug connector, providing contact for the power pins and the primary physical link.

Table 20 defines the pin assignments. The secondary physical link (i.e., pins S8 through S14) is not supported by the single-port internal cable receptacle.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.2.3.4 SAS backplane receptacle connector

SAS backplanes shall use the SAS backplane receptacle connector. The SAS backplane receptacle connector (see SFF-8482) attaches to either:

a) a SAS plug connector, providing contact for the power pins and both primary and secondary physical links; or
b) a SATA device plug connector, providing contact for the power pins and the primary physical link.

Table 20 defines the pin assignments.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.2.3.5 SAS internal connector pin assignments

Table 20 defines the SAS target device signal assignments for pins in the SAS internal connector.

Table 20 — SAS target device connector pin assignments

Segment	Pin	Name
Primary Signal	S1	GROUND
Primary Signal	S2	RP+
Primary Signal	S3	RP-
Primary Signal	S4	GROUND
Primary Signal	S5	TP-
Primary Signal	S6	TP+
Primary Signal	S7	GROUND
Secondary Signal [b]	S8	GROUND
Secondary Signal [b]	S9	RS+
Secondary Signal [b]	S10	RS-
Secondary Signal [b]	S11	GROUND
Secondary Signal [b]	S12	TS-
Secondary Signal [b]	S13	TS+
Secondary Signal [b]	S14	GROUND
Power [a]	P1	V_{33}
Power [a]	P2	V_{33}
Power [a]	P3	V_{33}, precharge
Power [a]	P4	GROUND
Power [a]	P5	GROUND
Power [a]	P6	GROUND
Power [a]	P7	V_5, precharge
Power [a]	P8	V_5
Power [a]	P9	V_5
Power [a]	P10	GROUND
Power [a]	P11	READY LED
Power [a]	P12	GROUND
Power [a]	P13	V_{12}, precharge
Power [a]	P14	V_{12}
Power [a]	P15	V_{12}

[a] The precharge pin and each corresponding voltage pin shall be connected together on the SAS target device (e.g., the V_5, precharge pin is connected to the to V_5 pins).

[b] S8 through S14 are no-connects on single port implementations.

SAS target device signal assignments, except for the addition of the secondary physical link when present, are in the same locations as they are in a SATA device. On cable assemblies, backplanes, or any other

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

connection media, the Tx signal from one internal connector pair shall be connected to the corresponding Rx signal of the other internal connector pair (i.e., the TP+ signal pin of connector 1 shall connect to the RP+ signal pin of connector 2) if there is an internal connector at both ends of the transmission media.

The TP+, TP-, RP+, and RP- signals are used by the primary physical link. The TS+, TS-, RS+, and RS- signals are used by the secondary physical link.

5.2.3.6 SAS external cable plug connector

SAS external cables shall use the SAS external cable plug connector. The SAS external cable plug connector is defined in SFF-8470 as the 4x configuration with thumbscrews. No special SAS keying is provided. The SAS external cable plug connector attaches to a SAS external receptacle connector, providing contact for up to four physical links.

Table 21 defines the pin assignments.

5.2.3.7 SAS external receptacle connector

SAS devices with external ports shall use the SAS external receptacle connector. The SAS external receptacle connector is defined in SFF-8470 as the 4x configuration with thumbscrews. No special SAS keying is provided. The SAS external receptacle connector attaches to a SAS external cable plug connector, providing contact for up to four physical links.

Table 21 defines the pin assignments.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.2.3.8 SAS external connector pin assignments

Table 21 defines how the connector signal pairs are used in external connectors for applications using one, two, three, or four of the physical links. External cables should be labeled to indicate how many physical links are included (e.g., 1X, 2X, 3X, and 4X on each connector's housing).

Table 21 — Physical link usage in SAS external connector

Signal	Signal pin to use based on number of physical links supported by the cable			
	One	Two	Three	Four
Rx 0+	S1	S1	S1	S1
Rx 0-	S2	S2	S2	S2
Rx 1+	N/C	S3	S3	S3
Rx 1-	N/C	S4	S4	S4
Rx 2+	N/C	N/C	S5	S5
Rx 2-	N/C	N/C	S6	S6
Rx 3+	N/C	N/C	N/C	S7
Rx 3-	N/C	N/C	N/C	S8
Tx 3-	N/C	N/C	N/C	S9
Tx 3+	N/C	N/C	N/C	S10
Tx 2-	N/C	N/C	S11	S11
Tx 2+	N/C	N/C	S12	S12
Tx 1-	N/C	S13	S13	S13
Tx 1+	N/C	S14	S14	S14
Tx 0-	S15	S15	S15	S15
Tx 0+	S16	S16	S16	S16
SIGNAL GROUND	G1 - G9			
CHASSIS GROUND	Housing			
Key: N/C = not connected				

SIGNAL GROUND shall not be connected to CHASSIS GROUND in the cable connector.

5.2.4 Cables

5.2.4.1 SAS internal cables

SAS internal cables shall use SAS internal cable receptacle connector on the SAS target device end and a SATA-style cable receptacle (see ATA/ATAPI-7 V3) on the SAS initiator device or expander device end. The power and READY LED signal connection is vendor specific.

A SAS initiator device shall use a SATA-style host plug connector (see ATA/ATAPI-7 V3) for connection to the SAS internal cable. The signal assignment for the SAS initiator device or expander device with this connector shall be the same as that defined for a SATA host (see ATA/ATAPI-7 V3).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 44 shows destination signal assignments and a connection diagram for the SAS single-port internal cable.

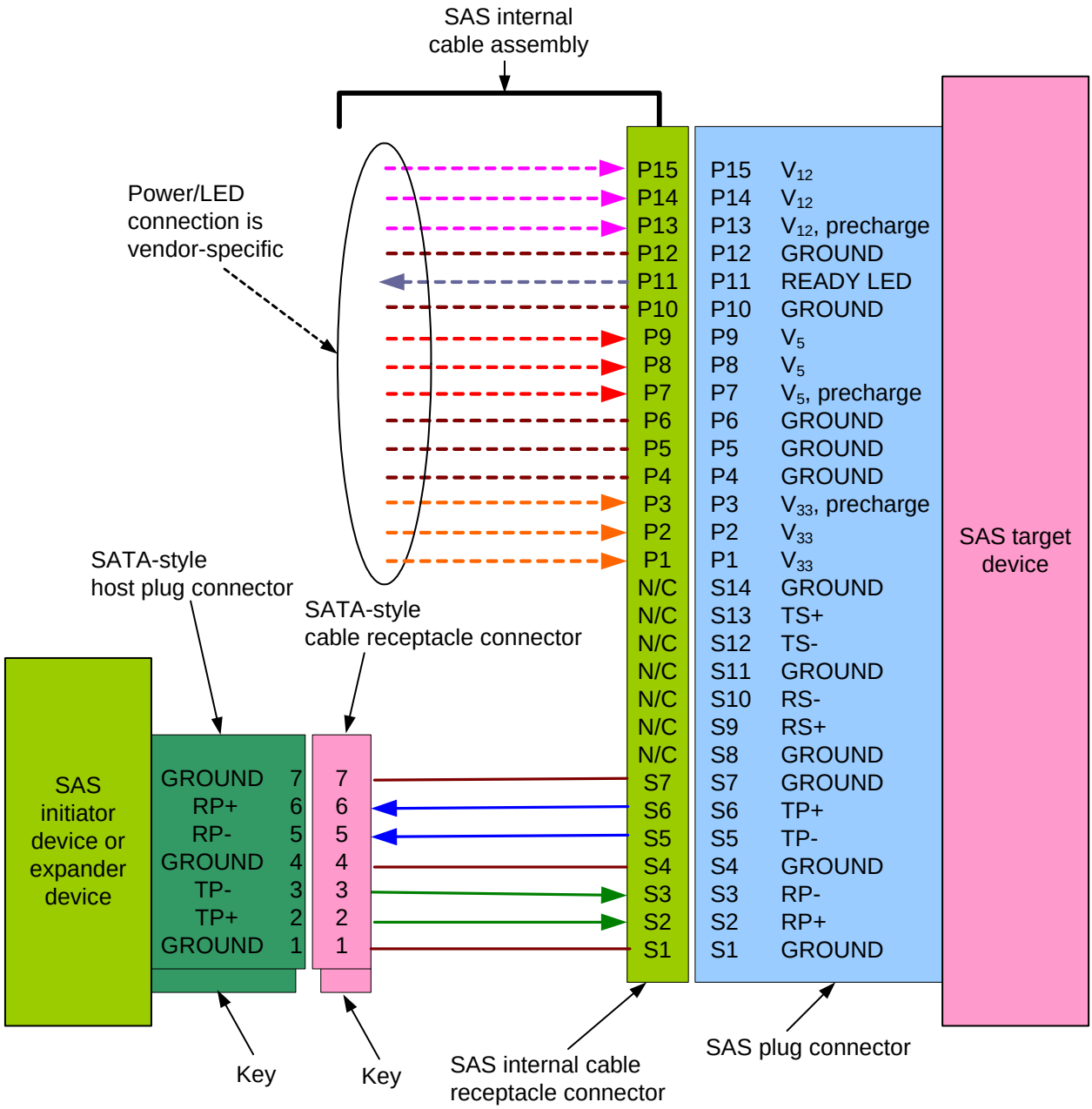


Figure 44 — SAS single-port internal cable assembly and destination pin assignments

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 45 shows destination signal assignments and a connection diagram for the SAS dual-port internal cable.

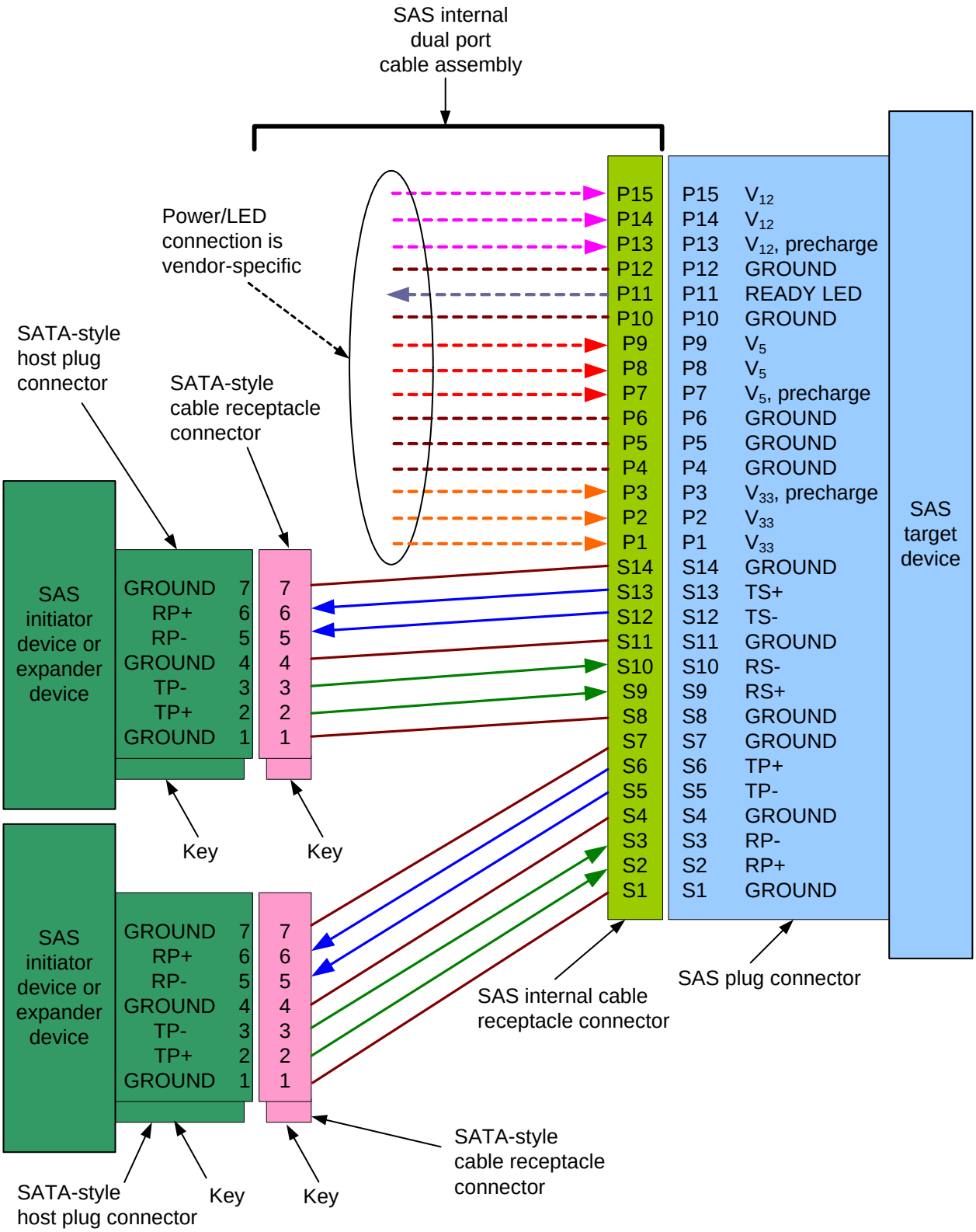


Figure 45 — SAS dual-port internal cable assembly and destination pin assignments

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5.2.4.2 SAS external cables

The SAS external cable connectors are defined in SFF-8470 as the 4x configuration with thumbscrews. The external cable does not include power or the READY LED signal.

Although the connector always supports four physical links, the cable may support one, two, three, or four physical links.

On external cable assemblies, the Tx signal from one connector shall be connected to the corresponding Rx signal of the other connector (e.g., Tx 0+ (S16) of connector shall connect to Rx 0+ (S1) of the other connector) (see 5.2.3.6).

SIGNAL GROUND shall not be connected to CHASSIS GROUND in the cable.

5.2.5 Backplanes

Backplane designs should follow the recommendations in SFF-8460.

5.3 Transmitter and receiver electrical characteristics

5.3.1 Compliance points

Signal behavior at separable connectors and integrated circuit package connections that satisfy the description for a compliance point require compliance with transmitter and receiver characteristics defined by this standard only if the connectors or integrated circuit package connections are identified as compliance points by the supplier of the parts that contain or comprise the candidate compliance point. Table 22 lists the compliance points.

Table 22 — Compliance points

Compliance point	Type	Description
IT	intra-enclosure	Internal connector; transmit serial port
IR	intra-enclosure	Internal connector; receive serial port
CT	inter-enclosure	External connector; transmit serial port
CR	inter-enclosure	External connector; receive serial port
XT	intra-enclosure	Expander or SAS initiator phy; transmit serial port
XR	intra-enclosure	Expander or SAS initiator phy; receive serial port

5.3.2 General interface specification

A TxRx connection is the complete simplex signal path between the output reference point of one phy or retimer to the input reference point of a second phy or retimer, over which a BER of $< 10^{-12}$ is achieved.

A TxRx connection segment is that portion of a TxRx connection delimited by separable connectors or changes in media.

This subclause defines the interfaces of the serial electrical signal at the compliance points IT, IR, CT, CR, XT, and XR in a TxRx connection. The IT, IR, CT, and CR points are located at the connectors of a TxRx connection.

Each compliant phy shall be compatible with this serial electrical interface to allow interoperability within a SAS environment. All TxRx connections described in this subclause shall exceed the BER objective of 10^{-12}. The parameters specified in this section support meeting this requirement under all conditions including the minimum input and output amplitude levels.

These signal specifications are consistent with using good quality passive cable assemblies constructed with shielded twinaxial cable with 24 gauge solid wire up to eight meters in length.

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 46 shows the transmitter transient test circuit.

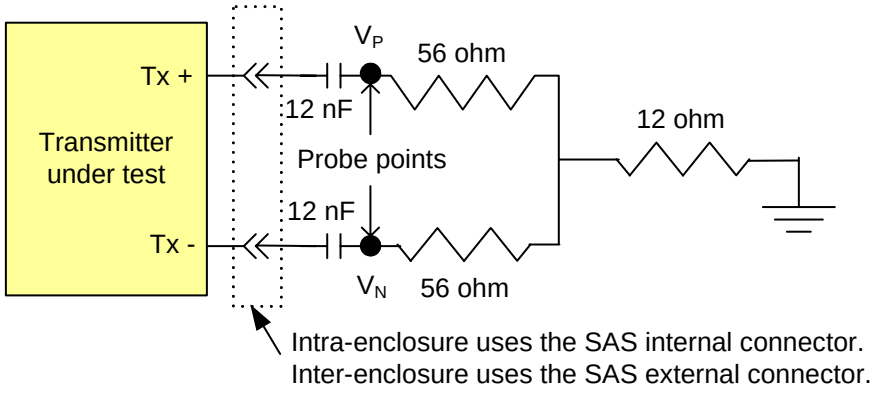


Figure 46 — Transmitter transient test circuit

Figure 47 shows the receiver transient test circuit.

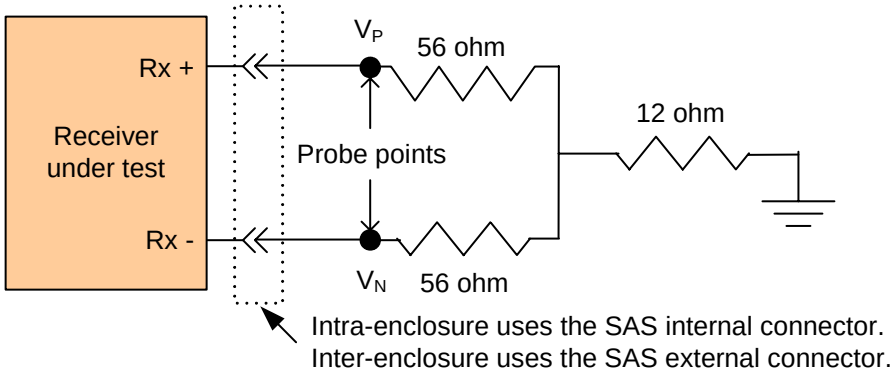


Figure 47 — Receiver transient test circuit

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 23 defines the general interface characteristics.

Table 23 — General interface characteristics

Characteristic	Units	1,5 Gbps	3,0 Gbps
Physical link rate	MBps	150	300
Bit rate (nominal)	Mbaud	1 500	3 000
Unit interval (UI)(nominal)	ps	666,$\overline{6}$	333,$\overline{3}$
Physical link rate tolerance at XR [b]	ppm	+350 / -5 350	+350 / -5 350
Physical link rate tolerance at IR and CR	ppm	± 100	± 100
Physical link rate tolerance at IT, CT, and XT	ppm	± 100	± 100
Media Impedance (nominal) [a]	ohm	100	100
A.C. coupling capacitor, maximum [c]	nF	12	12
Transmitter transients, maximum [d]	V	± 1,2	± 1,2
Receiver transients, maximum [d]	V	± 1,2	± 1,2
Receiver A.C. common mode voltage tolerance V_{CM}, minimum [e]	mV(P-P)	150	150
Receiver A.C. common mode frequency tolerance range F_{CM} [e]	MHz	2 to 200	2 to 200

[a] The media impedances are the differential impedances.
[b] Allows support for SATA devices with spread spectrum clocking (see ATA/ATAPI-7 V3). SAS initiator phys supporting being attached to SATA devices should also use these tolerances.
[c] The coupling capacitor value for A.C. coupled transmit and receive pairs.
[d] The maximum transmitter and receiver transients are measured at nodes V_P and V_N on the test loads shown in figure 46 (for the transmitter) and figure 47 (for the receiver) during all power state and mode transitions. Test conditions shall include the system power supply ramping at the fastest possible rate for both power on and power off conditions.
[e] Receivers shall tolerate sinusoidal common mode noise components within the peak-to-peak amplitude (V_{CM}) and the frequency range (F_{CM}).

5.3.3 Eye masks

5.3.3.1 Eye masks overview

The eye masks shown in this subclause shall be interpreted as graphical representations of the voltage and time limits on the signal at the compliance point. The time values between X1 and (1 - X1) cover all but 10^{-12} of the jitter population. The random content of the total jitter population has a range of ± 7 standard deviations.

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5.3.3.2 Receive eye mask at IR, CR, and XR

Figure 48 describes the receive eye mask. This eye mask applies to jitter after the application of a single pole high-pass frequency-weighting function that progressively attenuates jitter at 20 dB/decade below a frequency of ((bit rate) / 1 667).

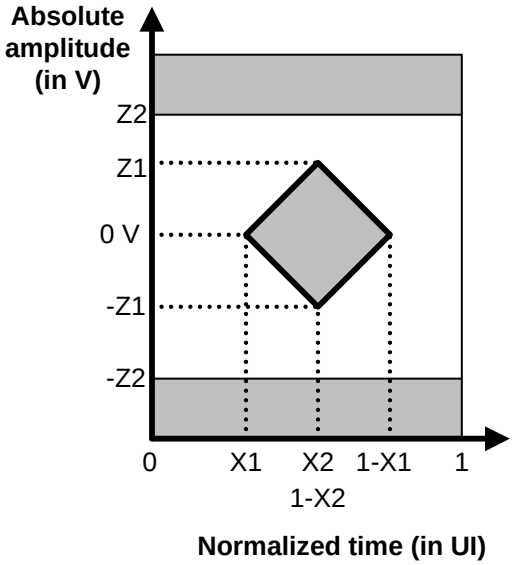


Figure 48 — Eye mask at IR, CR, and XR

Verifying compliance with the limits represented by the receive eye mask should be done with reverse channel traffic present in order that the effects of crosstalk are taken into account.

5.3.3.3 Jitter tolerance masks

Figure 49 describes the receive tolerance eye masks at IR, CR, and XR and shall be constructed using the X2 and Z2 values given in table 25. $X1_{OP}$ shall be half the value for total jitter in table 26 and $X1_{TOL}$ shall be half the value for total jitter in table 27, for jitter frequencies above ((bit rate) / 1 667).

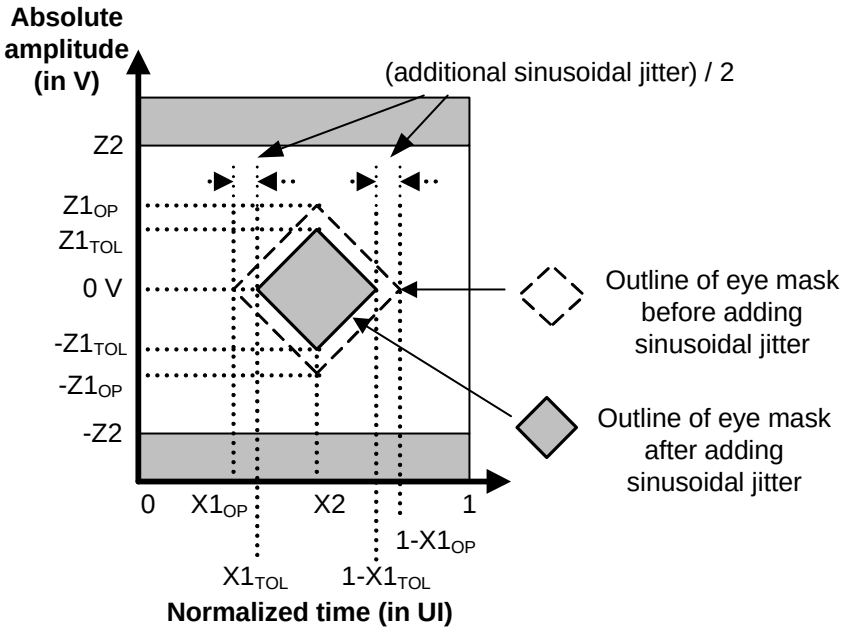


Figure 49 — Deriving a tolerance mask at IR, CR, or XR

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The leading and trailing edge slopes of figure 48 shall be preserved. As a result the amplitude value of Z1 is less than that given in table 25 and $Z1_{TOL}$ and $Z1_{OP}$ shall be defined from those slopes by the following equation:

$$Z1_{TOL} = Z1_{OP} \times \frac{X2_{OP} - (0,5 \times \text{additional sinusoidal jitter}) - X1_{OP}}{X2_{OP} - X1_{OP}}$$

where:

a) $Z1_{TOL}$ is the value for Z1 to be used for the tolerance masks; and
b) $Z1_{OP}$, $X1_{OP}$, and $X2_{OP}$ are the values in table 25 for Z1, X1, and X2.

The X1 points in the receive tolerance masks are greater than the X1 points in the receive masks, due to the addition of sinusoidal jitter.

Figure 50 defines the sinusoidal jitter mask.

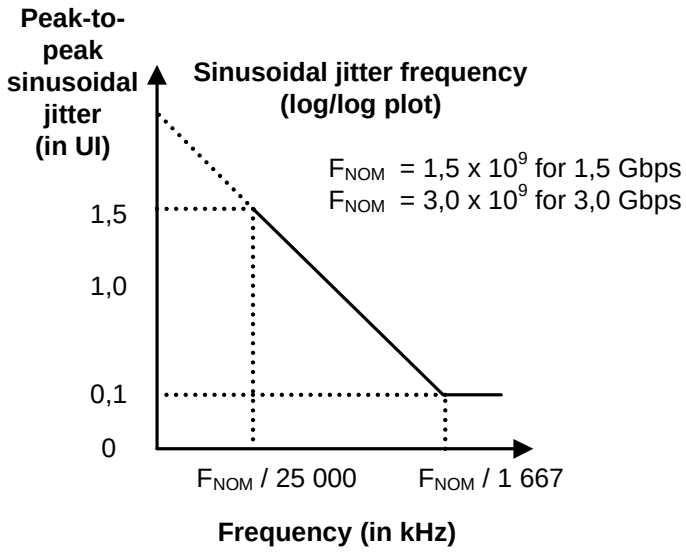


Figure 50 — Sinusoidal jitter mask

5.3.4 Signal characteristics at IT, CT, and XT

This subclause defines the inter-operability requirements of the signal at the transmitter end of a TxRx connection as measured into the zero-length test load specified in figure 52. All specifications are based on differential measurements.

The OOB sequence shall be performed at signal voltage levels corresponding to the lowest supported transfer rate. Expander phys supporting being attached to SATA devices shall use SATA 1.0 signal levels (see ATA/ATAPI-7 V3) during the first OOB sequence after a power on or hard reset if the 1,5 Gbps transfer rate is supported. As soon as COMSAS has been exchanged, the expander phy shall increase its transmit levels to the SAS voltage levels specified in table 25. If a COMINIT is not received within a hot-plug timeout at SATA 1.0 signal levels, the expander phy shall increase its transmit levels to the to SAS voltage levels and perform the OOB sequence again. If no COMINIT is received within a hot-plug timeout of the second OOB sequence the expander phy shall initiate another OOB sequence using SATA 1.0 signal levels. The expander phy shall continue alternating between sending COMINIT at SATA 1.0 signal levels and SAS signal levels until a COMINIT is received.

If the OOB sequence is completed at the SAS voltage level and a SATA device is detected rather than a SAS target device, the expander phy shall switch to SATA 1.0 voltage levels and repeat the OOB sequence.

NOTE 4 - SAS initiator phys supporting being attached to SATA devices may use the same algorithm as expander phys.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SAS initiator phys and SAS target phys shall transmit OOB signals at the lowest supported transfer rate using SAS signal levels.

Table 24 specifies the signal characteristics at IT, CT, and XT.

Table 24 — Signal characteristics at IT, CT, XT

Compliance point	Signal characteristic [a]	Units	1,5 Gbps	3,0 Gbps
IT, CT, XT	Skew [b]	ps	20	15
	Tx Off Voltage [c]	mV(P-P)	< 50	< 50
	Maximum rise/fall time [d]	ps	273	137
	Minimum rise/fall time [d]	ps	67	67
	Maximum transmitter output imbalance [e]	%	10	10
	OOB offset delta [f]	mV	± 25	± 25
	OOB common mode delta [g]	mV	± 50	± 50

[a] All tests in this table shall be performed with zero-length test load shown in figure 52.
[b] The skew measurement shall be made at the midpoint of the transition with a repeating 0101b pattern on the physical link. The same stable trigger, coherent to the data stream, shall be used for both the Tx+ and Tx- signals. Skew is defined as the time difference between the means of the midpoint crossing times of the Tx+ signal and the Tx- signal.
[c] The transmitter off voltage is the maximum A.C. voltage measured at compliance points IT, CT, and XT when the transmitter is unpowered or transmitting D.C. idle (e.g., during idle time of an OOB signal).
[d] Rise/fall times are measured from 20 % to 80 % of the transition with a repeating 0101b pattern on the physical link.
[e] The maximum difference between the V+ and V- A.C. RMS transmitter amplitudes measured on a CJTPAT test pattern (see 5.3.8) into the test load shown in figure 52, as a percentage of the average of the V+ and V- A.C. RMS amplitudes.
[f] The maximum difference in the average differential voltage (D.C. offset) component between the burst times and the idle times of an OOB signal.
[g] The maximum difference in the average of the common mode voltage between the burst times and the idle times of an OOB signal.

--`,,`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.3.5 Signal characteristics at IR, CR, and XR

Table 25 defines the compliance point requirements of the signal at the receiver end of a TxRx connection as measured into the test loads specified in figure 51 and figure 52.

Table 25 — Signal characteristics at IR, CR, and XR (Sheet 1 of 2)

Compliance point	Signal characteristic	Units	SATA	1,5 Gbps	3,0 Gbps
IR [e]	Jitter (see figure 48) [b]	N/A	N/A	See table 26	See table 26
	2 x Z2	mV(P-P)	N/A	1 600	1 600
	2 x Z1	mV(P-P)	N/A	325	275
	X1 [a]	UI	N/A	0,275	0,275
	X2	UI	N/A	0,50	0,50
	Skew [d]	ps	N/A	80	75
	Max voltage (non-op)	mV(P-P)	N/A	2 000	2 000
	Minimum OOB ALIGN burst amplitude [c]	mV(P-P)	N/A	240	240
	Maximum noise during OOB idle time [c]	mV(P-P)	N/A	120	120
	Max near-end crosstalk [f]	mV(P-P)	N/A	100	100
CR	Jitter (see figure 48) [b]	N/A	N/A	See table 26	See table 26
	2 x Z2	mV(P-P)	N/A	1 600	1 600
	2 x Z1	mV(P-P)	N/A	275	275
	X1 [a]	UI	N/A	0,275	0,275
	X2	UI	N/A	0,50	0,50
	Skew [d]	ps	N/A	80	75
	Max voltage (non-op)	mV(P-P)	N/A	2 000	2 000
	Minimum OOB ALIGN burst amplitude [c]	mV(P-P)	N/A	240	240
	Maximum noise during OOB idle time [c]	mV(P-P)	N/A	120	120
	Max near-end crosstalk [f]	mV(P-P)	N/A	100	100

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 25 — Signal characteristics at IR, CR, and XR (Sheet 2 of 2)

Compliance point	Signal characteristic	Units	SATA	1,5 Gbps	3,0 Gbps
XR	Jitter (see figure 48) [b]	N/A	See table 26	See table 26	See table 26
	2 x Z2	mV(P-P)	600	1 600	1 600
	2 x Z1	mV(P-P)	225	325	275
	X1 [a]	UI	0,275	0,275	0,275
	X2	UI	0,50	0,50	0,50
	Skew [d]	ps	50	80	75
	Max voltage (non-op)	mV(P-P)	2 000	2 000	2 000
	Minimum OOB ALIGN burst amplitude [c]	mV(P-P)	240	240	240
	Maximum noise during OOB idle time [c]	mV(P-P)	120	120	120
	Max near-end crosstalk [f]	mV(P-P)	< 50	100	100

[a] The value for X1 shall be half the value given for total jitter in table 26. The test or analysis shall include the effects of a single pole high-pass frequency-weighting function that progressively attenuates jitter at 20 dB/decade below a frequency of ((bit rate) / 1 667).

[b] The value for X1 applies at a total jitter probability of 10^{-12}. At this level of probability direct visual comparison between the mask and actual signals is not a valid method for determining compliance with the jitter output requirements.

[c] With a measurement bandwidth of 1,5 times the baud rate (i.e. 4,5 GHz for 3,0 Gbps).

[d] The skew measurement shall be made at the midpoint of the transition with a repeating 0101b pattern on the physical link. The same stable trigger, coherent to the data stream, shall be used for both the Rx+ and Rx- signals. Skew is defined as the time difference between the means of the midpoint crossing times of the Rx+ signal and the Rx- signal.

[e] If being attached to SATA devices is supported at the IR location, requirements of SATA shall be met at IR.

[f] Near-end crosstalk is the unwanted signal amplitude at receiver terminals DR, CR, and XR coupled from signals and noise sources other than the desired signal. Refer to SFF-8410.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

5.3.6 Jitter

Table 26 defines the maximum allowable jitter at IR, CR, and XR.

Table 26 — Maximum allowable jitter at IR, CR, XR

Compliance point	1,5 Gbps [a, b]		3,0 Gbps [a, b]	
	Deterministic jitter [e]	Total jitter [c, d, e, f]	Deterministic jitter [e]	Total jitter [c, d, e, f]
IR	0,35	0,55	0,35	0,55
CR	0,35	0,55	0,35	0,55
XR	0,35	0,55	0,35	0,55

[a] Units are in UI.
[b] The values for jitter in this section are measured at the average amplitude point.
[c] Total jitter is the sum of deterministic jitter and random jitter. If the actual deterministic jitter is less than the maximum specified, then the random jitter may increase as long as the total jitter does not exceed the specified maximum total jitter.
[d] Total jitter is specified at a probability of 10^{-12}.
[e] The deterministic and total values in this table apply to jitter after application of a single pole high-pass frequency-weighting function that progressively attenuates jitter at 20 dB/decade below a frequency of ((bit rate) / 1 667).
[f] If total jitter received at any point is less than the maximum allowed, then the jitter distribution of the signals is allowed to be asymmetric. The total jitter plus the magnitude of the asymmetry shall not exceed the allowed maximum total jitter. The numerical difference between the average of the peaks with a BER < 10^{-12} and the average of the individual events is the measure of the asymmetry. Jitter peak-to-peak measured < (maximum total jitter - |Asymmetry|).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.3.7 Receiver jitter tolerance

Table 27 defines the amount of jitter the receiver shall tolerate at IR, CR, and XR.

Table 27 — Receiver jitter tolerance

Compliance point	1,5 Gbps [a]			3,0 Gbps [a]		
	Sinusoidal jitter [b, c]	Deterministic jitter [e, f, h]	Total jitter [h]	Sinusoidal jitter [b, d]	Deterministic jitter [e, g, h]	Total jitter [h]
IR	0,10	0,35	0,65	0,10	0,35	0,65
CR	0,10	0,35	0,65	0,10	0,35	0,65
XR	0,10	0,35	0,65	0,10	0,35	0,65

[a] Units are in UI.
[b] The jitter values given are normative for a combination of deterministic jitter, random jitter, and sinusoidal jitter that receivers shall be able to tolerate without exceeding a BER of 10^{-12}. Receivers shall tolerate sinusoidal jitter of progressively greater amplitude at lower frequencies, according to the mask in figure 50 with the same deterministic jitter and random jitter levels as were used in the high frequency sweep.
[c] Sinusoidal swept frequency: 900 kHz to > 5 MHz.
[d] Sinusoidal swept frequency: 1 800 kHz to > 5 MHz.
[e] No value is given for random jitter. For compliance with this standard, the actual random jitter amplitude shall be the value that brings total jitter to the stated value at a probability of 10^{-12}. The additional 0,1 UI of sinusoidal jitter is added to ensure the receiver has sufficient operating margin in the presence of external interference.
[f] Deterministic jitter: 900 kHz to 750 MHz.
[g] Deterministic jitter: 1 800 kHz to 1 500 MHz.
[h] The deterministic and total values in this table apply to jitter after application of a single pole high-pass frequency-weighting function that progressively attenuates jitter at 20 dB/decade below a frequency of ((bit rate) / 1 667).

5.3.8 Compliant jitter test pattern (CJTPAT)

The CJTPAT within a compliant protocol frame shall be used for all jitter testing unless otherwise specified. Annex A defines the required pattern on the physical link and information regarding special considerations for scrambling and running disparity.

5.3.9 Impedance specifications

Table 28 defines impedance requirements.

Table 28 — Impedance requirements (Sheet 1 of 2)

Requirement	Units	1,5 Gbps	3,0 Gbps
Time domain reflectometer rise time 20 % to 80 % [a, b]	ps	100	50
Media (PCB or cable)			
Differential impedance [b, c, d]	ohm	100 ± 10	100 ± 10
Differential impedance imbalance [b, c, d, g]	ohm	5	5
Common mode impedance [b, c, d]	ohm	32,5 ± 7,5	32,5 ± 7,5
Mated connectors			
Differential impedance [b, c, d]	ohm	100 ± 15	100 ± 15
Differential impedance imbalance [b, c, d, g]	ohm	5	5
Common mode impedance [b, c, d]	ohm	32,5 ± 7,5	32,5 ± 7,5
Receiver termination			

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 28 — Impedance requirements (Sheet 2 of 2)

Requirement	Units	1,5 Gbps	3,0 Gbps
Differential impedance [b, e, f]	ohm	100 ± 15	100 ± 15
Differential impedance imbalance [b, e, f, g]	ohm	5	5
Receiver termination time constant [b, e, f]	ps	150 max	100 max
Common mode impedance [b, e]	ohm	20 min/40 max	20 min/40 max
Transmitter source termination			
Differential impedance [b]	ohm	60 min/115 max	60 min/115 max
Differential impedance imbalance [b, g]	ohm	5	5
Common mode impedance [b]	ohm	15 min/40 max	15 min/40 max

[a] All times indicated for time domain reflectometer measurements are recorded times. Recorded times are twice the transit time of the time domain reflectometer signal.

[b] All measurements are made through mated connector pairs.

[c] The media impedance measurement identifies the impedance mismatches present in the media when terminated in its characteristic impedance. This measurement excludes mated connectors at both ends of the media, when present, but includes any intermediate connectors or splices. The mated connectors measurement applies only to the mated connector pair at each end, as applicable.

[d] Where the media has an electrical length of > 4 ns the procedure detailed in SFF-8410, or an equivalent procedure, shall be used to determine the impedance.

[e] The receiver termination impedance specification applies to all receivers in a TxRx connection and covers all time points between the connector nearest the receiver, the receiver, and the transmission line terminator. This measurement shall be made from that connector.

[f] At the time point corresponding to the connection of the receiver to the transmission line the input capacitance of the receiver and its connection to the transmission line may cause the measured impedance to fall below the minimum impedances specified in this table. The area of the impedance dip (amplitude as ρ, the reflection coefficient, and duration in time) caused by this capacitance is the receiver termination time constant. The receiver time constant shall not be greater than the values shown in this table. An approximate value for the receiver termination time constant is given by the product of the amplitude of the dip (as ρ) and its width (in ps) measured at the half amplitude point. The amplitude is defined as being the difference in the reflection coefficient between the reflection coefficient at the nominal impedance and the reflection coefficient at the minimum impedance point. The value of the receiver excess input capacitance is given by the following equation:

$$C = \frac{\text{receiver termination time constant}}{(R0 \,||\, RR)}$$

where (R0 || RR) is the parallel combination of the transmission line characteristic impedance and termination resistance at the receiver.

[g] The difference in measured impedance to ground on the plus and minus terminals on the interconnect, transmitter or receiver, with a differential test signal applied to those terminals.

5.3.10 Electrical TxRx connections

TxRx connections may be divided into TxRx connection segments. In a single TxRx connection individual TxRx connection segments may be formed from differing media and materials, including traces on printed wiring boards and optical fibers. This subclause applies only to TxRx connection segments that are formed from electrically conductive media.

Each electrical TxRx connection segment shall comply with the impedance requirements of table 28 for the media from which they are formed. An equalizer network, if present, shall be part of the TxRx connection.

TxRx connections that are composed entirely of electrically conducting media shall be applied only to homogenous ground applications (e.g., between devices within an enclosure or rack, or between enclosures interconnected by a common ground return or ground plane).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5.3.11 Transmitter characteristics

For all inter-enclosure TxRx connections, the transmitter shall be A.C. coupled to the interconnect through a transmission network.

For intra-enclosure TxRx connections the expander transmitter shall be A.C. coupled to the interconnect. Other transmitters may be A.C. or D.C. coupled.

A combination of a zero-length test load and the transmitter compliance transfer function (TCTF) test load methodology is used for the specification of the inter-enclosure and intra-enclosure transmitter characteristics. This methodology specifies the transmitter signal at the test points on the required test loads. The transmitter shall use the same settings (e.g., pre-emphasis, voltage swing) with both the zero-length test load and the TCTF test load. The signal specifications at IR, CR, and XR shall be met under each of these loading conditions.

The TCTF is the mathematical statement of the transfer function through which the transmitter shall be capable of producing acceptable signals as defined by a receive mask. The transmission magnitude response of the TCTF in dB is given by the following equation for 1,5 Gbps:

$$|S_{21}| = -20 \times \log_{10}(e) \times ((6,5 \times 10^{-6} \times f^{0,5}) + (2,0 \times 10^{-10} \times f) + (3,3 \times 10^{-20} \times f^{2})) \text{ dB}$$

for 50 MHz < f < 1,5 GHz, and:

$$|S_{21}| = -5,437 \text{ dB}$$

for 1,5 GHz < f < 5,0 GHz,

where:

f is the signal frequency in hertz.

The transmission magnitude response of the TCTF in dB is given by the following equation for 3,0 Gbps:

$$|S_{21}| = -20 \times \log_{10}(e) \times ((6,5 \times 10^{-6} \times f^{0,5}) + (2,0 \times 10^{-10} \times f) + (3,3 \times 10^{-20} \times f^{2})) \text{ dB}$$

for 50 MHz < f < 3,0 GHz, and:

$$|S_{21}| = -10,884 \text{ dB}$$

for 3,0 GHz < f < 5,0 GHz,

where:

f is the signal frequency in hertz.

The TCTF is used to specify the requirements on transmitters that may or may not incorporate pre-emphasis or other forms of compensation. A compliance interconnect is any physical interconnect with loss equal to or greater than that of the TCTF at the above frequencies that also meets the ISI loss requirements shown in figure 53 and figure 54.

Compliance with the TCTF test load requirement shall be determined either:

a) by measuring the signal produced by the transmitter through a physical compliance interconnect attached to the transmitter; or
b) by mathematically processing through the TCTF the signal captured using a zero-length test load.

Compliance with the zero-length test load requirement shall be determined by measurement made across a load equivalent to the zero-length load shown in figure 52.

For both test load cases, the transmitter shall deliver the output voltages and timing listed in table 25 at the designated compliance points. The default mask shall be CR for inter-cabinet TxRx connections and IR for intra-cabinet TxRx connections. The eye masks are shown in 5.3.3.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 51 shows the compliance interconnect test load.

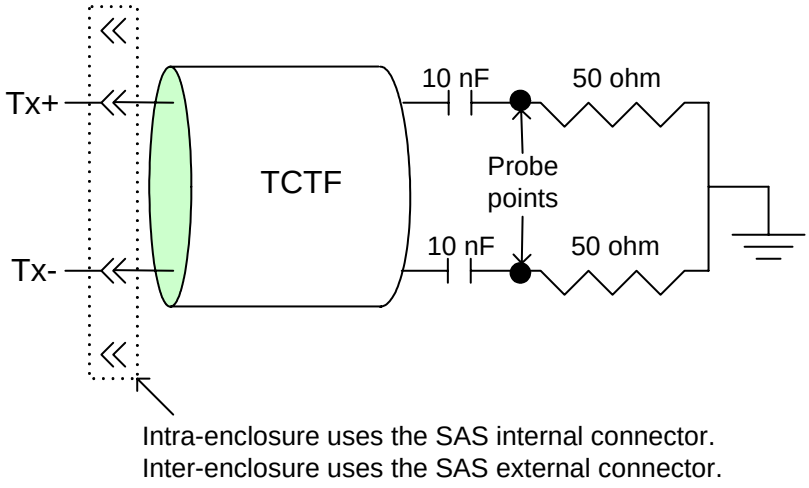


Figure 51 — Compliance interconnect test load

Figure 52 shows the zero-length test load.

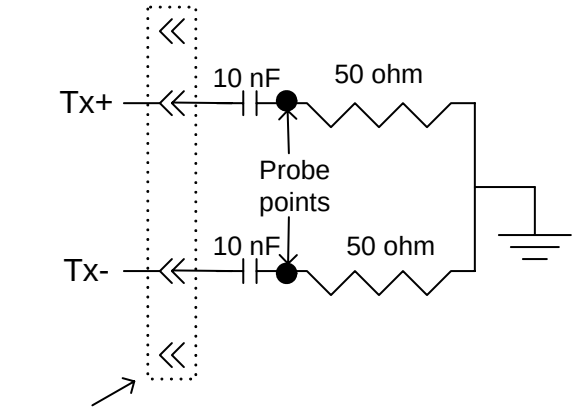


Figure 52 — Zero-length test load

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 53 shows an ISI loss example at 3,0 Gbps.

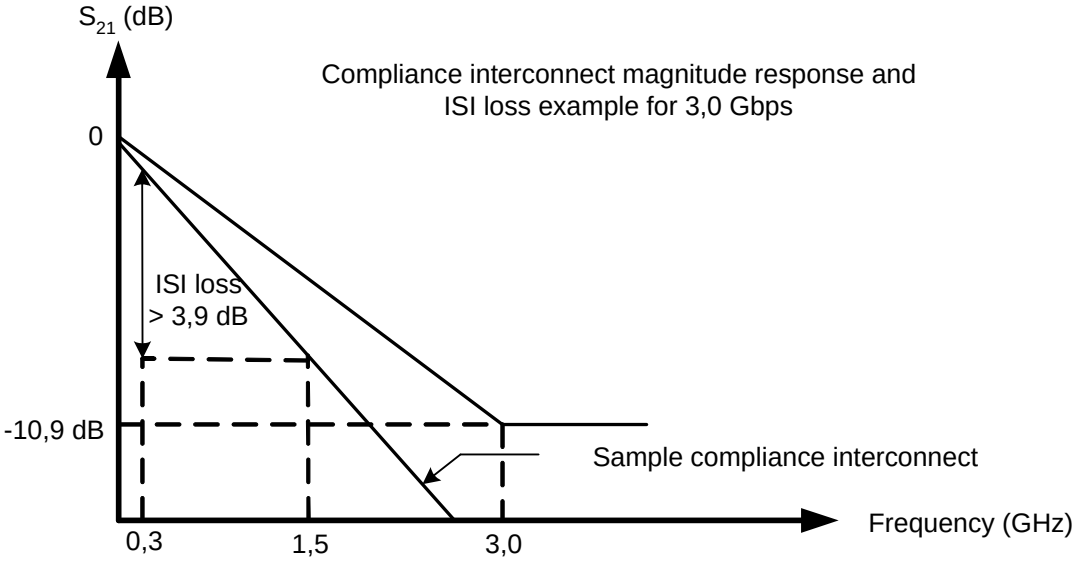


Figure 53 — ISI loss example at 3,0 Gbps

Figure 54 shows an ISI loss example at 1,5 Gbps.

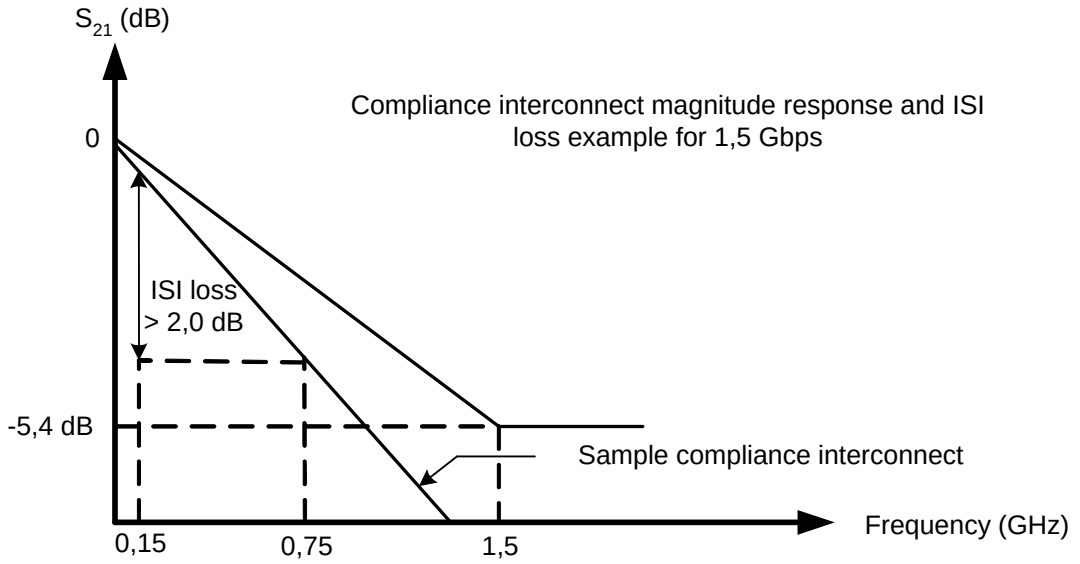


Figure 54 — ISI loss example at 1,5 Gbps

5.3.12 Receiver characteristics

The receiver shall be A.C. coupled to the interconnect through a receive network. The receive network shall terminate the TxRx connection by a 100 ohm equivalent impedance as specified in table 28.

The receiver shall operate within a BER of 10^{-12} when a SAS signal with valid voltage and timing characteristics is delivered to the compliance point from a 100 ohm source. The received SAS signal shall be considered valid if it meets the voltage and timing limits specified in table 25.

Additionally the receiver shall also operate within the BER objective when the signal at a receiving phy has the additional sinusoidal jitter present that is specified in table 27 and the common mode signal V_{CM} over frequency range F_{CM} as specified in table 23. The jitter tolerance figure is given in figure 49 for all Rx compliance points in a TxRx connection. The figure given assumes that any external interference occurs prior to the point at which the test is applied. When testing the jitter tolerance capability of a receiver, the additional

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

0,1 UI of sinusoidal jitter may be reduced by an amount proportional to the actual externally induced interference between the application point of the test and the input to the receiving phy. The additional jitter reduces the eye opening in both voltage and time.

5.3.13 Spread spectrum clocking

Phys shall not transmit with spread spectrum clocking. Expander phys that support being attached to SATA devices shall support receiving with spread spectrum clocking (see ATA/ATAPI-7 V3). The expander device shall retime data from a SATA device with an internal clock before forwarding to the rest of the SAS domain.

NOTE 5 - If SAS initiator devices support being attached to SATA devices, they should follow the same rules as expander phys.

5.3.14 Non-tracking clock architecture

Phys shall be designed with a non-tracking clock architecture; the receive clock derived from the received bit stream shall not be used as the transmit clock. Expander phys that support being attached to SATA devices shall tolerate clock tracking by the SATA device.

NOTE 6 - If SAS initiator devices support being attached to SATA devices, they should follow the same rules as expander phys.

5.4 READY LED signal electrical characteristics

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.

All SAS target devices using the SAS plug connector (see 5.2.3.2) shall support the READY LED signal.

The READY LED signal is designed to pull down the cathode of an LED using an open collector or open drain transmitter circuit. The LED and the current limiting circuitry shall be external to the SAS target device.

Table 29 describes the output characteristics of the READY LED signal.

Table 29 — Output characteristics of the READY LED signal

State	Test condition	Requirement
Negated (LED off)	$0\ V \le V_{OH} \le 3{,}6\ V$	$-100\ \mu A < I_{OH} < 100\ \mu A$
Asserted (LED on)	$I_{OL} = 15\ mA$	$0 \le V_{OL} \le 0{,}225\ V$

The READY LED signal behavior is defined in 10.4.1.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6 Phy layer

6.1 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.

6.2 Encoding (8b10b)

6.2.1 Encoding overview

All data bytes transferred in SAS are encoded into 10-bit data characters using 8b10b coding. Additional characters not related to data bytes are called control characters.

All characters transferred in SAS are grouped into four-character sequences called dwords. A primitive is a dword whose first character is a control character and remaining three characters are data characters.

Primitives are defined with both negative and positive starting running disparity (see 6.3.3.1). SAS defines primitives starting with the K28.5 and K28.6 control characters. Table 30 shows special character usage.

Table 30 — Special character usage

First character	Usage in SAS	Usage in SATA
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	SATA_ERROR (used on SATA physical links)	Not used
Dxx.y	Data	Data

Primitives are defined in 7.2.

A data dword is a dword starting with a data character.

Running disparity shall be maintained separately on each physical link. 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. Physical links may or may not begin operation with the same disparity after the reset sequence.

6.2.2 8b10b coding introduction

Information to be transmitted across a physical link shall be encoded eight bits at a time into a 10-bit transmission character and then transmitted serially bit-by-bit across the physical link. Information received over the physical link shall be collected ten bits at a time, and those transmission characters that are used for data, called data characters, shall be decoded into the correct 8-bit codes. The 10-bit transmission code supports all 256 8-bit combinations. Some of the remaining transmission characters, referred to as special characters, are used for functions that are to be distinguishable from the contents of a frame.

The encodings defined by the transmission code ensure that sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some of the special characters of the transmission code contain a distinct and easily recognizable bit pattern (a comma) which assists a receiver in achieving word alignment on the incoming bit stream.

6.2.3 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. It also describes the conventions used to name valid transmission characters. This text is provided for the purposes of terminology clarification only.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

An unencoded information byte is composed of eight information bits A, B, C, D, E, F, G, H and the control variable Z. This information is encoded into the bits a, b, c, d, e, i, f, g, h, j of a 10-bit transmission character.

An information bit contains either a binary zero or a binary one. A control variable has either the value D or the value K. When the control variable associated with an unencoded information byte contains the value D, that byte is referred to as a valid data byte. When the control variable associated with an unencoded information byte contains the value K, that byte is referred to as a special code.

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 31.

Table 31 — 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 transmission character has been given a name using the following convention:

Zxx.y

where:

a) Z is the control variable of the unencoded information byte. The value of Z is used to indicate whether the transmission character is a data character (Z = D) or a special character (Z = K);
b) 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; and
c) 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 32 shows the conversion from byte notation to the transmission character naming convention described above.

Table 32 — Conversion 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
Transmission character name	K 28 . 5

Most Kxx.y combinations do not result in valid transmission characters within the 8b10b coding scheme. Only those combinations that result in special characters as specified by table 34 are considered valid.

6.3 Character encoding and decoding

6.3.1 Introduction

This subclause describes how to select valid transmission characters (encoding) and check the validity of received transmission characters (decoding). It also specifies the ordering rules to be followed when transmitting the bits within a character and the characters within the higher-level constructs specified by the document (i.e., primitives and frames).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.3.2 Transmission order

Within the definition of the 8b10b transmission code, the bit positions of the transmission characters are 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. Bit i shall be transmitted between bit e and bit f, rather than in the order that would be indicated by the letters of the alphabet.

Characters within primitives shall be transmitted sequentially beginning with the special 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.

The contents of a frame shall be transmitted sequentially beginning with the primitive used to denote the start of frame 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 is transmitted.

6.3.3 Valid and invalid transmission characters

6.3.3.1 Definitions

Table 33 and table 34 define the valid data characters (Dxx.y characters) and valid special characters (Kxx.y characters), respectively, and shall be used for both generating valid transmission characters (encoding) and checking the validity of received transmission characters (decoding). Each Valid-Data-Byte or special code entry has two columns that represent two (not necessarily different) transmission characters, corresponding to the current value of the running disparity (current RD - or current RD +). Running disparity is a binary parameter with either the value negative (-) or the value positive (+). The running disparity at the beginning of a primitive is the beginning running disparity (beginning RD).

After powering on, the transmitter may initialize the current RD to positive or negative. Upon transmission of any transmission character, the transmitter shall calculate a new value for its running disparity based on the contents of the transmitted character.

After powering on or exiting diagnostic mode (the definition of diagnostic mode is beyond the scope of this standard), the receiver should assume either the positive or negative value for its initial running disparity. Upon reception of any transmission character, the receiver shall determine whether the transmission character is valid or invalid according to the following rules and shall calculate a new value for its running disparity based on the contents of the received character.

The following rules for running disparity shall be used to calculate the new running disparity value for transmission characters that have been transmitted (i.e. transmitter's running disparity) and that have been received (i.e. receiver's running disparity).

Running disparity for a transmission character shall be calculated on the basis of sub-blocks, where the first six bits ('abcdei' b) form one sub-block (six-bit sub-block) and the second four bits ('fghj' b) form the other sub-block (four-bit sub-block). Running disparity at the beginning of the six-bit sub-block is the running disparity at the end of the last transmission character. Running disparity at the beginning of the four-bit sub-block is the running disparity at the end of the six-bit sub-block. Running disparity at the end of the transmission character is the running disparity at the end of the four-bit sub-block.

Running disparity for the sub-blocks shall be calculated as follows:

a) Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111b, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011b.
b) Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000b, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100b.
c) Otherwise, running disparity 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 are disparity neutral. In order to limit the run length of zeros or ones between sub-blocks, the 8b10b transmission code rules specify that sub-blocks encoded as 000111b or 0011b are generated only when the running disparity at the beginning of the sub-block is positive; thus, running disparity at the end of these sub-blocks shall also be positive. Likewise, sub-blocks containing

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

111000b or 1100b are generated only when the running disparity at the beginning of the sub-block is negative; thus, running disparity at the end of these sub-blocks shall also be negative.

Table 33 defines the valid data characters (Dxx.y characters).

Table 33 — Valid data characters (Sheet 1 of 3)

Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)	Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)
D00.0	000 00000	100111 0100	011000 1011	D00.1	001 00000	100111 1001	011000 1001
D01.0	000 00001	011101 0100	100010 1011	D01.1	001 00001	011101 1001	100010 1001
D02.0	000 00010	101101 0100	010010 1011	D02.1	001 00010	101101 1001	010010 1001
D03.0	000 00011	110001 1011	110001 0100	D03.1	001 00011	110001 1001	110001 1001
D04.0	000 00100	110101 0100	001010 1011	D04.1	001 00100	110101 1001	001010 1001
D05.0	000 00101	101001 1011	101001 0100	D05.1	001 00101	101001 1001	101001 1001
D06.0	000 00110	011001 1011	011001 0100	D06.1	001 00110	011001 1001	011001 1001
D07.0	000 00111	111000 1011	000111 0100	D07.1	001 00111	111000 1001	000111 1001
D08.0	000 01000	111001 0100	000110 1011	D08.1	001 01000	111001 1001	000110 1001
D09.0	000 01001	100101 1011	100101 0100	D09.1	001 01001	100101 1001	100101 1001
D10.0	000 01010	010101 1011	010101 0100	D10.1	001 01010	010101 1001	010101 1001
D11.0	000 01011	110100 1011	110100 0100	D11.1	001 01011	110100 1001	110100 1001
D12.0	000 01100	001101 1011	001101 0100	D12.1	001 01100	001101 1001	001101 1001
D13.0	000 01101	101100 1011	101100 0100	D13.1	001 01101	101100 1001	101100 1001
D14.0	000 01110	011100 1011	011100 0100	D14.1	001 01110	011100 1001	011100 1001
D15.0	000 01111	010111 0100	101000 1011	D15.1	001 01111	010111 1001	101000 1001
D16.0	000 10000	011011 0100	100100 1011	D16.1	001 10000	011011 1001	100100 1001
D17.0	000 10001	100011 1011	100011 0100	D17.1	001 10001	100011 1001	100011 1001
D18.0	000 10010	010011 1011	010011 0100	D18.1	001 10010	010011 1001	010011 1001
D19.0	000 10011	110010 1011	110010 0100	D19.1	001 10011	110010 1001	110010 1001
D20.0	000 10100	001011 1011	001011 0100	D20.1	001 10100	001011 1001	001011 1001
D21.0	000 10101	101010 1011	101010 0100	D21.1	001 10101	101010 1001	101010 1001
D22.0	000 10110	011010 1011	011010 0100	D22.1	001 10110	011010 1001	011010 1001
D23.0	000 10111	111010 0100	000101 1011	D23.1	001 10111	111010 1001	000101 1001
D24.0	000 11000	110011 0100	001100 1011	D24.1	001 11000	110011 1001	001100 1001
D25.0	000 11001	100110 1011	100110 0100	D25.1	001 11001	100110 1001	100110 1001
D26.0	000 11010	010110 1011	010110 0100	D26.1	001 11010	010110 1001	010110 1001
D27.0	000 11011	110110 0100	001001 1011	D27.1	001 11011	110110 1001	001001 1001
D28.0	000 11100	001110 1011	001110 0100	D28.1	001 11100	001110 1001	001110 1001
D29.0	000 11101	101110 0100	010001 1011	D29.1	001 11101	101110 1001	010001 1001
D30.0	000 11110	011110 0100	100001 1011	D30.1	001 11110	011110 1001	100001 1001
D31.0	000 11111	101011 0100	010100 1011	D31.1	001 11111	101011 1001	010100 1001
D00.2	010 00000	100111 0101	011000 0101	D00.3	011 00000	100111 0011	011000 1100
D01.2	010 00001	011101 0101	100010 0101	D01.3	011 00001	011101 0011	100010 1100
D02.2	010 00010	101101 0101	010010 0101	D02.3	011 00010	101101 0011	010010 1100
D03.2	010 00011	110001 0101	110001 0101	D03.3	011 00011	110001 1100	110001 0011
D04.2	010 00100	110101 0101	001010 0101	D04.3	011 00100	110101 0011	001010 1100
D05.2	010 00101	101001 0101	101001 0101	D05.3	011 00101	101001 1100	101001 0011
D06.2	010 00110	011001 0101	011001 0101	D06.3	011 00110	011001 1100	011001 0011
D07.2	010 00111	111000 0101	000111 0101	D07.3	011 00111	111000 1100	000111 0011
D08.2	010 01000	111001 0101	000110 0101	D08.3	011 01000	111001 0011	000110 1100
D09.2	010 01001	100101 0101	100101 0101	D09.3	011 01001	100101 1100	100101 0011
D10.2	010 01010	010101 0101	010101 0101	D10.3	011 01010	010101 1100	010101 0011
D11.2	010 01011	110100 0101	110100 0101	D11.3	011 01011	110100 1100	110100 0011
D12.2	010 01100	001101 0101	001101 0101	D12.3	011 01100	001101 1100	001101 0011
D13.2	010 01101	101100 0101	101100 0101	D13.3	011 01101	101100 1100	101100 0011
D14.2	010 01110	011100 0101	011100 0101	D14.3	011 01110	011100 1100	011100 0011
D15.2	010 01111	010111 0101	101000 0101	D15.3	011 01111	010111 0011	101000 1100
D16.2	010 10000	011011 0101	100100 0101	D16.3	011 10000	011011 0011	100100 1100
D17.2	010 10001	100011 0101	100011 0101	D17.3	011 10001	100011 1100	100011 0011
D18.2	010 10010	010011 0101	010011 0101	D18.3	011 10010	010011 1100	010011 0011
D19.2	010 10011	110010 0101	110010 0101	D19.3	011 10011	110010 1100	110010 0011
D20.2	010 10100	001011 0101	001011 0101	D20.3	011 10100	001011 1100	001011 0011

--`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 33 — Valid data characters (Sheet 2 of 3)

Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)	Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)
D21.2	010 10101	101010 0101	101010 0101	D21.3	011 10101	101010 1100	101010 0011
D22.2	010 10110	011010 0101	011010 0101	D22.3	011 10110	011010 1100	011010 0011
D23.2	010 10111	111010 0101	000101 0101	D23.3	011 10111	111010 0011	000101 1100
D24.2	010 11000	110011 0101	001100 0101	D24.3	011 11000	110011 0011	001100 1100
D25.2	010 11001	100110 0101	100110 0101	D25.3	011 11001	100110 1100	100110 0011
D26.2	010 11010	010110 0101	010110 0101	D26.3	011 11010	010110 1100	010110 0011
D27.2	010 11011	110110 0101	001001 0101	D27.3	011 11011	110110 0011	001001 1100
D28.2	010 11100	001110 0101	001110 0101	D28.3	011 11100	001110 1100	001110 0011
D29.2	010 11101	101110 0101	010001 0101	D29.3	011 11101	101110 0011	010001 1100
D30.2	010 11110	011110 0101	100001 0101	D30.3	011 11110	011110 0011	100001 1100
D31.2	010 11111	101011 0101	010100 0101	D31.3	011 11111	101011 0011	010100 1100
D00.4	100 00000	100111 0010	011000 1101	D00.5	101 00000	100111 1010	011000 1010
D01.4	100 00001	011101 0010	100010 1101	D01.5	101 00001	011101 1010	100010 1010
D02.4	100 00010	101101 0010	010010 1101	D02.5	101 00010	101101 1010	010010 1010
D03.4	100 00011	110001 1101	110001 0010	D03.5	101 00011	110001 1010	110001 1010
D04.4	100 00100	110101 0010	001010 1101	D04.5	101 00100	110101 1010	001010 1010
D05.4	100 00101	101001 1101	101001 0010	D05.5	101 00101	101001 1010	101001 1010
D06.4	100 00110	011001 1101	011001 0010	D06.5	101 00110	011001 1010	011001 1010
D07.4	100 00111	111000 1101	000111 0010	D07.5	101 00111	111000 1010	000111 1010
D08.4	100 01000	111001 0010	000110 1101	D08.5	101 01000	111001 1010	000110 1010
D09.4	100 01001	100101 1101	100101 0010	D09.5	101 01001	100101 1010	100101 1010
D10.4	100 01010	010101 1101	010101 0010	D10.5	101 01010	010101 1010	010101 1010
D11.4	100 01011	110100 1101	110100 0010	D11.5	101 01011	110100 1010	110100 1010
D12.4	100 01100	001101 1101	001101 0010	D12.5	101 01100	001101 1010	001101 1010
D13.4	100 01101	101100 1101	101100 0010	D13.5	101 01101	101100 1010	101100 1010
D14.4	100 01110	011100 1101	011100 0010	D14.5	101 01110	011100 1010	011100 1010
D15.4	100 01111	010111 0010	101000 1101	D15.5	101 01111	010111 1010	101000 1010
D16.4	100 10000	011011 0010	100100 1101	D16.5	101 10000	011011 1010	100100 1010
D17.4	100 10001	100011 1101	100011 0010	D17.5	101 10001	100011 1010	100011 1010
D18.4	100 10010	010011 1101	010011 0010	D18.5	101 10010	010011 1010	010011 1010
D19.4	100 10011	110010 1101	110010 0010	D19.5	101 10011	110010 1010	110010 1010
D20.4	100 10100	001011 1101	001011 0010	D20.5	101 10100	001011 1010	001011 1010
D21.4	100 10101	101010 1101	101010 0010	D21.5	101 10101	101010 1010	101010 1010
D22.4	100 10110	011010 1101	011010 0010	D22.5	101 10110	011010 1010	011010 1010
D23.4	100 10111	111010 0010	000101 1101	D23.5	101 10111	111010 1010	000101 1010
D24.4	100 11000	110011 0010	001100 1101	D24.5	101 11000	110011 1010	001100 1010
D25.4	100 11001	100110 1101	100110 0010	D25.5	101 11001	100110 1010	100110 1010
D26.4	100 11010	010110 1101	010110 0010	D26.5	101 11010	010110 1010	010110 1010
D27.4	100 11011	110110 0010	001001 1101	D27.5	101 11011	110110 1010	001001 1010
D28.4	100 11100	001110 1101	001110 0010	D28.5	101 11100	001110 1010	001110 1010
D29.4	100 11101	101110 0010	010001 1101	D29.5	101 11101	101110 1010	010001 1010
D30.4	100 11110	011110 0010	100001 1101	D30.5	101 11110	011110 1010	100001 1010
D31.4	100 11111	101011 0010	010100 1101	D31.5	101 11111	101011 1010	010100 1010
D00.6	110 00000	100111 0110	011000 0110	D00.7	111 00000	100111 0001	011000 1110
D01.6	110 00001	011101 0110	100010 0110	D01.7	111 00001	011101 0001	100010 1110
D02.6	110 00010	101101 0110	010010 0110	D02.7	111 00010	101101 0001	010010 1110
D03.6	110 00011	110001 0110	110001 0110	D03.7	111 00011	110001 1110	110001 0001
D04.6	110 00100	110101 0110	001010 0110	D04.7	111 00100	110101 0001	001010 1110
D05.6	110 00101	101001 0110	101001 0110	D05.7	111 00101	101001 1110	101001 0001
D06.6	110 00110	011001 0110	011001 0110	D06.7	111 00110	011001 1110	011001 0001
D07.6	110 00111	111000 0110	000111 0110	D07.7	111 00111	111000 1110	000111 0001
D08.6	110 01000	111001 0110	000110 0110	D08.7	111 01000	111001 0001	000110 1110
D09.6	110 01001	100101 0110	100101 0110	D09.7	111 01001	100101 1110	100101 0001
D10.6	110 01010	010101 0110	010101 0110	D10.7	111 01010	010101 1110	010101 0001
D11.6	110 01011	110100 0110	110100 0110	D11.7	111 01011	110100 1110	110100 1000
D12.6	110 01100	001101 0110	001101 0110	D12.7	111 01100	001101 1110	001101 0001
D13.6	110 01101	101100 0110	101100 0110	D13.7	111 01101	101100 1110	101100 1000
D14.6	110 01110	011100 0110	011100 0110	D14.7	111 01110	011100 1110	011100 1000

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 33 — Valid data characters (Sheet 3 of 3)

Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)	Data byte name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj (binary)	Current RD + abcdei fghj (binary)
D15.6	110 01111	010111 0110	101000 0110	D15.7	111 01111	010111 0001	101000 1110
D16.6	110 10000	011011 0110	100100 0110	D16.7	111 10000	011011 0001	100100 1110
D17.6	110 10001	100011 0110	100011 0110	D17.7	111 10001	100011 0111	100011 0001
D18.6	110 10010	010011 0110	010011 0110	D18.7	111 10010	010011 0111	010011 0001
D19.6	110 10011	110010 0110	110010 0110	D19.7	111 10011	110010 1110	110010 0001
D20.6	110 10100	001011 0110	001011 0110	D20.7	111 10100	001011 0111	001011 0001
D21.6	110 10101	101010 0110	101010 0110	D21.7	111 10101	101010 1110	101010 0001
D22.6	110 10110	011010 0110	011010 0110	D22.7	111 10110	011010 1110	011010 0001
D23.6	110 10111	111010 0110	000101 0110	D23.7	111 10111	111010 0001	000101 1110
D24.6	110 11000	110011 0110	001100 0110	D24.7	111 11000	110011 0001	001100 1110
D25.6	110 11001	100110 0110	100110 0110	D25.7	111 11001	100110 1110	100110 0001
D26.6	110 11010	010110 0110	010110 0110	D26.7	111 11010	010110 1110	010110 0001
D27.6	110 11011	110110 0110	001001 0110	D27.7	111 11011	110110 0001	001001 1110
D28.6	110 11100	001110 0110	001110 0110	D28.7	111 11100	001110 1110	001110 0001
D29.6	110 11101	101110 0110	010001 0110	D29.7	111 11101	101110 0001	010001 1110
D30.6	110 11110	011110 0110	100001 0110	D30.7	111 11110	011110 0001	100001 1110
D31.6	110 11111	101011 0110	010100 0110	D31.7	111 11111	101011 0001	010100 1110

Table 34 defines the valid special characters (Kxx.y characters). Comma patterns, two bits of one polarity followed by five bits of the opposite polarity, are underlined.

Table 34 — Valid special characters

Special code name	Bits HGF EDCBA (binary)	Current RD - abcdei fghj	Current RD + abcdei fghj
K28.0	000 11100	001111 0100	110000 1011
K28.1	001 11100	001111 1001	110000 0110
K28.2	010 11100	001111 0101	110000 1010
K28.3	011 11100	001111 0011	110000 1100
K28.4	100 11100	001111 0010	110000 1101
K28.5	101 11100	001111 1010	110000 0101
K28.6	110 11100	001111 0110	110000 1001
K28.7	111 11100	001111 1000	110000 0111
K23.7	111 10111	111010 1000	000101 0111
K27.7	111 11011	110110 1000	001001 0111
K29.7	111 11101	101110 1000	010001 0111
K30.7	111 11110	011110 1000	100001 0111

Only K28.3, K28.5 and K28.6 are used in this standard (see 7.2).

NOTE 7 - K28.1, K28.5, and K28.7 are the only valid characters which contain comma patterns. The K28.7 special character is not used because it introduces a false comma pattern when followed by any of the following special or data characters: K28.x, D3.x, D11.x, D12.x, D19.x, D20.x, or D28.x, where x is a value in the range 0 to 7, inclusive.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.3.3.2 Generating transmission characters

The appropriate entry in table 33 or table 34 shall be found for the data byte or special code for which a transmission character is to be generated (encoded). The current value of the transmitter's running disparity shall be used to select the transmission character from its corresponding column. For each transmission character transmitted, a new value of the running disparity shall be calculated. This new value shall be used as the transmitter's current running disparity for the next valid data byte or special code to be encoded and transmitted.

6.3.3.3 Validity of received transmission characters

The columns in table 33 and table 34 corresponding to the current value of the receiver's running disparity shall be searched for each received transmission character. If the received transmission character is found in the proper column, then the transmission character shall be considered valid and the associated data byte or special code determined (decoded). If the received transmission character is not found in the proper column, then the transmission character shall be considered invalid and the dword containing the character shall be considered an invalid dword. Independent of the transmission character's validity, the received transmission character shall be used to calculate a new value of running disparity.

This new value shall be used as the receiver's current running disparity for the next received transmission character.

Detection of a code violation does not necessarily indicate that the transmission character in which the code violation was detected is in error. Code violations may result from a prior error that altered the running disparity of the bit stream but did not result in a detectable error at the transmission character in which the error occurred. The example shown in table 35 exhibits this behavior.

Table 35 — 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 1011	+	010101 0101	+	111010 1010	+
Decoded character stream	-	D21.0	+	D10.2	+	Code violation	+

6.4 Bit order

Dwords transmitted in a STP connection shall be transmitted in the bit order specified by SATA.

--`,,,,,,-`-`,,,,,,`,,`-`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Dwords for other types of connections and outside of connections shall be transmitted in the bit order in figure 55.

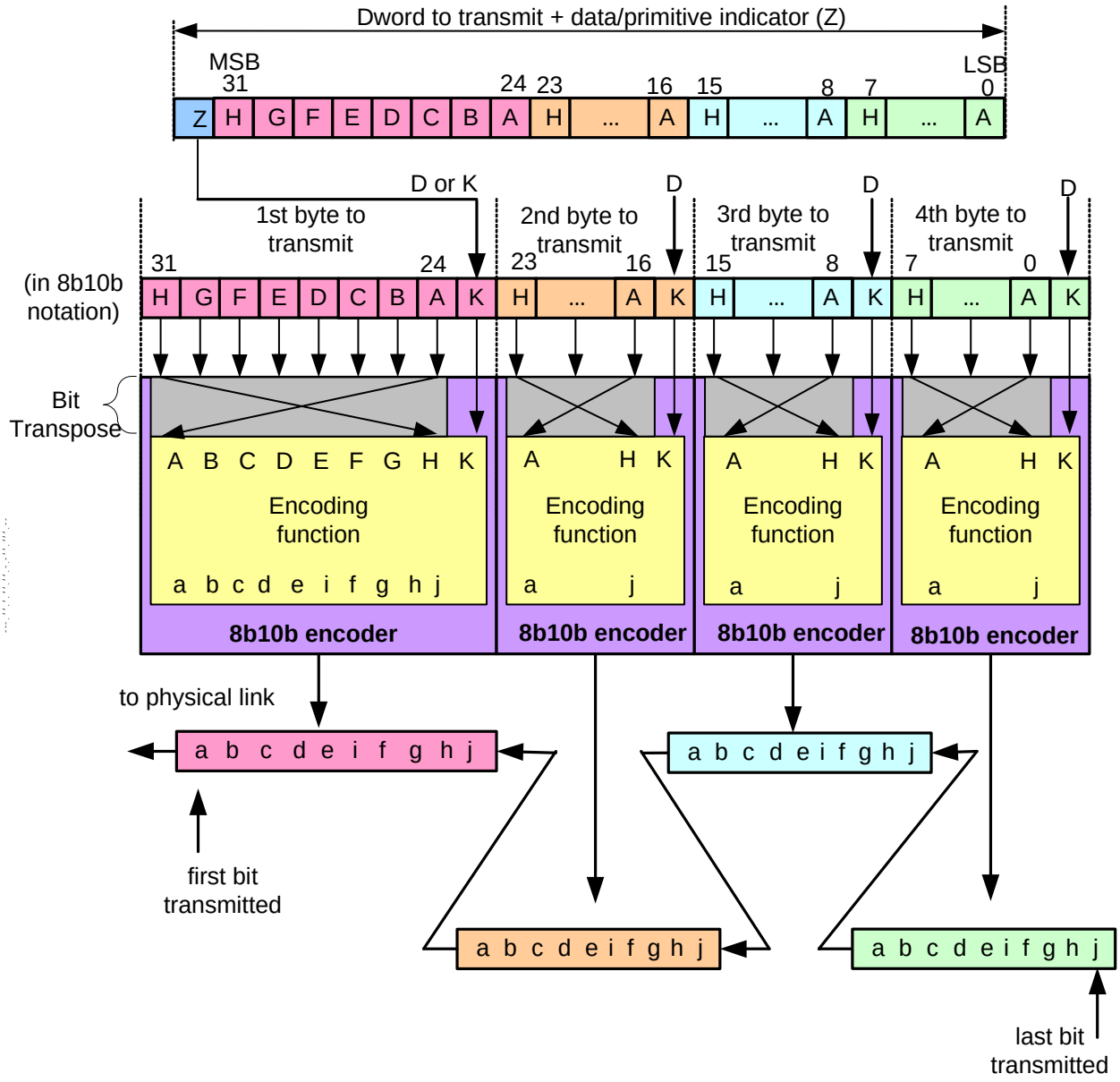


Figure 55 — SAS bit transmission logic

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 56 shows the SAS bit reception order.

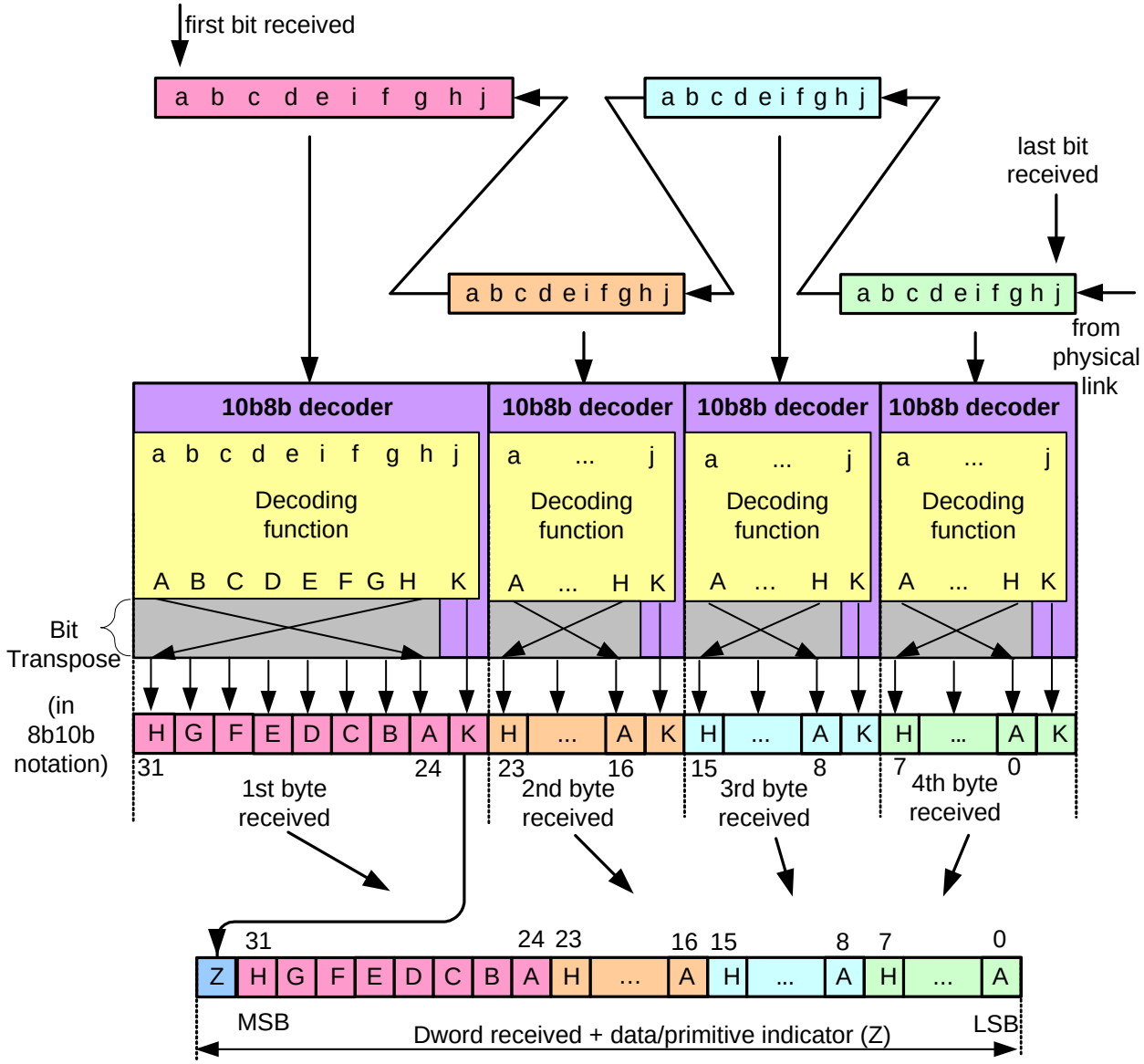


Figure 56 — SAS bit reception logic

6.5 Out of band (OOB) signals

Out of band (OOB) signals are low-speed signal patterns detected by the phy that do not appear in normal data streams. They consist of defined amounts of idle time followed by defined amounts of burst time. During the idle time, D.C. idle (see 3.1.21) is transmitted. During the burst time, ALIGN (0) primitives are transmitted repeatedly. The transmitter output levels during burst time and idle time are described in 5.3.4. The signals are differentiated by the length of idle time between the burst times.

SATA defines two OOB signals: COMINIT/COMRESET and COMWAKE. COMINIT and COMRESET are used in this standard interchangeably. Phys compliant with this standard identify themselves with an additional SAS-specific OOB signal called COMSAS.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 36 defines the timing specifications for OOB signals.

Table 36 — OOB signal timing specifications

Parameter	Minimum	Nominal	Maximum	Comments
OOB Interval (OOBI) [a]	666,600 ps	666,$\overline{6}$ ps	666,733 ps	The time basis for burst times and idle times used to create OOB signals. Based on 1,5 Gbps clock tolerance.
COMSAS detect timeout	13,65 μs			The minimum time a receiver shall allow to detect COMSAS after transmitting COMSAS. Derived from: OOBI x 512 x 40

[a] OOBI is different than UI(OOB) defined in SATA (e.g., SAS has tighter clock tolerance). This is a fixed value equal to the UI for G1, regardless of the actual transfer rate being used to create the burst time.

Table 37 describes the OOB signal transmitter requirements for the burst time, idle time, and negation times that comprise each OOB signal.

Table 37 — OOB signal transmitter requirements

Signal	Burst time	Idle time	Negation time
COMWAKE	160 OOBI	160 OOBI	280 OOBI
COMINIT/RESET	160 OOBI	480 OOBI	800 OOBI
COMSAS	160 OOBI	1 440 OOBI	2 400 OOBI

To transmit an OOB signal, a transmitter shall repeat these steps six times:

1) transmit D.C. idle for an idle time; and
2) transmit an ALIGN burst for a burst time.

It shall then transmit D.C. idle for an OOB signal negation time.

The ALIGNs used in OOB signals are not required to be at generation 1 (G1) physical link rates (i.e., 1,5 Gbps), as this rate may not be supported in phys compliant with future generations of this standard. The ALIGNs are only required to generate an envelope for the detection circuitry, as required for any signaling that may be A.C. coupled. If G2 ALIGNs are used, the number of ALIGNs doubles compared with G1 ALIGNs.

A SAS transmitter should transmit ALIGNs at the G1 physical link rate to create the burst portion of the OOB signal, but may transmit ALIGNs at its slowest supported physical link rate if it does not support the G1 physical link rate and shall not transmit them at a physical link rate faster than its slowest supported physical link rate.

--`,,,`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 57 describes OOB signal transmission by the SP transmitter (see 6.7). The COMWAKE Transmitted, COMINIT Transmitted, and COMSAS Transmitted messages are sent to the SP state machine (see 6.7).

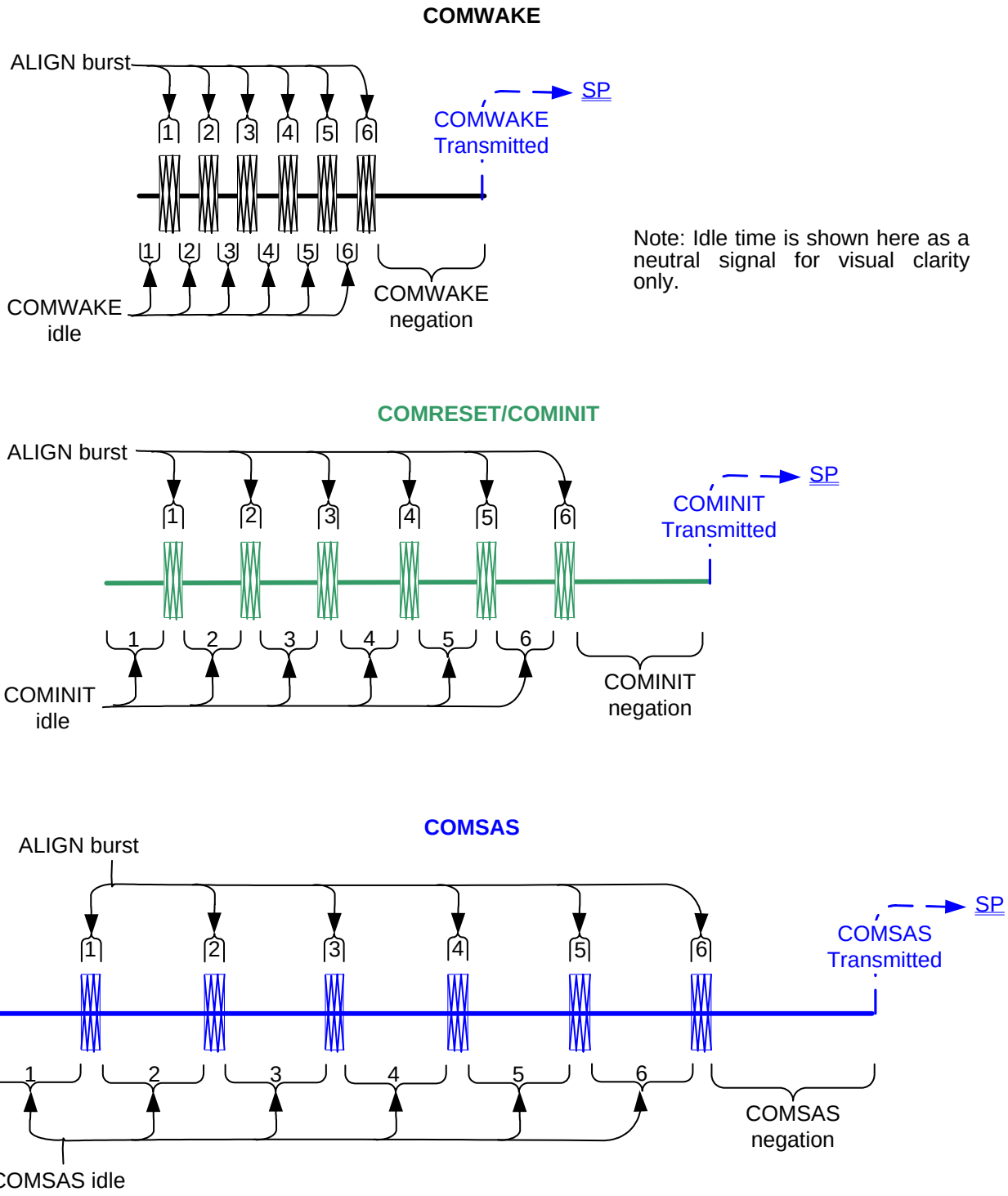


Figure 57 — OOB signal transmission

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 38 describes the OOB signal receiver requirements for detecting burst times, assuming T_{burst} is the length of the detected burst time. The burst time is not used to distinguish between signals.

Table 38 — OOB signal receiver burst time detection requirements

Signal	may detect	shall detect
COMWAKE	$T_{burst} \leq 100$ ns	$T_{burst} > 100$ ns
COMINIT/COMRESET	$T_{burst} \leq 100$ ns	$T_{burst} > 100$ ns
COMSAS	$T_{burst} \leq 100$ ns	$T_{burst} > 100$ ns

Table 39 describes the OOB signal receiver requirements for detecting idle times, assuming T_{idle} is the length of the detected idle time.

Table 39 — OOB signal receiver idle time detection requirements

Signal	may detect	shall detect	shall not detect
COMWAKE	55 ns $\leq T_{idle} <$ 175 ns	101,3 ns $\leq T_{idle} \leq$ 112 ns	$T_{idle} <$ 55 ns or $T_{idle} \geq$ 175 ns
COMINIT/ COMRESET	175 ns $\leq T_{idle} <$ 525 ns	304 ns $\leq T_{idle} \leq$ 336 ns	$T_{idle} <$ 175 ns or $T_{idle} \geq$ 525 ns
COMSAS	525 ns $\leq T_{idle} <$ 1 575 ns	911,7 ns $\leq T_{idle} \leq$ 1 008 ns	$T_{idle} <$ 525 ns or $T_{idle} \geq$ 1 575 ns

Table 40 describes the OOB signal receiver requirements for detecting negation times, assuming T_{idle} is the length of the detected idle time.

Table 40 — OOB signal receiver negation time detection requirements

Signal	shall detect
COMWAKE	$T_{idle} >$ 175 ns
COMINIT/COMRESET	$T_{idle} >$ 525 ns
COMSAS	$T_{idle} >$ 1 575 ns

A receiver shall detect an OOB signal after receiving four consecutive idle time/burst time pairs (see figure 58). It is not an error to receive more than four idle time/burst time pairs. A receiver shall not detect the same OOB signal again until it has detected the corresponding negation time (i.e., a COMINIT negation time for a COMINIT) or has detected a different OOB signal (e.g., if a COMINIT was previously detected, then four sets of COMWAKE idle times followed by burst times are detected, a COMWAKE is detected; another COMINIT may follow).

A SAS receiver shall detect OOB signals comprised of ALIGNs transmitted at any rate up to its highest supported physical link rate. This includes physical link rates below its lowest supported physical link rate (e.g., a SAS receiver supporting only 3,0 Gbps needs to detect 1,5 Gbps based ALIGNs to interoperate with a SAS transmitter supporting both 1,5 Gbps and 3,0 Gbps).

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 58 describes SAS OOB signal detection by the SP receiver (see 6.7). The COMWAKE Detected, COMWAKE Completed, COMINIT Detected, COMSAS Detected, and COMSAS Completed messages are sent to the SP state machine (see 6.7) to indicate that an OOB signal has been partially or fully detected.

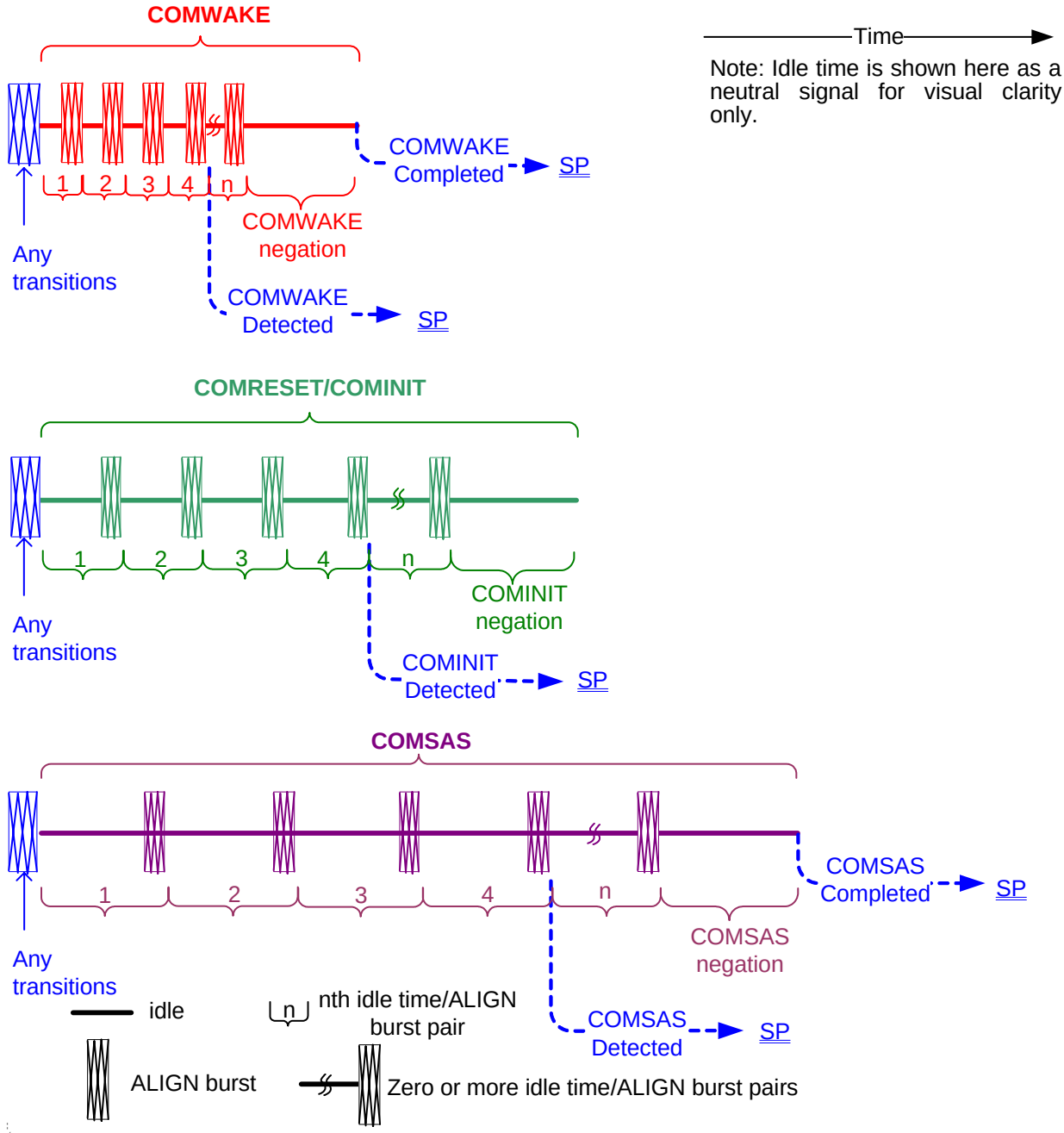


Figure 58 — OOB signal detection

Expander devices shall not forward OOB signals. An expander device shall run the link reset sequence independently on each physical link.

6.6 Phy reset sequences

6.6.1 Phy reset sequences overview

The phy reset sequence consists of an OOB sequence and a speed negotiation sequence.

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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A phy shall originate a phy reset sequence after:

a) power on;
b) hard reset (i.e., receiving a HARD_RESET) (see 4.4.2);
c) management application layer request (see 6.7.1);
d) losing dword synchronization (see 6.7.4.9); and
e) for expander phys, after a hot-plug timeout (see 6.6.5).

A SAS phy may originate a phy reset sequence after a hot-plug timeout (see 6.6.5).

After receiving a HARD_RESET, a phy should start the phy reset sequence within 250 ms.

Phys shall not originate a phy reset sequence until 10 ms have elapsed since the previous attempt at running a phy reset sequence (e.g., if a reply to COMINIT is not detected in an OOB sequence, or after a speed negotiation sequence fails).

Table 41 defines phy reset sequence timing parameters used by the SP state machine (see 6.7).

Table 41 — Phy reset sequence timing specifications

Parameter	Time	Comments
Hot-plug timeout	500 ms	The maximum time after which an expander phy shall retry an unsuccessful phy reset sequence (see 6.6.5).

6.6.2 SATA phy reset sequence

6.6.2.1 SATA OOB sequence

Figure 59 shows the SATA OOB sequence between a SATA host and SATA device. The SATA OOB sequence is defined by SATA; see ATA/ATAPI-7 V3 for detailed requirements.

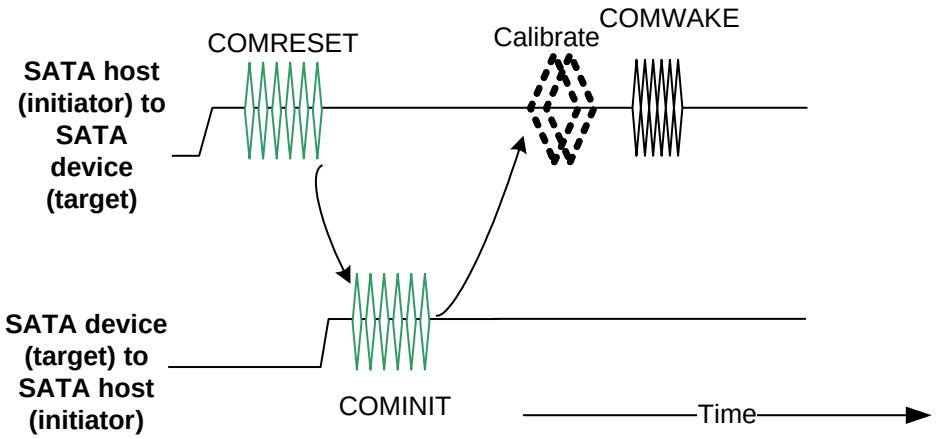


Figure 59 — SATA OOB sequence

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.6.2.2 SATA speed negotiation sequence

Figure 60 shows the speed negotiation sequence between a SATA host and SATA device. The SATA speed negotiation sequence is defined by SATA; see ATA/ATAPI-7 V3 for detailed requirements.

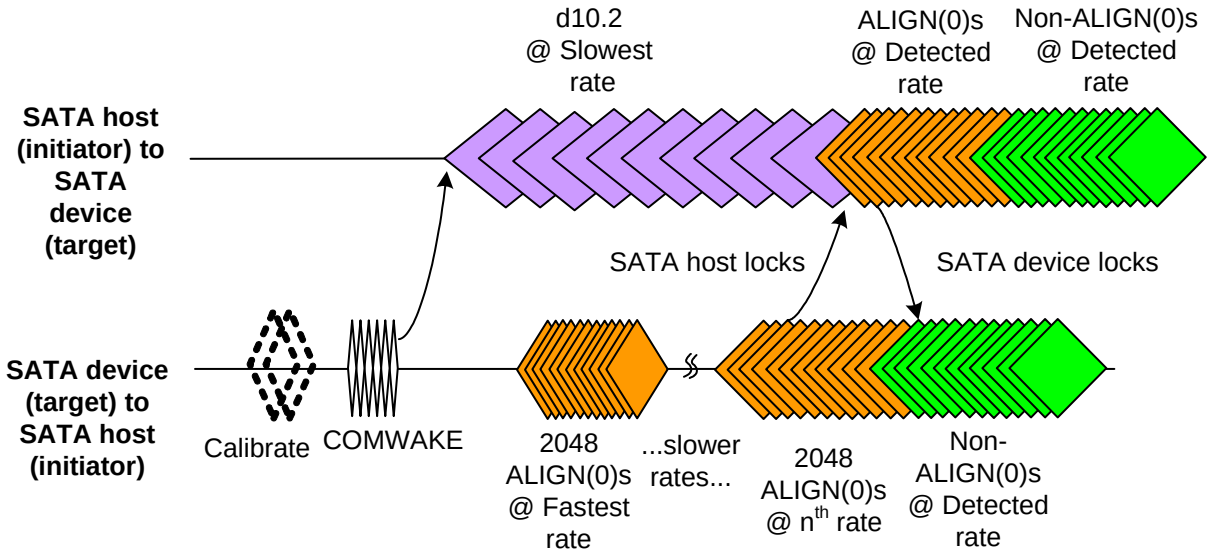


Figure 60 — SATA speed negotiation sequence

Table 42 defines SATA speed negotiation sequence timing parameters used by the SP state machine (see 6.7).

Table 42 — SATA speed negotiation sequence timing specifications

Parameter	Time	Comments
Await ALIGN timeout	1 310 720 OOBI	The minimum time during SATA speed negotiation that a phy shall allow for an ALIGN to be received after detecting COMWAKE Completed.
COMWAKE response time	533 ns	The maximum time during SATA speed negotiation after detecting COMWAKE Completed before which a phy shall start transmitting D10.2 characters.

6.6.3 SAS to SATA phy reset sequence

SAS initiator devices and expander devices may be directly attached to SATA devices.

To initiate a phy reset sequence a 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 it 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 phy that a SATA phy is attached. As a result the phy shall initiate transmit COMWAKE and enter the SATA speed negotiation sequence.

Figure 61 shows a reset 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

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

COMRESET and responds with a COMINIT. The SP state machine treats these two cases the same, and determines that a SATA phy is attached after a COMSAS detect timeout. The SATA speed negotiation sequence shall be entered after COMWAKE.

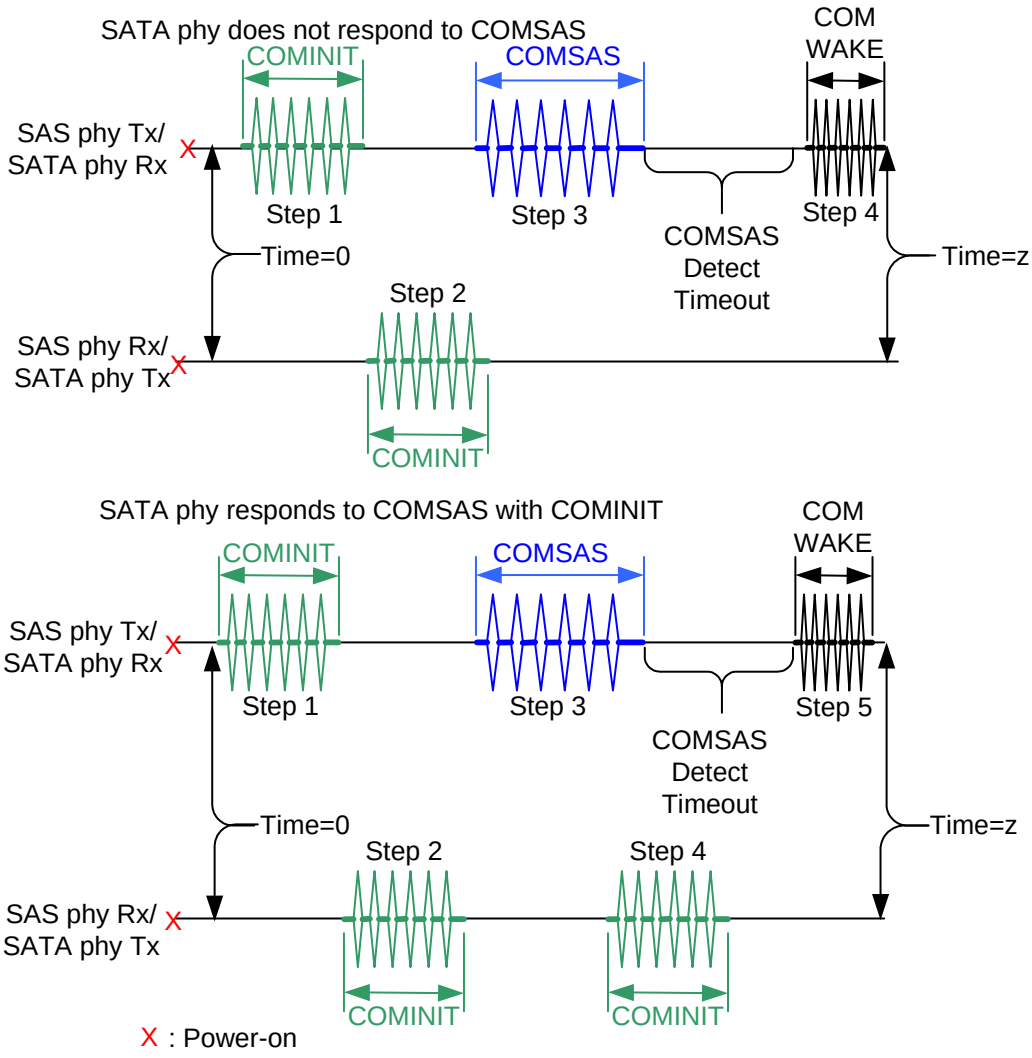


Figure 61 — SAS to SATA OOB sequence

6.6.4 SAS to SAS phy reset sequence

6.6.4.1 SAS OOB sequence

To initiate a SAS OOB sequence a phy shall transmit a COMINIT.

On receipt of a COMINIT a phy shall either:

a) if the receiving phy has not yet transmitted a COMINIT, transmit a COMINIT followed by a COMSAS; or
b) if the receiving phy has transmitted a COMINIT, transmit a COMSAS.

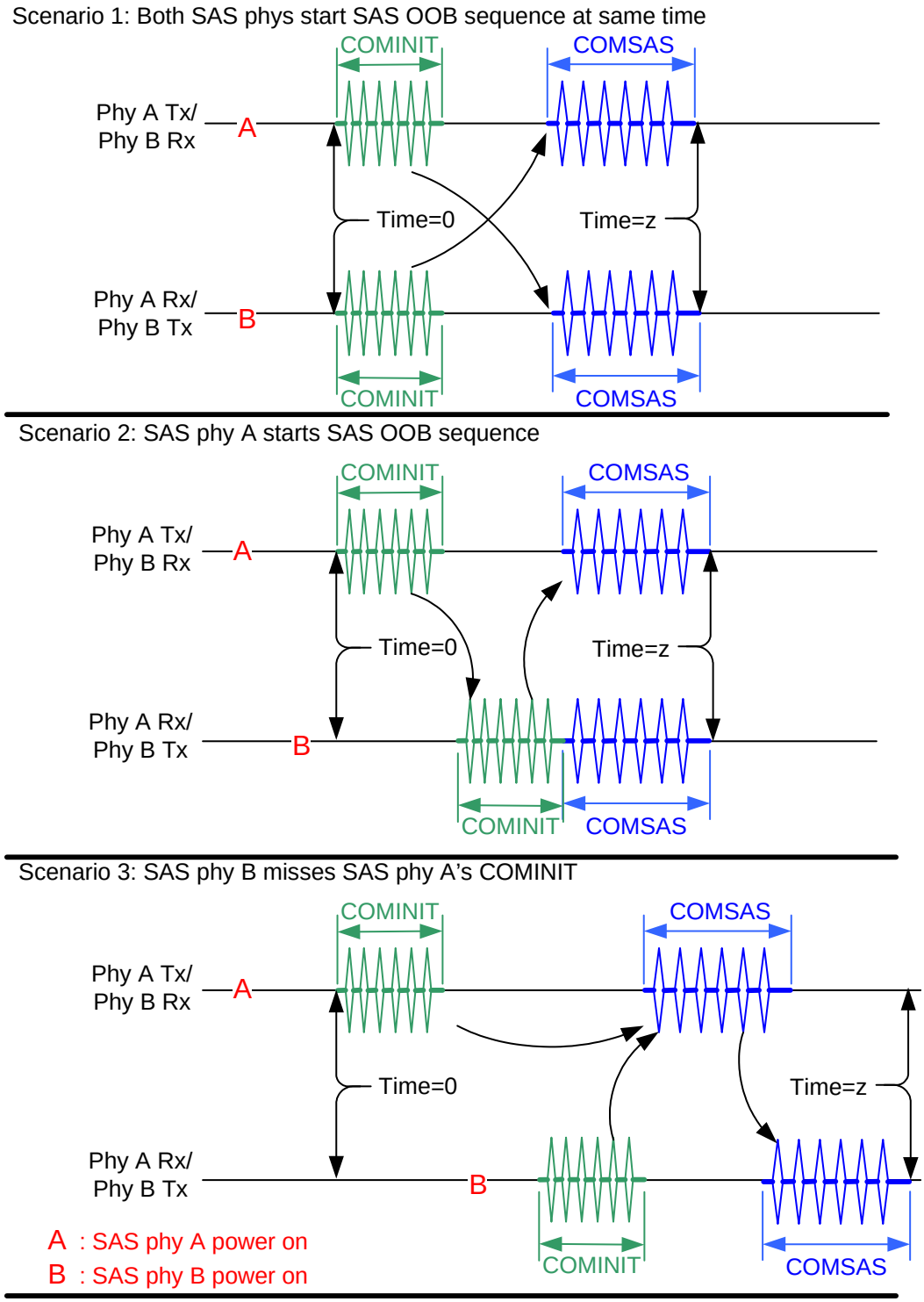
On receipt of a COMSAS, if the receiving phy has not yet transmitted a COMSAS, the phy shall transmit a COMSAS.

After completing the transmission of a COMSAS and the successful receipt a COMSAS the SAS OOB sequence is complete and the SAS speed negotiation sequence begins.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A phy shall distinguish between COMINIT and COMSAS and continue with a SAS speed negotiation sequence after completing the SAS OOB sequence.

Figure 62 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.



Time 0: SAS phy reset sequence begins

Time z: SAS speed negotiation sequence begins

Figure 62 — SAS to SAS OOB sequence

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

6.6.4.2 SAS speed negotiation sequence

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 sequence consists of a set of speed negotiation windows for each physical link rate, starting with 1,5 Gbps, then 3,0 Gbps, then the next rate. The length of the speed negotiation sequence is determined by the number of physical link rates supported by the phys.

Figure 63 defines the speed negotiation window, including:

a) speed negotiation window time;
b) rate change delay time (RCDT);
c) speed negotiation transmit time (SNTT); and
d) speed negotiation lock time (SNLT).

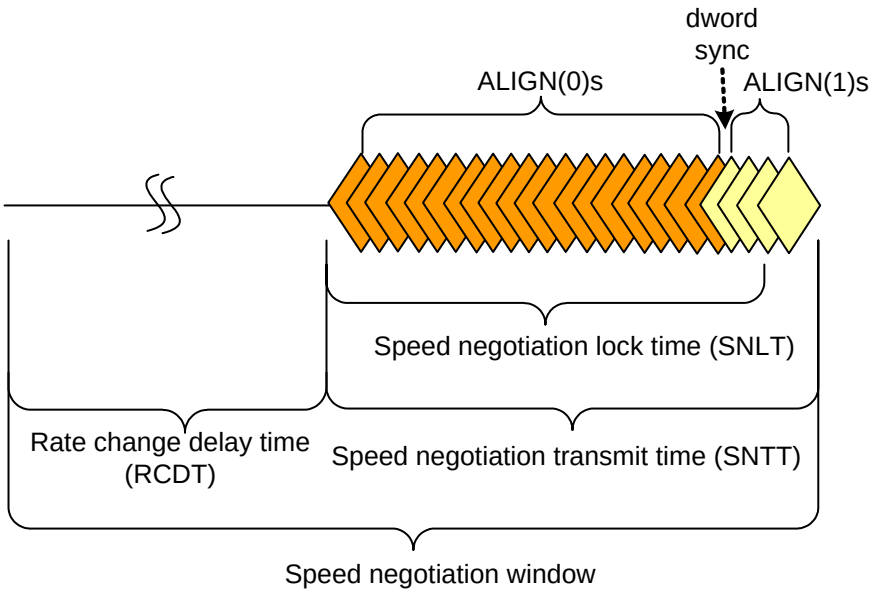


Figure 63 — SAS speed negotiation window

Table 43 defines the timing specifications for the SAS speed negotiation sequence.

Table 43 — SAS speed negotiation sequence timing specifications

Parameter	Time	Comments
Rate change delay time (RCDT)	750 000 OOBI	The time the transmitter shall transmit D.C. idle between rates during speed negotiation.
Speed negotiation transmit time (SNTT)	163 840 OOBI	The time during which ALIGN (0) or ALIGN (1) is transmitted at each physical link rate during the speed negotiation sequence. Derived from: OOBI x 4 096 x 40.
Speed negotiation lock time (SNLT)	153 600 OOBI	The maximum time during the speed negotiation window for a transmitter to reply with ALIGN (1). Derived from: OOBI x 3 840 x 40
Speed negotiation window time	913 840 OOBI	The duration of a speed negotiation window. Derived from: RCDT + SNTT.

The speed negotiation window shall consist of the following transmission sequence for each speed negotiation window:

1) a transmission of D.C. idle for an RCDT; and

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

2) if the phy supports the physical link rate, a transmission of ALIGNs at that physical link rate for the remainder of the entire speed negotiation window. If the phy does not support the physical link rate, transmission of D.C. idle for the remainder of the entire speed negotiation window.

If the phy supports the physical link rate, it shall attempt to synchronize on an incoming series of dwords at that rate for the SNLT. The received dwords may be ALIGN (0) or ALIGN (1) primitives. If the phy achieves dword synchronization within the SNLT, 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 speed negotiation window). If the phy does not achieve dword synchronization within the SNLT, it shall continue transmitting ALIGN(0)s for the remainder of the SNTT (i.e., the remainder of the speed negotiation window).

At the end of the SNTT, if a phy is both transmitting and receiving ALIGN (1) primitives, it shall consider that physical link rate valid. The phy shall then proceed to the next speed negotiation window. A phy shall participate in all speed negotiation windows:

a) up to its highest supported physical link rate plus one; or
b) until it runs a speed negotiation window that does not detect a valid physical link rate after having detected a valid physical link rate in a previous speed negotiation window.

If the phy has detected a valid physical link rate in the previous speed negotiation window, it shall enter the final speed negotiation window using the highest previously successful link rate.

Figure 64 shows speed negotiation between a phy A that supports G1 through G3 link rates and a phy B that only supports the G2 link rate. Both phys run:

1) the G1 speed negotiation window, supported by phy A but not by phy B;
2) the G2 speed negotiation window, supported by both phys; and
3) the G3 speed negotiation window, supported by phy A but not by phy B.

Both phys then select G2 for the final speed negotiation window to establish the negotiated physical link rate.

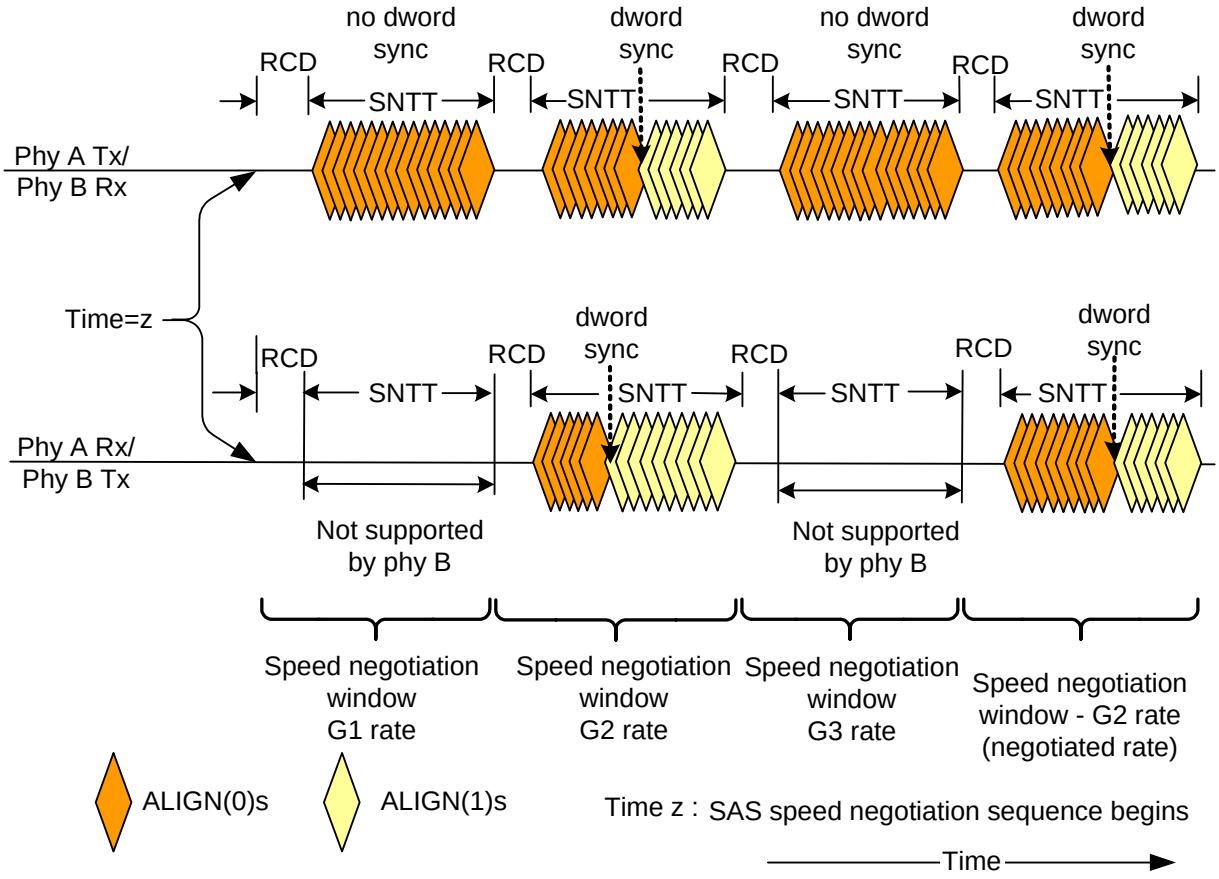


Figure 64 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G2 only)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

If the phy does not obtain dword synchronization during the final speed negotiation window the SAS speed negotiation sequence fails. This may be counted and reported in the PHY RESET PROBLEM field in the SMP REPORT PHY ERROR LOG page (see 10.4.3.6) and the REPORT PHY ERROR LOG log page (see 10.2.7.1).

Figure 65 shows the same speed negotiation sequence as in figure 64 when phy B does not obtain dword synchronization during the final speed negotiation window. If this occurs, the handshake is not complete and the OOB sequence shall be retried starting with COMINIT, forcing the phy to retry the whole reset sequence.

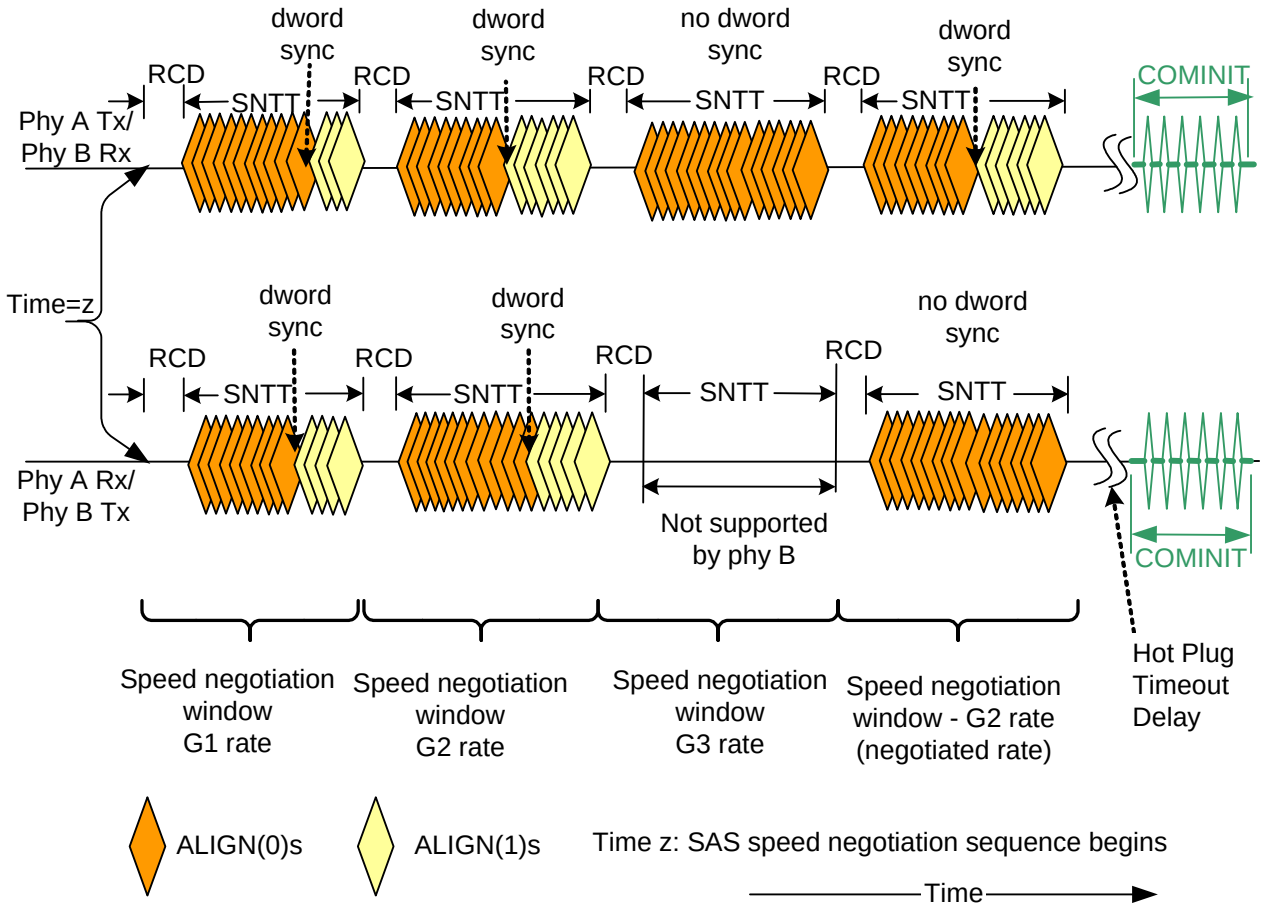


Figure 65 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G1, G2) that fails

For more examples of speed negotiations between phys that support various speeds, see Annex B.

6.6.5 Phy reset sequence after devices are attached

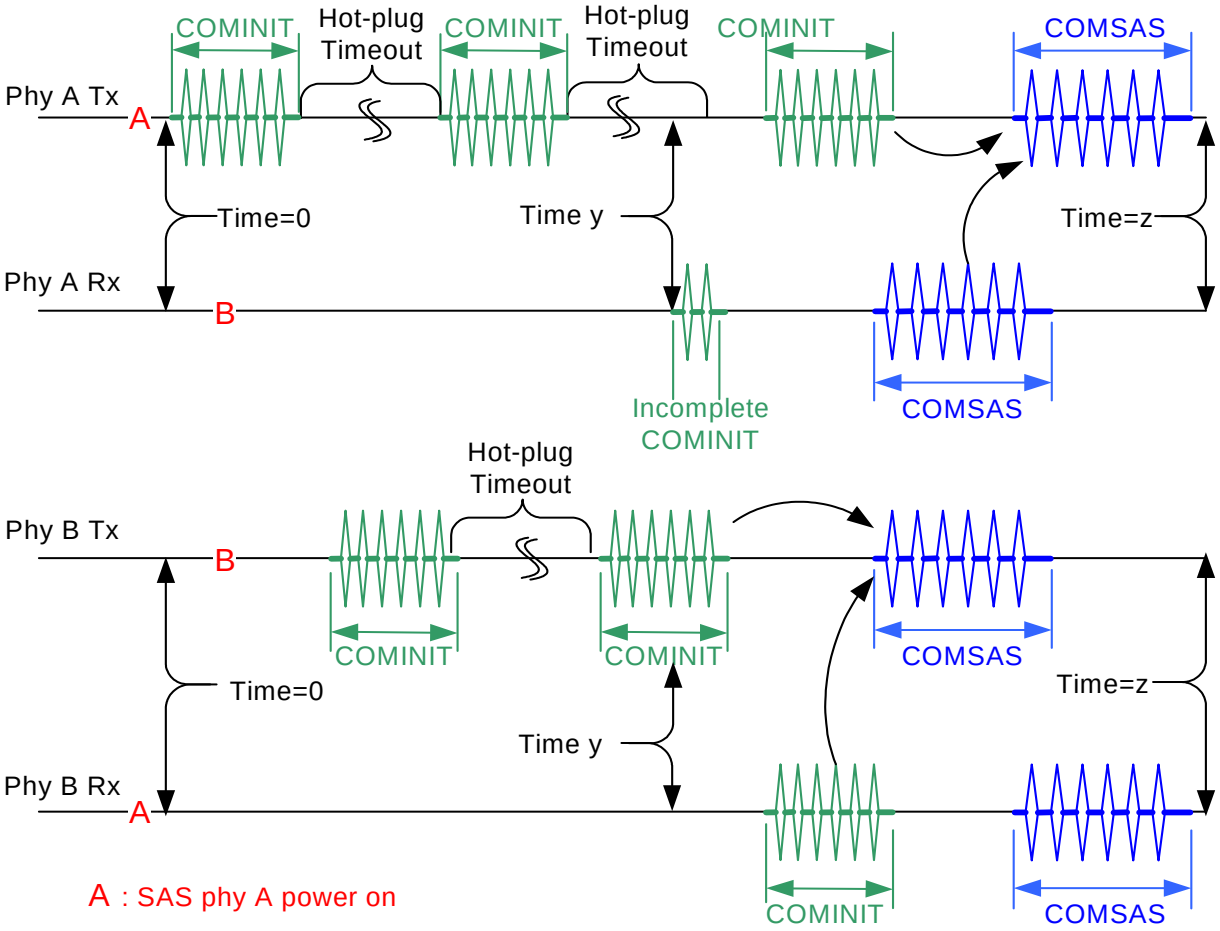
Since SATA and SAS signal cable connectors do not include power lines, it is not possible to detect the physical insertion of the signal cable connector onto a plug. Non-cabled environments may similarly not have a way to detect physical insertion of a device. As a result, every time a phy reset sequence 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 41) between each attempt, until a speed negotiation sequence completes successfully;
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 66 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,

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

since it has both transmitted and received a COMINIT. Upon receiving COMSAS, phy A transmits its own COMSAS. The SAS speed negotiation sequence follows.



A : SAS phy A power on

B : SAS phy B power on

Time y : SAS phy A attached to SAS phy B

Time z : SAS phy A and SAS phy B start the SAS speed negotiation sequence

Figure 66 — Hot-plug and the phy reset sequence

6.7 SP (phy layer) state machine

6.7.1 SP state machine overview

The SP state machine controls the phy reset sequence. This state machine consists of three sets of states:

a) OOB sequence (OOB) states;
b) SAS speed negotiation (SAS) states; and
c) SATA host emulation (SATA) states.

This state machine consists of the following states:

a) SP0:OOB_COMINIT (see 6.7.3.2)(initial state);
b) SP1:OOB_AwaitCOMX (see 6.7.3.3);
c) SP2:OOB_NoCOMSASTimeout;
d) SP3:OOB_AwaitCOMINIT_Sent (see 6.7.3.5);
e) SP4:OOB_COMSAS (see 6.7.3.6);
f) SP5:OOB_AwaitCOMSAS_Sent (see 6.7.3.7);
g) SP6:OOB_AwaitNoCOMSAS (see 6.7.3.8);
h) SP7:OOB_AwaitCOMSAS (see 6.7.3.9);
i) SP8:SAS_Start (see 6.7.4.2);

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

j) SP9:SAS_RateNotSupported (see 6.7.4.3);
k) SP10:SAS_AwaitALIGN (see 6.7.4.4);
l) SP11:SAS_AwaitALIGN1 (see 6.7.4.5);
m) SP12:SAS_AwaitSNW (see 6.7.4.6);
n) SP13:SAS_Pass (see 6.7.4.7);
o) SP14 SAS_Fail (see 6.7.4.8);
p) SP15:SAS_PHY_Ready (see 6.7.4.9);
q) SP16:SATA_COMWAKE (see 6.7.5.2);
r) SP17:SATA_AwaitCOMWAKE (see 6.7.5.3);
s) SP18:SATA_AwaitNoCOMWAKE (see 6.7.5.4);
t) SP19:SATA_AwaitALIGN (see 6.7.5.5);
u) SP20:SATA_AdjustSpeed (see 6.7.5.6);
v) SP21:SATA_Transmit_ALIGN (see 6.7.5.7);
w) SP22:SATA_PHY_Ready (see 6.7.5.8);
x) SP23:SATA_PM_Partial (see 6.7.5.9); and
y) SP24:SATA_PM_Slumber (see 6.7.5.10).

The SP state machine shall start in the SP0:OOB_COMINIT state after:

a) a power on;
b) a hard reset; or
c) receiving a Management Reset request from the management layer (e.g., from the SMP PHY CONTROL function in an expander device). Receipt of a COMINIT in any state that does not have an exit transition triggered by receipt of COMINIT should cause a Management Reset request.

The SP state machine sends the following messages to the SP_DWS state machine (see 6.8):

a) Start DWS; and
b) Stop DWS.

The SP state machine receives the following messages from the SP_DWS state machine:

a) DWS Lost; and
b) DWS Reset.

The SP state machine shall maintain the timers listed in table 44.

Table 44 — SP state machine timers

Timer	Initial value
COMSAS Detect Timeout timer	COMSAS detect timeout (see table 36)
Await ALIGN Timeout timer	Await ALIGN timeout (see table 42)
Hot-Plug Timeout timer	Hot plug timeout (see 6.6.1)
RCDT timer	RCDT (see table 43)
SNLT timer	SNLT (see table 43)
SNTT timer	SNTT (see table 43)

6.7.2 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 6.7).

The SP transmitter receives the following messages from the SP state machine:

a) Transmit COMINIT;
b) Transmit COMSAS;
c) Transmit COMWAKE;
d) Transmit D10.2;
e) Set Rate (Physical Link Rate); and

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

f) Transmit ALIGN with an argument indicating the specific type (e.g., Transmit ALIGN (0)).

When not otherwise instructed, the SP transmitter transmits idle time.

The SP transmitter sends the following messages to the SP state machine:

a) COMINIT Transmitted;
b) COMSAS Transmitted; and
c) COMWAKE Transmitted.

The SP receiver receives OOB signals and dwords from the physical link and sends messages to the SP state machine indicating what it has received.

The SP receiver sends the following messages to the SP state machine:

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)); and
g) Dword Received.

The ALIGN Received and Dword Received messages are only sent when 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)s were detected. For SAS speed negotiation, only ALIGNs at the physical link rate specified by the last Set Rate message received by the SP transmitter cause ALIGN Received messages.

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.3 OOB sequence states

6.7.3.1 OOB sequence states overview

Figure 67 shows the OOB sequence states. These states are indicated by state names with a prefix of OOB.

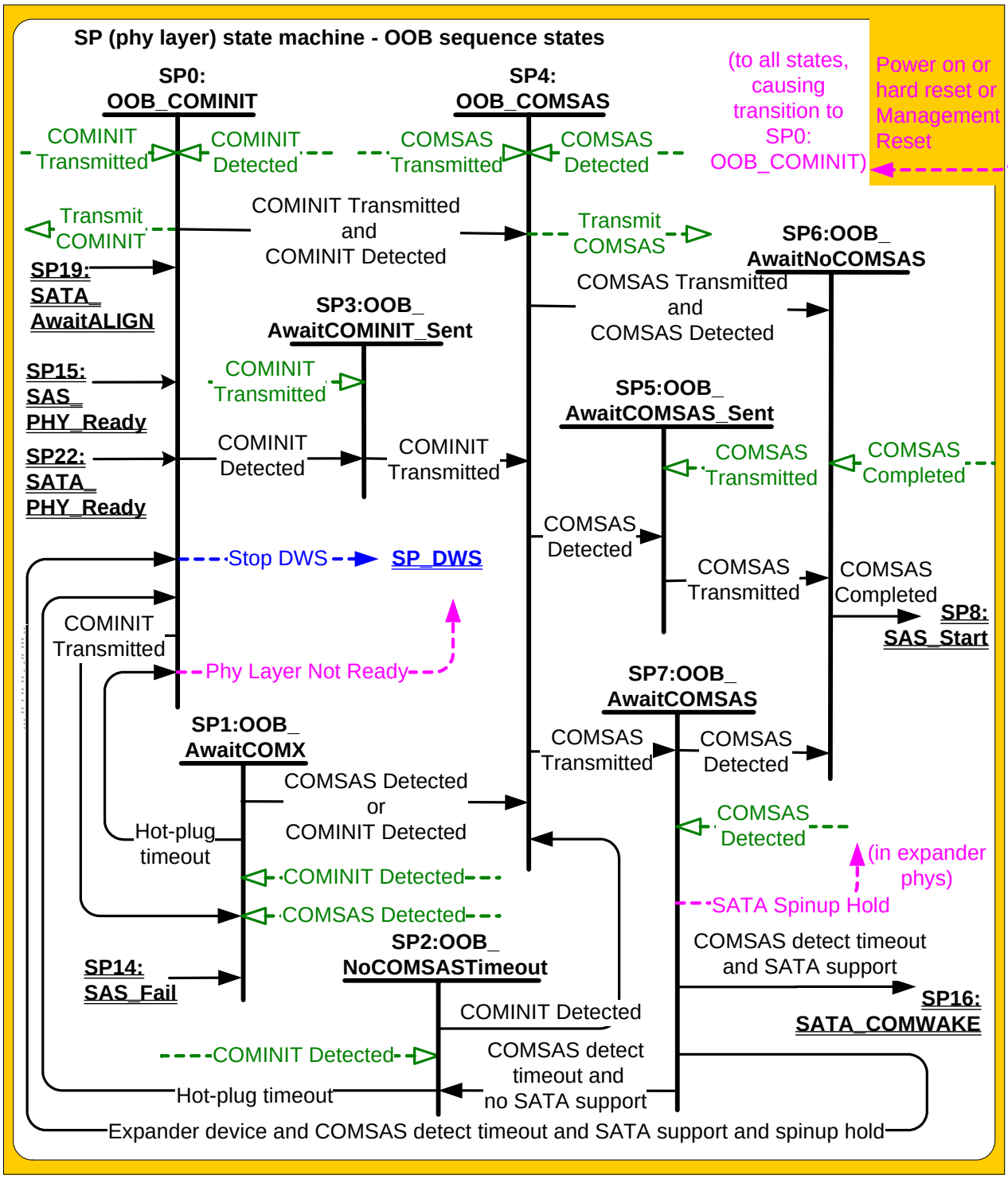


Figure 67 — SP (phy layer) state machine - OOB sequence states

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.7.3.2 SP0:OOB_COMINIT state

6.7.3.2.1 State description

This state is the initial state for this state machine.

Upon entry into this state, this state shall:

a) send a Transmit COMINIT message to the SP transmitter;
b) send a Stop DWS message to the SP_DWS state machine; and
c) send a Phy Layer Not Ready confirmation to the link layer.

This state machine waits for receipt of a COMINIT Transmitted message and/or a COMINIT Detected message.

6.7.3.2.2 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.

6.7.3.2.3 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.

6.7.3.2.4 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.

6.7.3.3 SP1:OOB_AwaitCOMX state

6.7.3.3.1 State description

Upon entry into this state, the Hot-Plug Timeout timer shall be initialized and started if this phy is:

a) an expander phy; or
b) an initiator phy or target phy implementing the Hot-Plug Timeout timer.

6.7.3.3.2 Transition SP1:OOB_AwaitCOMX to SP0:OOB_COMINIT

This transition shall occur if the Hot-Plug Timeout timer expires.

6.7.3.3.3 Transition SP1:OOB_AwaitCOMX to SP4:OOB_COMSAS

This transition shall occur after receiving either a COMINIT Detected message or a COMSAS Detected message. If COMSAS Detected was received, this transition shall include a COMSAS Detected argument.

6.7.3.4 SP2:OOB_NoCOMSASTimeout state

6.7.3.4.1 State description

Upon entry into this state, the Hot-Plug Timeout timer shall be initialized and started 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.

6.7.3.4.2 Transition SP2:OOB_NoCOMSASTimeout to SP0:OOB_COMINIT

This transition shall occur if the Hot-Plug Timeout timer expires.

6.7.3.4.3 Transition SP2:OOB_NoCOMSASTimeout to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Detected message.

--`,,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.3.5 SP3:OOB_AwaitCOMINIT_Sent state

6.7.3.5.1 State description

This state waits for a COMINIT Transmitted message.

6.7.3.5.2 Transition SP3:OOB_AwaitCOMINIT_Sent to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Transmitted message.

6.7.3.6 SP4:OOB_COMSAS state

6.7.3.6.1 State description

Upon entry into this state, this state 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.

6.7.3.6.2 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.

6.7.3.6.3 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.

6.7.3.6.4 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.

6.7.3.7 SP5:OOB_AwaitCOMSAS_Sent state

6.7.3.7.1 State description

This state waits for receipt of a COMSAS Transmitted message.

6.7.3.7.2 Transition SP5:OOB_AwaitCOMSAS_Sent to SP6:OOB_AwaitNoCOMSAS

This transition shall occur after receiving a COMSAS Transmitted message.

6.7.3.8 SP6:OOB_AwaitNoCOMSAS state

6.7.3.8.1 State description

This state machine waits for a COMSAS Completed message, which indicates that COMSAS has been completely received.

6.7.3.8.2 Transition SP6:OOB_AwaitNoCOMSAS to SP8:SAS_Start

This transition shall occur after receiving a COMSAS Completed message. The COMSAS Completed message may be received before this state is entered.

6.7.3.9 SP7:OOB_AwaitCOMSAS state

6.7.3.9.1 State description

Upon entering this state the COMSAS Detect Timeout timer shall be initialized and started.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.3.9.2 Transition SP7:OOB_AwaitCOMSAS to SP0:OOB_COMINIT

This state shall send a SATA Spinup Hold confirmation to the link layer and perform this transition if.

a) this phy is in an expander device;
b) this phy supports attachment to a SATA device;
c) the COMSAS Detect Timeout timer expires;
d) this expander device implements SATA spinup hold; and
e) the SP0:OOB_COMINIT state was not originally entered because of an SMP Reset request (i.e., SMP PHY CONTROL-based requests to reset the phy bypass spinup hold).

6.7.3.9.3 Transition SP7:OOB_AwaitCOMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur after receiving a COMSAS Detected message.

6.7.3.9.4 Transition SP7:OOB_AwaitCOMSAS to SP16:SATA_COMWAKE

This transition shall occur if:

a) the phy supports attachment to SATA devices; and
b) the COMSAS Detect Timeout timer expires.

6.7.3.9.5 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.

6.7.4 SAS speed negotiation states

6.7.4.1 SAS speed negotiation states overview

Figure 68 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

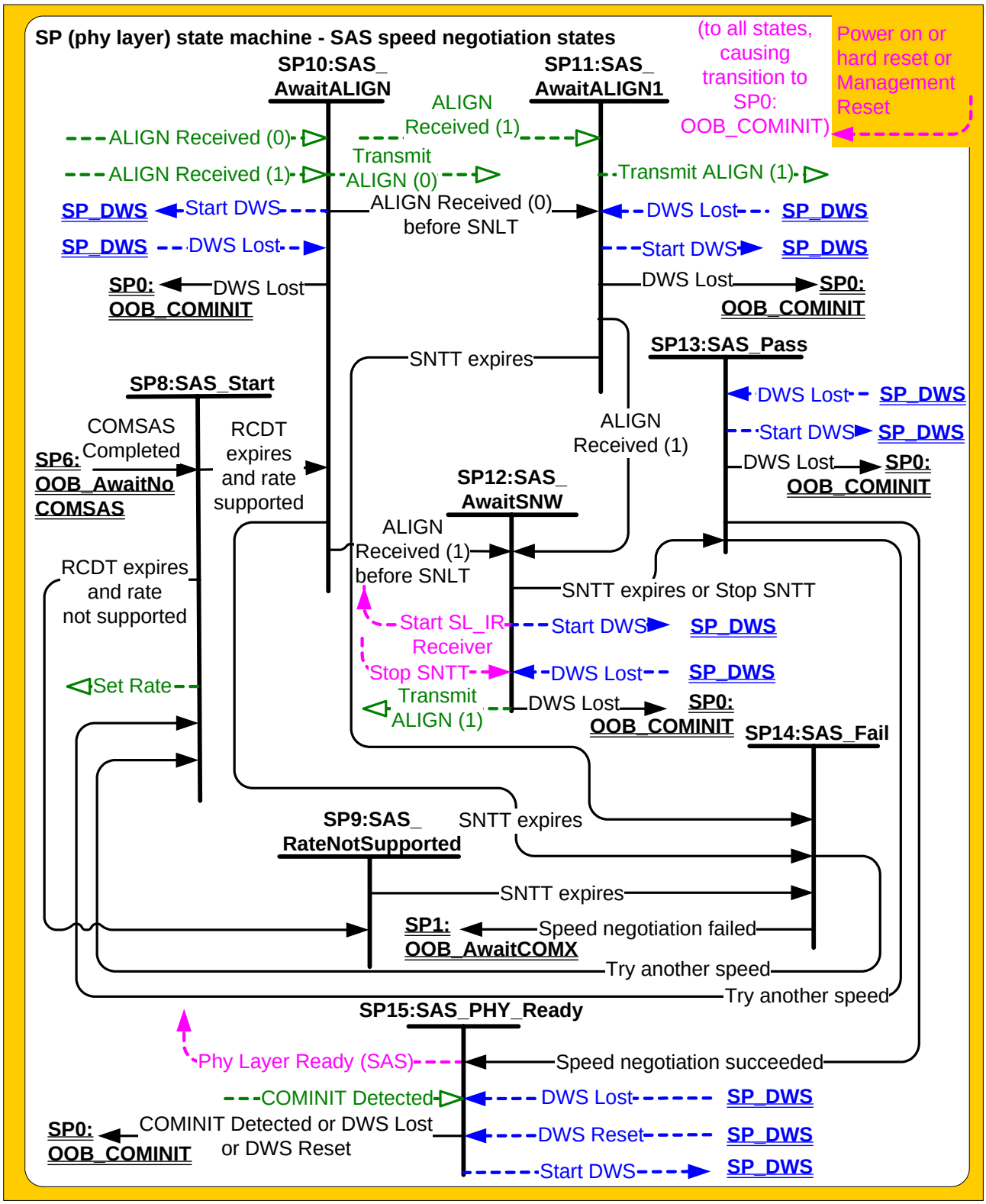


Figure 68 — SP (phy layer) state machine - SAS speed negotiation states

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.7.4.2 SP8:SAS_Start state

6.7.4.2.1 State description

This is the initial state for the SAS speed negotiation sequence.

Upon entering this state, this state shall:

a) initialize and start the RCDT timer. This allows time required for a transmitter to switch to either the next higher or next lower supported speed;
b) send a Set Rate message to the SP transmitter selecting the next rate for attempted speed negotiation.

The argument to the Set Rate message shall be:

a) 1.5 Gbps if the transition into this state is from the SP6:OOB_AwaitNoCOMSAS state; or
b) to the value of the SAS Speed Negotiation Window Rate argument.

During this state D.C. idle shall be transmitted.

6.7.4.2.2 Transition SP8:SAS_Start to SP10:SAS_AwaitALIGN

This transition shall occur after the RCDT timer expires if the current speed negotiation window rate is supported.

6.7.4.2.3 Transition SP8:SAS_Start to SP9:SAS_RateNotSupported

This transition shall occur after the RCDT timer expires if the current speed negotiation window rate is not supported.

6.7.4.3 SP9:SAS_RateNotSupported state

6.7.4.3.1 State description

Upon entering this state the SNTT timer shall be initialized and started.

During this state D.C. idle shall be transmitted.

6.7.4.3.2 Transition SP9:SAS_RateNotSupported to SP14:SAS_Fail

This transition shall occur after the SNTT timer expires.

6.7.4.4 SP10:SAS_AwaitALIGN state

6.7.4.4.1 State description

Upon entering this state, the SNTT timer and SNLT timer shall be initialized and started and this state shall 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.

Upon entering this state, this state shall send a Start DWS message to the SP_DWS state machine.

6.7.4.4.2 Transition SP10:SAS_AwaitALIGN to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message if this state does not send a Start DWS message.

6.7.4.4.3 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.

--`,,,``,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.4.4.4 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.

6.7.4.4.5 Transition SP10:SAS_AwaitALIGN to SP14:SAS_Fail

This transition shall occur if the SNTT timer expires.

6.7.4.5 SP11:SAS_AwaitALIGN1 state

6.7.4.5.1 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.

6.7.4.5.2 Transition SP11:SAS_AwaitALIGN1 to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message if this state does not send a Start DWS message.

6.7.4.5.3 Transition SP11:SAS_AwaitALIGN1 to SP14:SAS_Fail

This transition shall occur if the SNTT timer expires. This indicates that the other phy has not been able to lock at the current rate.

6.7.4.5.4 Transition SP11:SAS_AwaitALIGN1 to SP12:SAS_AwaitSNW

This transition shall occur if this state receives an ALIGN Received (1) message before the SNLT timer expires. This indicates that the other phy has been able to lock at the current rate.

6.7.4.6 SP12:SAS_AwaitSNW state

6.7.4.6.1 State description

This state shall repeatedly send Transmit ALIGN (1) messages to the SP transmitter.

If this is the last speed negotiation window, 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.

6.7.4.6.2 Transition SP12:SAS_AwaitSNW to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message if this state does not send a Start DWS message.

6.7.4.6.3 Transition SP12:SAS_AwaitSNW to SP13:SAS_Pass

This transition shall occur after the SNTT timer expires or after receiving a Stop SNTT request.

6.7.4.7 SP13:SAS_Pass state

6.7.4.7.1 State description

This state determines if:

a) another SAS speed negotiation window is required; and
b) the SAS speed negotiation is complete.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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.

6.7.4.7.2 Transition SP13:SAS_Pass to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message if this state does not send a Start DWS message.

6.7.4.7.3 Transition SP13:SAS_Pass to SP8:SAS_Start

This transition shall occur if the state machine has not fallen back (i.e., transitioned from SP14:SAS_Fail to SP8:SAS_Start) during this current SAS speed negotiation sequence.

This transition shall include a SAS Speed Negotiation Window Rate argument with the transition.

6.7.4.7.4 Transition SP13:SAS_Pass to SP15:SAS_PHY_Ready

This transition shall occur if speed negotiation has progressed to where it failed and then had fallen back to the last negotiated speed and then subsequently passed.

6.7.4.8 SP14:SAS_Fail state

6.7.4.8.1 State description

This state determines if the SAS speed negotiation window failure occurred because:

a) the maximum SAS speed negotiation window has been attempted and there haven't been any successful negotiated physical link rates;
b) the SAS speed negotiation failed after dropping back to the last successful SAS speed negotiation window;
c) the SAS speed negotiation has failed and there was a previous successful SAS speed negotiation; or
d) no SAS speed negotiation has previously passed and the maximum SAS speed negotiation window has not yet been attempted.

6.7.4.8.2 Transition SP14:SAS_Fail to SP1:OOB_AwaitCOMX

This transition shall occur if:

a) the maximum SAS speed negotiation window has been attempted and there haven't been any successful negotiated physical link rates; or
b) the SAS speed negotiation failed after dropping back to the last successful SAS speed negotiation window rate.

6.7.4.8.3 Transition SP14:SAS_Fail to SP8:SAS_Start

This transition shall occur:

a) after setting the SAS speed negotiation window rate to one less than the current SAS speed negotiation window rate; and
b) if the SAS speed negotiation has failed and there was a previous successful SAS speed negotiation;

or:

a) after setting the SAS speed negotiation window rate to one greater than the current SAS speed negotiation window rate; and
b) if no SAS speed negotiation has previously passed and the maximum supported SAS speed negotiation window rate has not yet been attempted.

This transition shall include which speed negotiation window rate to attempt next in a SAS Speed Negotiation Window Rate argument.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.4.9 SP15:SAS_PHY_Ready state

6.7.4.9.1 State description

This state waits for a COMINIT Detected message, a DWS Lost message, or a DWS Reset message.

While in this state dwords from the link layer are transmitted at the negotiated physical link rate at the rate established in the previous speed negotiation window.

Upon entering this state, this state shall send a Phy Layer Ready (SAS) confirmation to the link layer to indicate that the physical link has been brought up successfully in SAS mode.

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.

6.7.4.9.2 Transition SP15:SAS_PHY_Ready to SP0:OOB_COMINIT

This transition shall occur after:

a) receiving a COMINIT Detected message;
b) receiving a DWS Lost message, if this state does not send a Start DWS message; or
c) receiving a DWS Reset message.

6.7.5 SATA host emulation states

6.7.5.1 SATA host emulation states overview

Figure 69 shows the SATA host emulation states, in which the phy has detected that it is attached to a SATA device 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 devices that support being attached to SATA devices; expander devices attached to SATA devices do not support power management in this standard.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

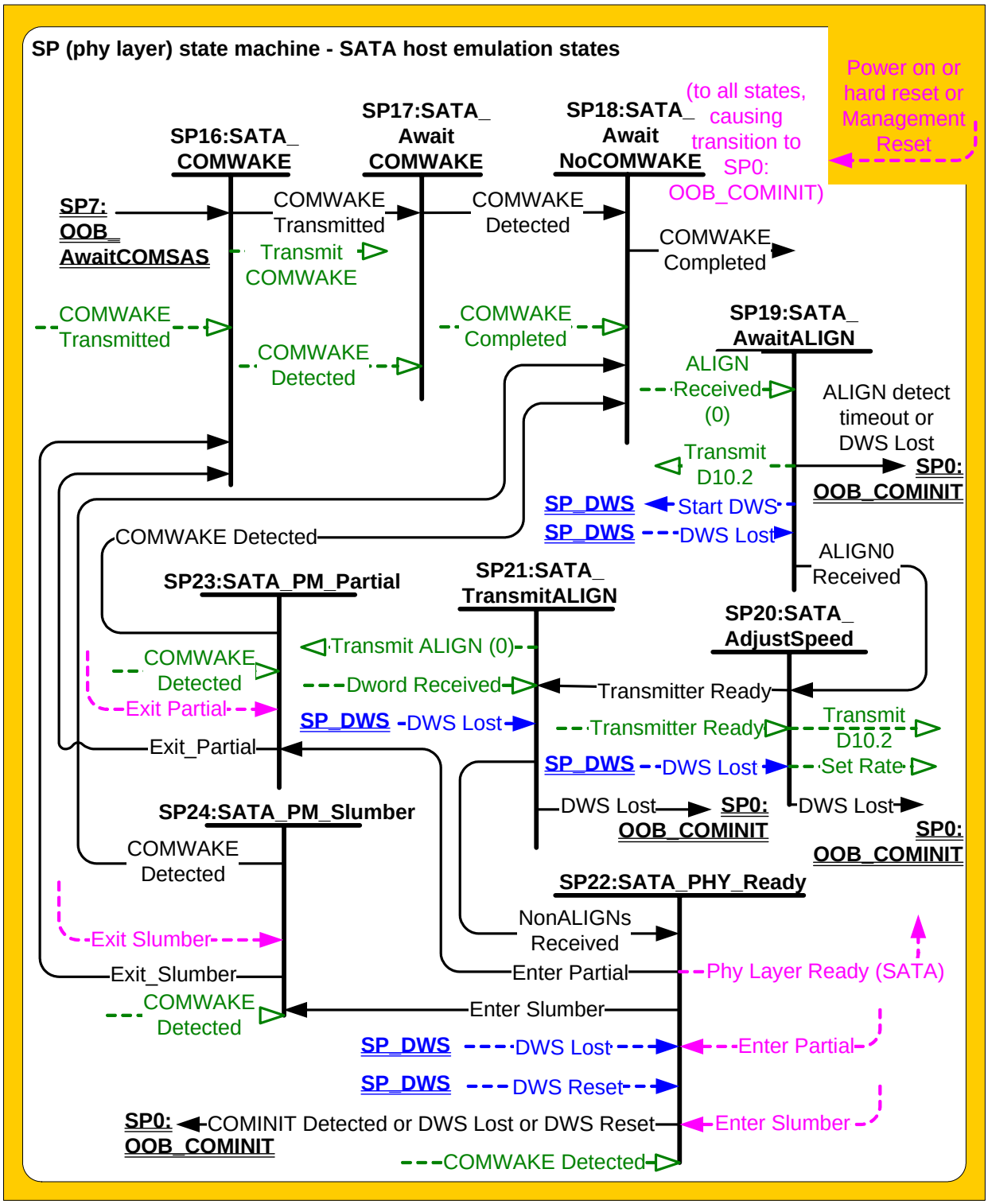


Figure 69 — SP (phy layer) state machine - SATA host emulation states

6.7.5.2 SP16:SATA_COMWAKE state

6.7.5.2.1 State description

This state shall send a Transmit COMWAKE message to the SP transmitter and wait for a COMWAKE Transmitted message.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

6.7.5.2.2 Transition SP16:SATA_COMWAKE to SP17:SATA_AwaitCOMWAKE

This transition shall occur after receiving a COMWAKE Transmitted message.

6.7.5.3 SP17:SATA_AwaitCOMWAKE state

6.7.5.3.1 State description

This state waits for COMWAKE to be received.

6.7.5.3.2 Transition SP17:SATA_AwaitCOMWAKE to SP18:SATA_AwaitNoCOMWAKE

This transition shall occur after receiving a COMWAKE Detected message.

6.7.5.4 SP18:SATA_AwaitNoCOMWAKE state

6.7.5.4.1 State description

This state waits for a COMWAKE Completed message.

6.7.5.4.2 Transition SP18:SATA_AwaitNoCOMWAKE to SP19:SATA_AwaitALIGN

This transition shall occur after receiving a COMWAKE Completed message.

6.7.5.5 SP19:SATA_AwaitALIGN state

6.7.5.5.1 State description

Upon entering this state, this state 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 6.6.2.2) after entry into this state.

6.7.5.5.2 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.

6.7.5.5.3 Transition SP19:SATA_AwaitALIGN to SP0:OOB_COMINIT

This transition shall occur if the Await ALIGN Timeout timer expires.

6.7.5.6 SP20:SATA_AdjustSpeed state

6.7.5.6.1 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 Rate message to the SP transmitter; and
2) repeatedly send Transmit D10.2 messages to the SP transmitter.

6.7.5.6.2 Transition SP20:SATA_AdjustSpeed to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.5.6.3 Transition SP20:SATA_AdjustSpeed to SP21:SATA_TransmitALIGN

This transition shall occur after receiving a Transmitter Ready message.

6.7.5.7 SP21:SATA_TransmitALIGN state

6.7.5.7.1 State description

This state shall repeatedly send Transmit ALIGN (0) messages to the SP transmitter.

6.7.5.7.2 Transition SP21:SATA_TransmitALIGN to SP0:OOB_COMINIT

This transition shall occur after receiving a DWS Lost message.

6.7.5.7.3 Transition SP21:SATA_TransmitALIGN to SP22:SATA_PHY_Ready

This transition shall occur after receiving three consecutive Dword Received messages containing dwords other than ALIGN(0).

6.7.5.8 SP22:SATA_PHY_Ready state

6.7.5.8.1 State description

While in this state dwords from the link layer are transmitted at the negotiated physical link rate at the rate established in the previous speed negotiation window.

This state shall send a Phy Layer Ready (SATA) confirmation to the link layer to indicate that the physical link has been brought up successfully in SATA mode.

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.

6.7.5.8.2 Transition SP22:SATA_PHY_Ready to SP1:OOB_COMINIT

This transition shall occur after:

a) receiving a COMINIT Detected message;
b) receiving a DWS Lost message, if this state does not send a Start DWS message; or
c) receiving a DWS Reset message.

6.7.5.8.3 Transition SP22:SATA_PHY_Ready to SP23:SATA_PM_Partial

This transition shall occur after receiving an Enter Partial request.

6.7.5.8.4 Transition SP22:SATA_PHY_Ready to SP24:SATA_PM_Slumber

This transition shall occur after receiving an Enter Slumber request.

6.7.5.9 SP23:SATA_PM_Partial state

6.7.5.9.1 State description

This state waits for a COMWAKE Detected message or an Exit Partial request.

6.7.5.9.2 Transition SP23:SATA_PM_Partial to SP16:SATA_COMWAKE

This transition shall occur after receiving a Exit Partial request.

6.7.5.9.3 Transition SP23:SATA_PM_Partial to SP18:SATA_AwaitNoCOMWAKE

This transition shall occur after receiving a COMWAKE Detected message.

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.7.5.10 SP24:SATA_PM_Slumber state

6.7.5.10.1 State description

This state waits for a COMWAKE Detected message or an Exit Slumber request.

6.7.5.10.2 Transition SP24:SATA_PM_Slumber to SP16:SATA_COMWAKE

This transition shall occur after receiving a Exit Slumber request.

6.7.5.10.3 Transition SP24:SATA_PM_Slumber to SP18:SATA_AwaitNoCOMWAKE

This transition shall occur after receiving a COMWAKE Detected message.

6.8 SP_DWS (phy layer dword synchronization) state machine

6.8.1 SP_DWS state machine overview

Each phy includes an SP_DWS state machine.

This state machine establishes the same dword boundaries at the receiver as at the attached transmitter by searching for control characters. A receiver in the phy monitors and decodes the incoming data stream and forces K28.5 characters into the first character position to effectively perform dword alignment when requested by the SP_DWS state machine. The receiver continues to reestablish dword alignment by forcing received K28.5 characters into the first character position until a valid primitive 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 monitors invalid dwords that are received. When an invalid dword is detected, it requires two valid dwords to nullify its effect. 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 6.8.3)(initial state);
b) SP_DWS1:Valid1 (see 6.8.4);
c) SP_DWS2:Valid2 (see 6.8.5);
d) SP_DWS3:SyncAcquired (see 6.8.6);
e) SP_DWS4:Lost1 (see 6.8.7);
f) SP_DWS5:Lost1Recovered (see 6.8.8);
g) SP_DWS6:Lost2 (see 6.8.9);
h) SP_DWS7:Lost2Recovered (see 6.8.10);
i) SP_DWS8:Lost3 (see 6.8.11); and
j) SP_DWS9:Lost3Recovered (see 6.8.12).

This state machine shall start in the SP_DWS0:AcquireSync state after:

a) power on;
b) hard reset;
c) receiving a Management Reset request from the management layer (e.g., from the SMP PHY CONTROL function in an expander device); or
d) receiving a Stop DWS message from the SP state machine.

This state machine receives the following messages from the SP state machine (see 6.7):

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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The SP_DWS state machines shall maintain the timers listed in table 45.

Table 45 — SP_DWS timers

Timer	Initial value
DWS Reset Timeout timer	1 ms

Figure 70 shows the SP_DWS state machine.

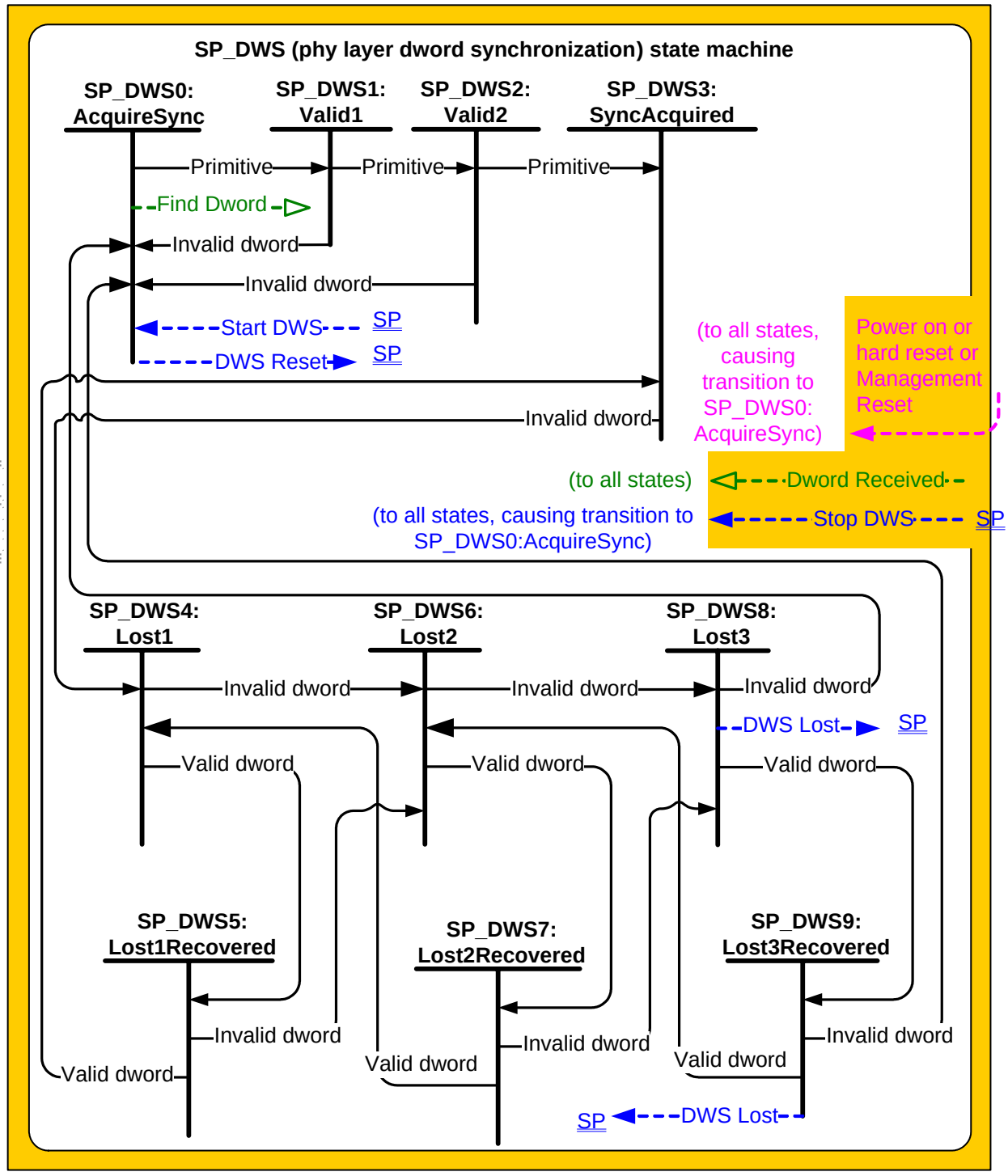


Figure 70 — SP_DWS (phy layer dword synchronization) state machine

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.8.2 SP_DWS receiver

The SP_DWS receiver receives the following messages from the SP_DWS state machine:

a) Find Dword.

The SP_DWS receiver sends the following messages to the SP_DWS state machine:

a) Dword Received (Valid Primitive);
b) Dword Received (Valid Data Dword); and
c) Dword Received (Invalid).

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 as a possible dword. If the next three characters are data characters, it shall send the dword as a Dword Received (Valid Primitive) message to the SP_DWS state machine. Until it receives another Find Dword message, for every four characters 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 3.1.66);
b) send a Dword Received (Valid 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 (Valid Data Dword) message to the SP_DWS state machine if the dword is not an invalid dword or a primitive.

6.8.3 SP_DWS0:AcquireSync state

6.8.3.1 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; and
b) initialize and start the DWS Reset Timeout timer;

If this state is entered from SP_DWS1:Valid1 or SP_DWS2:Valid2, this state shall send a Find Dword message to the SP_DWS receiver. and the DWS Reset Timeout timer shall continue running.

If this state is entered from SP_DWS1:Valid1 or SP_DWS2:Valid2 and the DWS Reset Timeout timer has expired, 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.

If the DWS Reset Timeout timer expires, this state may send a DWS Reset message to the SP state machine.

6.8.3.2 Transition SP_DWS0:AcquireSync to SP_DWS1:Valid1

This transition shall occur after sending a Find Dword message and receiving a Dword Received (Valid Primitive) message.

6.8.4 SP_DWS1:Valid1 state

6.8.4.1 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.

6.8.4.2 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.8.4.3 Transition SP_DWS1:Valid1 to SP_DWS2:Valid2

This transition shall occur after receiving a Dword Received (Valid Primitive) message.

6.8.5 SP_DWS2:Valid2 state

6.8.5.1 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.

6.8.5.2 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.

6.8.5.3 Transition SP_DWS2:Valid2 to SP_DWS3:SyncAcquired

This transition shall occur after receiving a Dword Received (Valid Primitive) message.

6.8.6 SP_DWS3:SyncAcquired state

6.8.6.1 State description

This state is reached after three valid primitives have been received without adjusting the dword synchronization.

The most recently received primitive and all subsequent dwords shall be forwarded for processing by the link layer.

This state waits for a Dword Received (Invalid) message, which indicates that dword synchronization might be lost.

6.8.6.2 Transition SP_DWS3:SyncAcquired to SP_DWS4:Lost1

This transition shall occur after receiving a Dword Received (Invalid) message.

6.8.7 SP_DWS4:Lost1 state

6.8.7.1 State description

This state is reached when one invalid dword has been received and not nullified. This state waits for a Dword Received message.

6.8.7.2 Transition SP_DWS4:Lost1 to SP_DWS5:Lost1Recovered

This transition shall occur after receiving a Dword Received (Valid Data Dword) message or a Dword Received (Valid Primitive) message.

6.8.7.3 Transition SP_DWS4:Lost1 to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Invalid) message.

6.8.8 SP_DWS5:Lost1Recovered state

6.8.8.1 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.8.8.2 Transition SP_DWS5:Lost1Recovered to SP_DWS3:SyncAcquired

This transition shall occur after receiving a Dword Received (Valid Data Dword) message or a Dword Received (Valid Primitive) message.

6.8.8.3 Transition SP_DWS5:Lost1Recovered to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Invalid) message.

6.8.9 SP_DWS6:Lost2 state

6.8.9.1 State description

This state is reached when two invalid dwords have been received and not nullified. This state waits for a Dword Received message.

6.8.9.2 Transition SP_DWS6:Lost2 to SP_DWS7:Lost2Recovered

This transition shall occur after receiving a Dword Received (Valid Data Dword) message or a Dword Received (Valid Primitive) message.

6.8.9.3 Transition SP_DWS6:Lost2 to SP_DWS8:Lost3

This transition shall occur after receiving a Dword Received (Invalid) message.

6.8.10 SP_DWS7:Lost2Recovered state

6.8.10.1 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.

6.8.10.2 Transition SP_DWS7:Lost2Recovered to SP_DWS4:Lost1

This transition shall occur after receiving a Dword Received (Valid Data Dword) message or a Dword Received (Valid Primitive) message.

6.8.10.3 Transition SP_DWS7:Lost2Recovered to SP_DWS8:Lost3

This transition shall occur after receiving a Dword Received (Invalid) message.

6.8.11 SP_DWS8:Lost3 state

6.8.11.1 State description

This state is reached when three invalid dwords have been received and not nullified. This state waits for a Dword Received message.

If a Dword Received (Invalid) message is received (i.e., the fourth non-nullified invalid dword is received), this state shall send a DWS Lost message to the SP state machine.

6.8.11.2 Transition SP_DWS8:Lost3 to SP_DWS9:Lost3Recovered

This transition shall occur after receiving a Dword Received (Valid Data Dword) message or a Dword Received (Valid Primitive) message.

6.8.11.3 Transition SP_DWS8:Lost3 to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6.8.12 SP_DWS9:Lost3Recovered state

6.8.12.1 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.

If a Dword Received (Invalid) message is received (i.e., the fourth non-nullified invalid dword is received), this state shall send a DWS Lost message to the SP state machine.

6.8.12.2 Transition SP_DWS9:Lost3Recovered to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Valid Data Dword) message or a Dword Received (Valid Primitive) message.

6.8.12.3 Transition SP_DWS9:Lost3Recovered to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

6.9 Spin-up

If a SAS target device receives COMSAS during the reset sequence, it shall not spin-up until allowed by the SA_PC state machine (see 10.2.8).

If a SAS target device supporting SATA does not receive COMSAS during the reset sequence, it shall follow SATA spin-up rules (see ATA/ATAPI-7 V3).

Expander devices attached to SATA devices may halt the automatic phy reset sequence to delay spin-up; this is called SATA spinup hold. This is reported in the SMP DISCOVER function (see 10.4.3.5) and is released with the SMP PHY CONTROL function (see 10.4.3.10).

NOTE 8 - 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 spin-up automatically after power on.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7 Link layer

7.1 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.

7.2 Primitives

7.2.1 Primitives overview

Primitives are dwords whose first character is a K28.3, K28.5, or K28.6 control character. Primitives are not considered big-endian or little-endian; they are just interpreted as first, second, third, and last characters. Table 46 defines the primitive format.

Table 46 — Primitive format

Character	Description
First	K28.5 or K28.6 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.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.2.2 Primitive summary

Table 47 defines the primitives not specific to the type of connection.

Table 47 — Primitives not specific to type of connection (Sheet 1 of 2)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [c]
		I	E	T	I	E	T	
AIP (NORMAL)	NoConn		E		I	E	T	Single
AIP (RESERVED 0)	NoConn				I	E	T	Single
AIP (RESERVED 1)	NoConn				I	E	T	Single
AIP (RESERVED 2)	NoConn				I	E	T	Single
AIP (RESERVED WAITING ON PARTIAL)	NoConn				I	E	T	Single
AIP (WAITING ON CONNECTION)	NoConn		E		I	E	T	Single
AIP (WAITING ON DEVICE)	NoConn		E		I	E	T	Single
AIP (WAITING ON PARTIAL)	NoConn		E		I	E	T	Single
ALIGN (0)	All	I	E	T	I	E	T	Single
ALIGN (1)	All	I	E	T	I	E	T	Single
ALIGN (2)	All	I	E	T	I	E	T	Single
ALIGN (3)	All	I	E	T	I	E	T	Single
BREAK	All	I	E	T	I	E	T	Redundant
BROADCAST (CHANGE)	NoConn	I	E		I	E	T	Redundant
BROADCAST (RESERVED 0)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 1)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 2)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 3)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 4)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 0)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 1)	NoConn				I	E	T	Redundant
CLOSE (CLEAR AFFILIATION)	STP	I					T	Triple
CLOSE (NORMAL)	Conn	I		T	I		T	Triple
CLOSE (RESERVED 0)	Conn				I		T	Triple
CLOSE (RESERVED 1)	Conn				I		T	Triple
EOAF	NoConn	I	E	T	I	E	T	Single
ERROR	All		E		I	E	T	Single
HARD_RESET	NoConn	I	E		I	E	T	Redundant
NOTIFY (ENABLE SPINUP)	All	I	E				T	Single
NOTIFY (RESERVED 0)	All				I	E	T	Single
NOTIFY (RESERVED 1)	All				I	E	T	Single
NOTIFY (RESERVED 2)	All				I	E	T	Single
OPEN_ACCEPT	NoConn	I		T	I		T	Single

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 47 — Primitives not specific to type of connection (Sheet 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		I		T	Single
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	NoConn	I	E	T	I		T	Single
OPEN_REJECT (NO DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (PATHWAY BLOCKED)	NoConn		E		I		T	Single
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	NoConn	I		T	I		T	Single
OPEN_REJECT (RESERVED ABANDON 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 2)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 3)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 1)	NoConn				I		T	Single
OPEN_REJECT (RETRY)	NoConn	I		T	I		T	Single
OPEN_REJECT (STP RESOURCES BUSY)	NoConn		E	T	I			Single
OPEN_REJECT (WRONG DESTINATION)	NoConn	I		T	I		T	Single
SOAF	NoConn	I	E	T	I	E	T	Single

[a] The Use column indicates when the primitive is used:
a) NoConn: SAS physical links, outside connections;
b) Conn: SAS physical links, inside connections;
c) All: SAS physical links, both outside connections or inside any type of connection; or
d) STP: SAS physical links, inside STP 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 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 passing through from expander port to expander port.

[c] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

--`,,`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 48 defines the primitives used only inside SSP and SMP connections.

Table 48 — Primitives used only 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	I		T	Single
DONE (CREDIT TIMEOUT)	SSP	I		T	I		T	Single
DONE (NORMAL)	SSP	I		T	I		T	Single
DONE (RESERVED 0)	SSP				I		T	Single
DONE (RESERVED 1)	SSP				I		T	Single
DONE (RESERVED TIMEOUT 0)	SSP				I		T	Single
DONE (RESERVED TIMEOUT 1)	SSP				I		T	Single
EOF	SSP, SMP	I		T	I		T	Single
NAK (CRC ERROR)	SSP	I		T	I		T	Single
NAK (RESERVED 0)	SSP				I		T	Single
NAK (RESERVED 1)	SSP				I		T	Single
NAK (RESERVED 2)	SSP				I		T	Single
RRDY (NORMAL)	SSP	I		T	I		T	Single
RRDY (RESERVED 0)	SSP				I		T	Single
RRDY (RESERVED 1)	SSP				I		T	Single
SOF	SSP, SMP	I		T	I		T	Single

[a] The Use column indicates when the primitive is used:
a) SSP: SAS physical links, inside SSP connections; or
b) SMP: SAS physical 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 passing through from expander port to expander port.

[c] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 49 lists the primitives used only inside STP connections and on SATA physical links.

Table 49 — Primitives used only 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	SATA		E				T	Single
SATA_HOLD	STP, SATA	I		T	I		T	Repeated
SATA_HOLDA	STP, SATA	I		T	I		T	Repeated
SATA_PMACK	STP, SATA							Single
SATA_PMNAK	STP, SATA	I	E				T	Single
SATA_PMREQ_P	STP, SATA							Repeated
SATA_PMREQ_S	STP, SATA							Repeated
SATA_R_ERR	STP, SATA	I		T	I		T	Repeated
SATA_R_IP	STP, SATA	I		T	I		T	Repeated
SATA_R_OK	STP, SATA	I		T	I		T	Repeated
SATA_R_RDY	STP, SATA	I		T	I		T	Repeated
SATA_SOF	STP, SATA	I		T	I		T	Single
SATA_SYNC	STP, SATA	I		T	I		T	Repeated
SATA_WTRM	STP, SATA	I		T	I		T	Repeated
SATA_X_RDY	STP, SATA	I		T	I		T	Repeated

[a] The Use column indicates when the primitive is used:
a) STP: SAS physical 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 passing through from expander port to expander port.

[c] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.2.3 Primitive encodings

Table 50 defines the primitive encoding for primitives not specific to type of connection.

Table 50 — Primitive encoding for primitives not specific to type of connection (Sheet 1 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
AIP (NORMAL)	K28.5	D27.4	D27.4	D27.4
AIP (RESERVED 0)	K28.5	D27.4	D31.4	D16.7
AIP (RESERVED 1)	K28.5	D27.4	D16.7	D30.0
AIP (RESERVED 2)	K28.5	D27.4	D29.7	D01.4
AIP (RESERVED WAITING ON PARTIAL)	K28.5	D27.4	D01.4	D07.3
AIP (WAITING ON CONNECTION)	K28.5	D27.4	D07.3	D24.0
AIP (WAITING ON DEVICE)	K28.5	D27.4	D30.0	D29.7
AIP (WAITING ON PARTIAL)	K28.5	D27.4	D24.0	D04.7
ALIGN (0)	K28.5	D10.2	D10.2	D27.3
ALIGN (1)	K28.5	D07.0	D07.0	D07.0
ALIGN (2)	K28.5	D01.3	D01.3	D01.3
ALIGN (3)	K28.5	D27.3	D27.3	D27.3
BREAK	K28.5	D02.0	D24.0	D07.3
BROADCAST (CHANGE)	K28.5	D04.7	D02.0	D01.4
BROADCAST (RESERVED 0)	K28.5	D04.7	D07.3	D29.7
BROADCAST (RESERVED 1)	K28.5	D04.7	D01.4	D24.0
BROADCAST (RESERVED 2)	K28.5	D04.7	D04.7	D04.7
BROADCAST (RESERVED 3)	K28.5	D04.7	D16.7	D02.0
BROADCAST (RESERVED 4)	K28.5	D04.7	D29.7	D30.0
BROADCAST (RESERVED CHANGE 0)	K28.5	D04.7	D24.0	D31.4
BROADCAST (RESERVED CHANGE 1)	K28.5	D04.7	D27.4	D07.3
CLOSE (CLEAR AFFILIATION)	K28.5	D02.0	D07.3	D04.7
CLOSE (NORMAL)	K28.5	D02.0	D30.0	D27.4
CLOSE (RESERVED 0)	K28.5	D02.0	D31.4	D30.0
CLOSE (RESERVED 1)	K28.5	D02.0	D04.7	D01.4
EOAF	K28.5	D24.0	D07.3	D31.4
ERROR	K28.5	D02.0	D01.4	D29.7
HARD_RESET	K28.5	D02.0	D02.0	D02.0

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 50 — Primitive encoding for primitives not specific to type of connection (Sheet 2 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
NOTIFY (ENABLE SPINUP)	K28.5	D31.3	D31.3	D31.3
NOTIFY (RESERVED 0)	K28.5	D31.3	D07.0	D01.3
NOTIFY (RESERVED 1)	K28.5	D31.3	D01.3	D07.0
NOTIFY (RESERVED 2)	K28.5	D31.3	D10.2	D10.2
OPEN_ACCEPT	K28.5	D16.7	D16.7	D16.7
OPEN_REJECT (BAD DESTINATION)	K28.5	D31.4	D31.4	D31.4
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	K28.5	D31.4	D04.7	D29.7
OPEN_REJECT (NO DESTINATION)	K28.5	D29.7	D29.7	D29.7
OPEN_REJECT (PATHWAY BLOCKED)	K28.5	D29.7	D16.7	D04.7
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	K28.5	D31.4	D29.7	D07.3
OPEN_REJECT (RESERVED ABANDON 0)	K28.5	D31.4	D02.0	D27.4
OPEN_REJECT (RESERVED ABANDON 1)	K28.5	D31.4	D30.0	D16.7
OPEN_REJECT (RESERVED ABANDON 2)	K28.5	D31.4	D07.3	D02.0
OPEN_REJECT (RESERVED ABANDON 3)	K28.5	D31.4	D01.4	D30.0
OPEN_REJECT (RESERVED CONTINUE 0)	K28.5	D29.7	D02.0	D30.0
OPEN_REJECT (RESERVED CONTINUE 1)	K28.5	D29.7	D24.0	D01.4
OPEN_REJECT (RESERVED INITIALIZE 0)	K28.5	D29.7	D30.0	D31.4
OPEN_REJECT (RESERVED INITIALIZE 1)	K28.5	D29.7	D07.3	D16.7
OPEN_REJECT (RESERVED STOP 0)	K28.5	D29.7	D31.4	D07.3
OPEN_REJECT (RESERVED STOP 1)	K28.5	D29.7	D04.7	D27.4
OPEN_REJECT (RETRY)	K28.5	D29.7	D27.4	D24.0
OPEN_REJECT (STP RESOURCES BUSY)	K28.5	D31.4	D27.4	D01.4
OPEN_REJECT (WRONG DESTINATION)	K28.5	D31.4	D16.7	D24.0
SOAF	K28.5	D24.0	D30.0	D01.4

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 51 defines the primitive encodings for primitives used only inside SSP and SMP connections.

Table 51 — Primitive encoding for primitives used only inside SSP and SMP connections

Primitive	Character			
	1st	2nd	3rd	4th (last)
ACK	K28.5	D01.4	D01.4	D01.4
CREDIT_BLOCKED	K28.5	D01.4	D07.3	D30.0
DONE (ACK/NAK TIMEOUT)	K28.5	D30.0	D01.4	D04.7
DONE (CREDIT TIMEOUT)	K28.5	D30.0	D07.3	D27.4
DONE (NORMAL)	K28.5	D30.0	D30.0	D30.0
DONE (RESERVED 0)	K28.5	D30.0	D16.7	D01.4
DONE (RESERVED 1)	K28.5	D30.0	D29.7	D31.4
DONE (RESERVED TIMEOUT 0)	K28.5	D30.0	D27.4	D29.7
DONE (RESERVED TIMEOUT 1)	K28.5	D30.0	D31.4	D24.0
EOF	K28.5	D24.0	D16.7	D27.4
NAK (CRC ERROR)	K28.5	D01.4	D27.4	D04.7
NAK (RESERVED 0)	K28.5	D01.4	D31.4	D29.7
NAK (RESERVED 1)	K28.5	D01.4	D04.7	D24.0
NAK (RESERVED 2)	K28.5	D01.4	D16.7	D07.3
RRDY (NORMAL)	K28.5	D01.4	D24.0	D16.7
RRDY (RESERVED 0)	K28.5	D01.4	D02.0	D31.4
RRDY (RESERVED 1)	K28.5	D01.4	D30.0	D02.0
SOF	K28.5	D24.0	D04.7	D07.3

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 52 lists the primitive encodings for primitives used only inside STP connections and on SATA physical links.

Table 52 — Primitive encoding for primitives used only inside STP connections and on SATA physical links

Primitive	Character			
	1st	2nd	3rd	4th (last)
SATA_CONT	K28.3	D10.5	D25.4	D25.4
SATA_DMAT	K28.3	D21.5	D22.1	D22.1
SATA_EOF	K28.3	D21.5	D21.6	D21.6
SATA_ERROR [a] [b]	K28.6	D02.0	D01.4	D29.7
SATA_HOLD	K28.3	D10.5	D21.6	D21.6
SATA_HOLDA	K28.3	D10.5	D21.4	D21.4
SATA_PMACK	K28.3	D21.4	D21.4	D21.4
SATA_PMNAK	K28.3	D21.4	D21.7	D21.7
SATA_PMREQ_P	K28.3	D21.5	D23.0	D23.0
SATA_PMREQ_S	K28.3	D21.4	D21.3	D21.3
SATA_R_ERR	K28.3	D21.5	D22.2	D22.2
SATA_R_IP	K28.3	D21.5	D21.2	D21.2
SATA_R_OK	K28.3	D21.5	D21.1	D21.1
SATA_R_RDY	K28.3	D21.4	D10.2	D10.2
SATA_SOF	K28.3	D21.5	D23.1	D23.1
SATA_SYNC	K28.3	D21.4	D21.5	D21.5
SATA_WTRM	K28.3	D21.5	D24.2	D24.2
SATA_X_RDY	K28.3	D21.5	D23.2	D23.2

[a] Except for SATA_ERROR, all values are defined by SATA (see ATA/ATAPI-7 V3).

[b] SATA_ERROR does not appear inside STP connections. It is an invalid dword, used by expander devices forwarding an error onto a SATA physical link (see 7.2.7.1).

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.2.4 Primitive sequences

7.2.4.1 Primitive sequences overview

Table 53 summarizes the types of primitive sequences.

Table 53 — Primitive sequences

Primitive sequence type	Number of times to transmit	Number of times received to detect
Single	1	1
Repeated	2	1
Triple	3	3
Redundant	6	3

Any number of ALIGNs and NOTIFYs may be sent inside primitive sequences without affecting the count or breaking the consecutiveness requirements. Rate matching ALIGNs and NOTIFYs shall be sent inside primitive sequences inside of connections if rate matching is enabled (see 7.13).

7.2.4.2 Single primitive sequence

Primitives labeled as single primitive sequences are sent one time.

7.2.4.3 Repeated primitive sequence

Primitives that form repeated primitive sequences (e.g., SATA_HOLD) shall be sent two times, then be followed by SATA_CONT, then be followed by vendor-specific scrambled data dwords. ALIGNs and NOTIFYs may be sent inside primitive sequences as described in 7.2.4.1.

Figure 71 shows an example of a repeated primitive sequence.

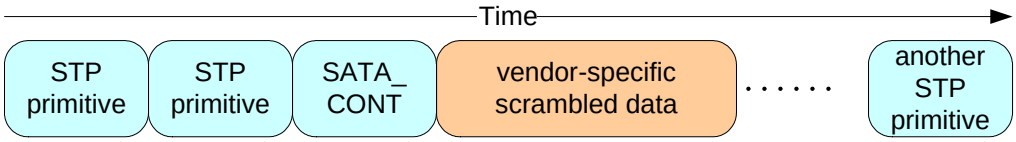


Figure 71 — Repeated primitive sequence

7.2.4.4 Triple primitive sequence

Primitives that form triple primitive sequences (e.g., CLOSE (NORMAL)) shall be sent three times consecutively. ALIGNs and NOTIFYs may be sent inside primitive sequences as described in 7.2.4.1.

Receivers shall detect a triple primitive sequence after the identical primitive is received in three consecutive dwords. After receiving 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:

a) the original primitive; or
b) an ALIGN or NOTIFY.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 72 shows examples of triple primitive sequences.

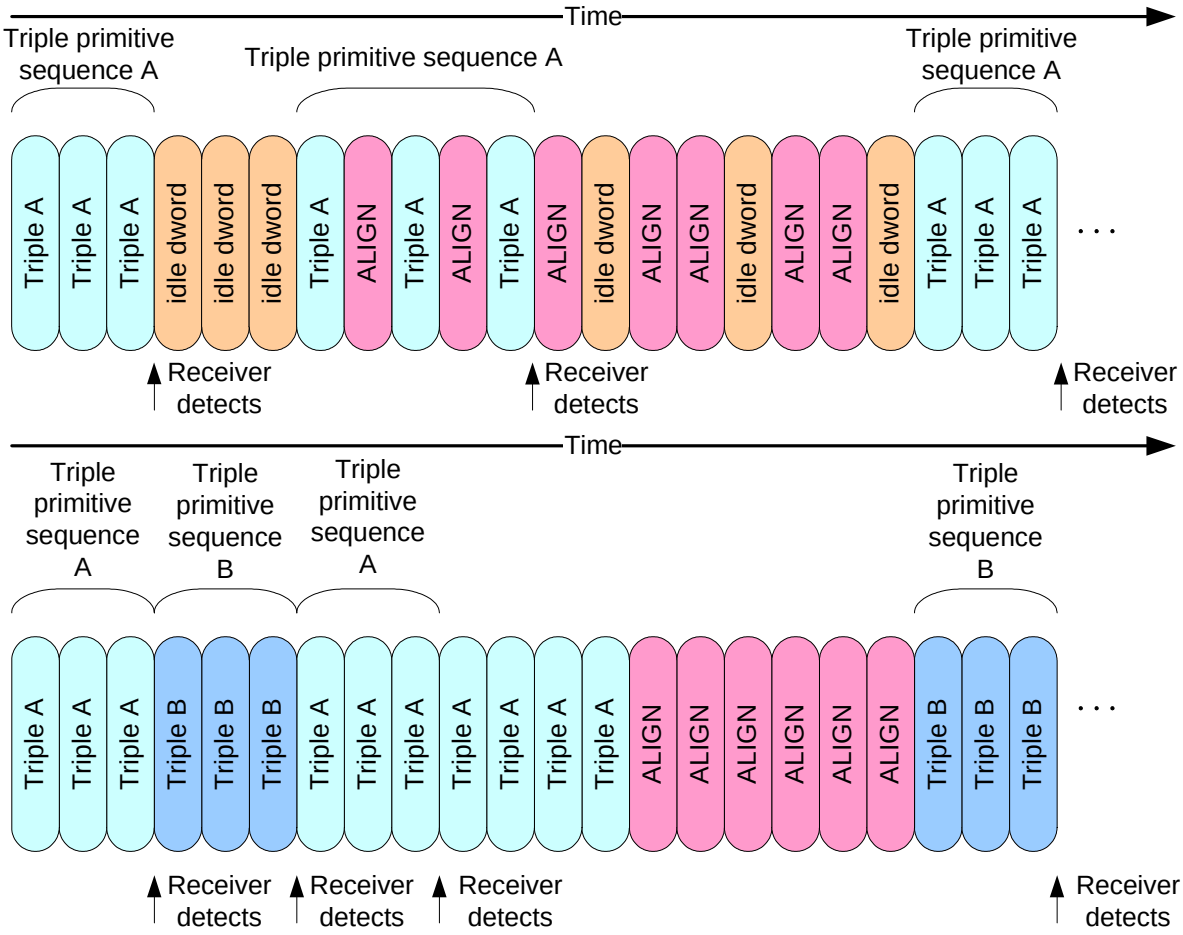


Figure 72 — Triple primitive sequence

7.2.4.5 Redundant primitive sequence

Primitives that form redundant primitive sequences (e.g., BROADCAST (CHANGE)) shall be sent six times consecutively. ALIGNs and NOTIFYs may be sent inside primitive sequences as described in 7.2.4.1.

A receiver shall detect a redundant primitive sequence after the identical primitive is received in three consecutive dwords. After receiving 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) an ALIGN or NOTIFY.

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 73 shows examples of redundant primitive sequences.

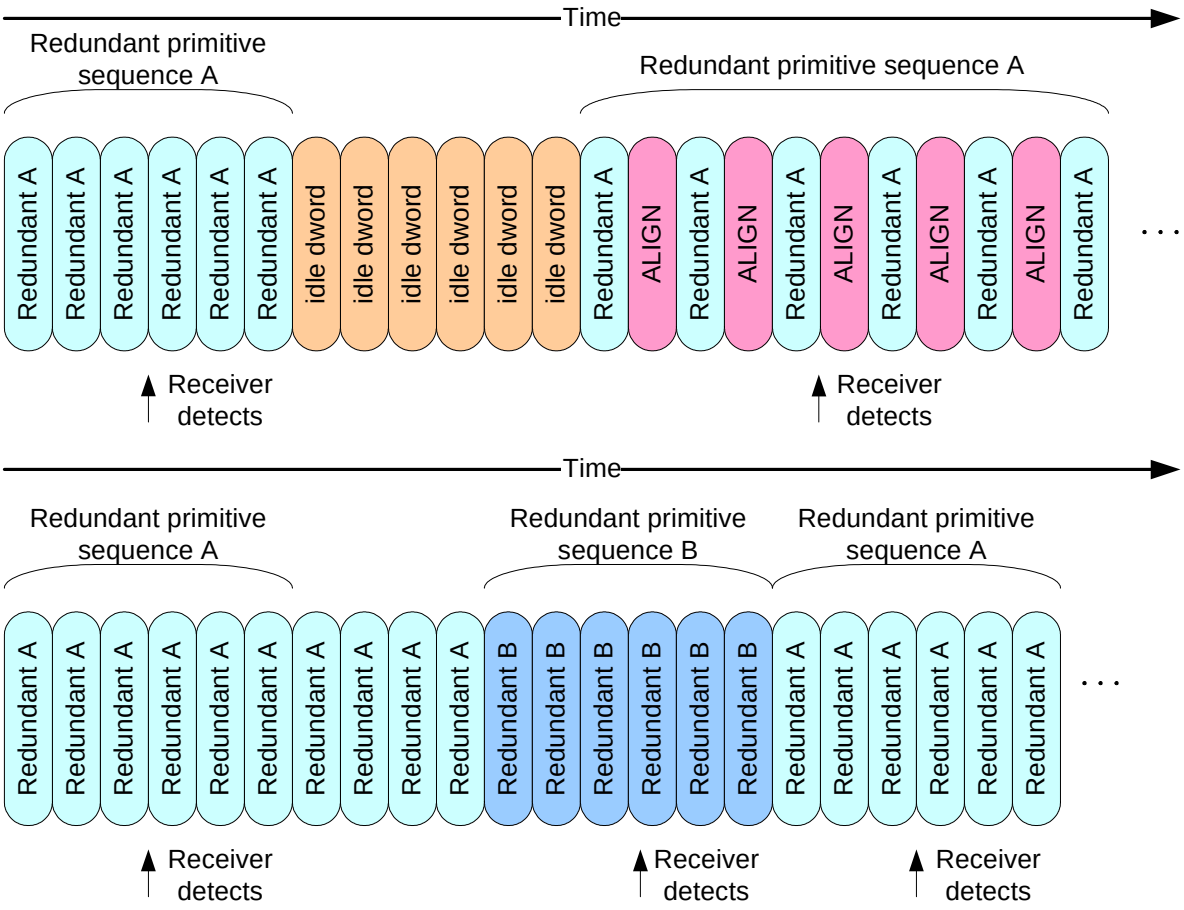


Figure 73 — Redundant primitive sequence

7.2.5 Primitives not specific to type of connections

7.2.5.1 AIP (Arbitration in progress)

AIP is sent by an expander device after a connection request to indicate that the connection request is being processed and indicate the status of the connection request.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The versions of AIP representing different statuses are defined in table 54.

Table 54 — AIP primitives

Primitive	Description
AIP (NORMAL)	Expander device has just accepted the connection request.
AIP (RESERVED 0)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 1)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 2)	Reserved. Processed the same as AIP (NORMAL).
AIP (WAITING ON CONNECTION)	Expander device has determined the routing for the connection request, but the destination phys are all being used for connections.
AIP (WAITING ON DEVICE)	Expander device has determined the routing for the connection request and forwarded it to the output physical link.
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 (i.e., connection requests that have not reached the destination phy).
AIP (RESERVED WAITING ON PARTIAL)	Reserved. Processed the same as AIP (WAITING ON PARTIAL).

See 7.12 for details on connections.

7.2.5.2 ALIGN

ALIGNs are used for:

a) OOB signals;
b) character and dword alignment during the speed negotiation sequence;
c) clock skew management after the phy reset sequence; and
d) rate matching during connections.

Table 55 defines the different versions of ALIGN primitives.

Table 55 — ALIGN primitives

Primitive	Description
ALIGN (0)	Used for OOB signals, the speed negotiation sequence, clock skew management and rate matching.
ALIGN (1)	Used for the speed negotiation sequence, clock skew management and rate matching.
ALIGN (2)	Used for clock skew management and rate matching.
ALIGN (3)	Used for clock skew management and rate matching.

Phys shall use ALIGN(0) to construct OOB signals as described in 6.5. Phys shall use ALIGN(0) and ALIGN(1) during the speed negotiation sequence as described in 6.6.4.2. Phys shall rotate through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) for all ALIGNs sent after the phy reset sequence for clock skew management (see 7.3) and rate matching (see 7.13).

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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

NOTE 9 - 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 clock skew management and rate matching, ALIGNs may be replaced by NOTIFYs (see 7.2.5.9).

7.2.5.3 BREAK

BREAK is used to abort a connection request or break a connection.

See 7.12.6 and 7.12.8 for details on breaking connections.

7.2.5.4 BROADCAST

BROADCASTs are used to notify all SAS ports in a domain of an event.

The versions of BROADCAST representing different reasons are defined in table 56.

Table 56 — BROADCAST primitives

Primitive	Description
BROADCAST (CHANGE)	Notification of a configuration change.
BROADCAST (RESERVED CHANGE 0)	Reserved. Processed the same as BROADCAST (CHANGE) by SAS ports (i.e, SAS initiator ports and SAS target ports).
BROADCAST (RESERVED CHANGE 1)	Reserved. Processed the same as BROADCAST (CHANGE) by SAS ports (i.e., SAS initiator ports and SAS target ports).
BROADCAST (RESERVED 0)	Reserved.
BROADCAST (RESERVED 1)	Reserved.
BROADCAST (RESERVED 2)	Reserved.
BROADCAST (RESERVED 3)	Reserved.
BROADCAST (RESERVED 4)	Reserved.

When an expander port receives a BROADCAST it shall transmit the same BROADCAST on at least one phy in all other expander ports. BROADCAST shall only be sent outside of connections after the phy reset sequence has completed.

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, the second BROADCAST may be ignored.

BROADCAST (CHANGE) is sent by an expander device to notify SAS initiator ports and other expander devices that a configuration change has occurred. BROADCAST (CHANGE) may also be transmitted by SAS initiator ports. BROADCAST (CHANGE) shall be ignored by SAS target ports.

BROADCAST (RESERVED CHANGE 0) and BROADCAST (RESERVED CHANGE 1) shall be processed the same as BROADCAST (CHANGE) by SAS ports. BROADCAST (RESERVED 0), BROADCAST (RESERVED 1), BROADCAST (RESERVED 2), BROADCAST (RESERVED 3), and BROADCAST (RESERVED 4) shall be ignored by SAS ports.

See 7.11 for details on SAS domain changes. See 10.4.3.3 for details on counting BROADCAST (CHANGE) generation in an expander device.

7.2.5.5 CLOSE

CLOSE is used to close a connection. This primitive may be originated by a SAS initiator port or a SAS target port.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The versions of CLOSE representing different reasons are defined in table 57.

Table 57 — CLOSE primitives

Primitive	Description
CLOSE (CLEAR AFFILIATION)	Close an open STP connection and clear the affiliation (see 7.17.3). 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)	Reserved. Processed the same as CLOSE (NORMAL).

See 7.12.7 for details on closing connections.

7.2.5.6 EOAF (End of address frame)

EOAF indicates the end of an address frame.

See 7.8 for details on address frames.

7.2.5.7 ERROR

ERROR is sent by an expander device when it is forwarding dwords from a SAS physical link or SATA physical link to a SAS physical link and it receives an invalid dword.

See 7.15 for details on error handling by expander devices.

SAS phys may ignore ERROR or treat it as an invalid dword.

7.2.5.8 HARD_RESET

HARD_RESET is used to force a phy to generate a hard reset 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.

7.2.5.9 NOTIFY

NOTIFY may be transmitted in place of any ALIGN being transmitted for clock skew management (see 7.3) or rate matching (see 7.13). Substitution of a NOTIFY may or may not affect the ALIGN sequencing (i.e., the NOTIFY may take the place of one of the ALIGNs in the rotation through ALIGN (0), ALIGN (1), ALIGN (2), or ALIGN (3) or it may delay the rotation). A specific NOTIFY shall not be transmitted a second time until at least three ALIGNs or different NOTIFYs have been transmitted.

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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The versions of NOTIFY representing different reasons are defined in table 58.

Table 58 — NOTIFY primitives

Primitive	Description
NOTIFY (ENABLE SPINUP)	Specify to an SAS target device that it may temporarily consume additional power while transitioning into the active or idle power condition state.
NOTIFY (RESERVED 0)	Reserved.
NOTIFY (RESERVED 1)	Reserved.
NOTIFY (RESERVED 2)	Reserved.

NOTIFY (ENABLE SPINUP) is transmitted by a SAS initiator port or expander port and is used to specify to an SAS target device that it may temporarily consume additional power (e.g. while spinning-up rotating media) while transitioning into the active or idle power condition state. 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-3) and/or the START STOP UNIT command (see SBC-2), as described in 10.2.8.

SAS initiator devices and expander devices shall use NOTIFY (ENABLE SPINUP) while attached to SAS target devices (i.e., devices that report SSP target support in their IDENTIFY address frames). They shall transmit one NOTIFY (ENABLE SPINUP) after power on when the enclosure is ready for initial spin-up. After the initial NOTIFY (ENABLE SPINUP), they 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 10 - The SAS initiator device or expander device uses NOTIFY (ENABLE SPINUP) to avoid exceeding enclosure power supply capabilities during spin-up of 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 spinning up at a time. An expander device may allow this timing to be configured by a NVROM programming with enclosure-specific sequencing patterns, or may employ more complex, dynamic interaction with the enclosure power supply.

NOTE 11 - NOTIFY (ENABLE SPINUP) should be transmitted as frequently as possible to avoid incurring application layer timeouts.

I_T nexus loss, logical unit reset, and hard reset shall not cause a SAS target device to spin-up automatically on receipt of NOTIFY (ENABLE SPINUP).

SAS target devices with multiple SAS target ports shall honor NOTIFY (ENABLE SPINUP) from all SAS target ports equivalently (e.g., NOTIFY (ENABLE SPINUP) received on SAS target port A serves as a wakeup for a START STOP UNIT command received through SAS target port B).

NOTIFY (RESERVED 0), NOTIFY (RESERVED 1), and NOTIFY (RESERVED 2) shall be ignored by all devices.

7.2.5.10 OPEN_ACCEPT

OPEN_ACCEPT indicates the acceptance of a connection request.

See 7.12 for details on connection requests.

7.2.5.11 OPEN_REJECT

OPEN_REJECT indicates that a connection request has been rejected and indicates 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

All of the OPEN_REJECT versions defined in table 59 shall result in the originating device abandoning the connection request.

Table 59 — OPEN_REJECT abandon primitives

Primitive	Originator	Description
OPEN_REJECT (BAD DESTINATION)	Expander phy	An expander device receives a request in which the destination SAS address equals the source SAS address, or a connection request arrives through an expander phy using the direct routing or table routing method and the expander device determines the connection request would have to be routed to the same expander port as the expander port through which the connection request arrived.
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. The connection request may be modified and reattempted as described in 7.12.2.2.
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	Destination phy	Device with destination SAS address exists but the destination device does not support the requested initiator/ 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 0)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 1)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 2)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 3)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (STP RESOURCES BUSY)	Destination phy	STP target port with destination SAS address exists but the STP target port has an affiliation with another STP initiator port or all of the available task file registers have been allocated to other STP initiator ports (see 7.17.3). 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.

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

All of the OPEN_REJECT versions defined in table 60 shall result in the originating device retrying the connection request.

Table 60 — OPEN_REJECT retry primitives

Primitive	Originator	Description
OPEN_REJECT (NO DESTINATION) [c]	Expander phy	Either: a) No such destination device; b) a connection request arrives through an expander phy using the subtractive routing method and the expander device determines the connection request would have to be routed to the same expander port as the expander port through which the connection request arrived; 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 not been successfully received (see 10.4.3.7).
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) [a]	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED CONTINUE 1) [a]	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED INITIALIZE 0) [c]	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED INITIALIZE 1) [c]	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED STOP 0) [b]	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED)
OPEN_REJECT (RESERVED STOP 1) [b]	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED).
OPEN_REJECT (RETRY) [a]	Destination phy	Device with destination SAS address exists but is not able to accept connections.

[a] If the I_T Nexus Loss timer (see 8.2.2) is already running, it is stopped.
[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 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.

When a destination device detects more than one reason to transmit an OPEN_REJECT, the device shall transmit only one OPEN_REJECT and shall select the primitive using the following priority:

1) OPEN_REJECT (WRONG DESTINATION) (highest priority selection);
2) OPEN_REJECT (PROTOCOL NOT SUPPORTED);
3) OPEN_REJECT (CONNECTION RATE NOT SUPPORTED);
4) OPEN_REJECT (STP RESOURCES BUSY); or
5) OPEN_REJECT (RETRY) (lowest priority selection).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

When an expander device detects more than one reason to transmit an OPEN_REJECT, the expander shall transmit only one OPEN_REJECT primitive and shall select that primitive using the following priority:

1) OPEN_REJECT (BAD DESTINATION) or OPEN_REJECT (NO DESTINATION) (highest priority selection);
2) OPEN_REJECT (CONNECTION RATE NOT SUPPORTED); or
3) OPEN_REJECT (STP RESOURCES BUSY) or OPEN_REJECT (PATHWAY BLOCKED) (lowest priority selection).

See 7.12 for details on connection requests.

7.2.5.12 SOAF (Start of address frame)

SOAF indicates the start of an address frame.

See 7.8 for details on address frames.

7.2.6 Primitives used only inside SSP and SMP connections

7.2.6.1 ACK (Acknowledge)

ACK indicates the positive acknowledgement of an SSP frame.

See 7.16.3 for details on SSP frame transmission.

7.2.6.2 CREDIT_BLOCKED

CREDIT_BLOCKED indicates that no more credit is going to be sent during this connection.

See 7.16.4 for details on SSP flow control.

7.2.6.3 DONE

DONE is used to start closing an SSP connection and indicate a reason for doing so. This primitive may be originated by an SSP initiator port or an SSP target port. DONE is not used to close an SMP or STP connection.

The versions of DONE representing different reasons are defined in table 61. The SSP state machine describes when these are used (see 7.16.7).

Table 61 — DONE primitives

Primitive	Description
DONE (ACK/NAK TIMEOUT)	The SSP state machine (see 7.16.7) timed out waiting for an ACK or NAK and the transmitter is going to transmit BREAK 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)	Reserved. Processed the same as DONE (ACK/NAK TIMEOUT).
DONE (NORMAL)	Finished transmitting all frames.
DONE (RESERVED 0)	Reserved. Processed the same as DONE (NORMAL).
DONE (RESERVED 1)	Reserved. Processed the same as DONE (NORMAL).
DONE (CREDIT TIMEOUT)	The SSP state machine (see 7.16.7) timed out waiting for an RRDY or received a CREDIT BLOCKED and the transmitter is going to transmit BREAK if credit is extended for 1 ms without receiving a frame or a DONE.

See 7.16.6 for details on closing SSP connections.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.2.6.4 EOF (End of frame)

EOF indicates the end of an SSP or SMP frame.

See 7.16.3 for details on SSP frame transmission and 7.18.1 for details on SMP frame transmission.

7.2.6.5 NAK (Negative acknowledgement)

NAK indicates the negative acknowledgement of an SSP frame and the reason for doing so.

The versions of NAK representing different reasons are defined in table 62.

Table 62 — NAK primitives

Primitive	Description
NAK (CRC ERROR)	The frame had a bad CRC.
NAK (RESERVED 0)	Reserved. Processed the same as NAK (CRC ERROR).
NAK (RESERVED 1)	Reserved. Processed the same as NAK (CRC ERROR).
NAK (RESERVED 2)	Reserved. Processed the same as NAK (CRC ERROR).

See 7.16.3 for details on SSP frame transmission.

7.2.6.6 RRDY (Receiver ready)

RRDY is used to increase SSP frame credit.

The versions of RRDY representing different reasons are defined in table 63.

Table 63 — 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)	Reserved. Processed the same as RRDY (NORMAL).

See 7.16.4 for details on SSP flow control.

7.2.6.7 SOF (Start of frame)

SOF indicates the start of an SSP or SMP frame.

See 7.16.3 for details on SSP frame transmission and 7.18.1 for details on SMP frame transmission.

7.2.7 Primitives used only inside STP connections and on SATA physical links

7.2.7.1 SATA_ERROR

SATA_ERROR is sent by an expander device when it is forwarding dwords from a SAS physical link to a SATA physical link and it receives an invalid dword or an ERROR. SATA_ERROR is an invalid dword.

See 6.8 for details on error handling by expander devices.

7.2.7.2 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 interface power management partial and slumber states. SATA_PMACK is used to accept a power management request. SATA_PMNAK is used to reject a power management request.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

See 7.10 for rules on handling the power management primitives.

7.2.7.3 SATA_HOLD and SATA_HOLDA (Hold and hold acknowledge)

See 7.17.2 for rules on STP flow control, which uses SATA_HOLD and SATA_HOLDA.

7.2.7.4 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. Expander devices shall not transmit SATA_R_RDY or SATA_X_RDY on the SATA physical link until the STP connection is established.

7.2.7.5 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.

7.3 Clock skew management

The internal clock for a device is typically based on a PLL with its own clock generator and is used when transmitting dwords on the physical link. When receiving, however, dwords need to be latched based on a clock derived from the input bit stream itself. Although the input clock is nominally a fixed frequency, it may differ slightly from the internal clock frequency due to accepted manufacturing tolerance and, for SATA physical links, due to spread spectrum clocking. Over time, if the input clock is faster than the internal clock, the device may receive a dword and not be able to forward it to an internal buffer; this is called an overrun. If the input clock is slower than the internal clock, the device may not have a dword when needed in an internal buffer; this is called an underrun.

To solve this problem, transmitting devices insert ALIGNs or NOTIFYs in the dword stream. Receivers may pass ALIGNs and NOTIFYs through to their internal buffers, or may strip them out when an overrun occurs. Receivers add ALIGNs or NOTIFYs when an underrun occurs. The internal logic shall ignore all ALIGNs and NOTIFYs that arrive in the internal buffers.

Elasticity buffer circuitry, as shown in figure 74, is required to absorb the slight differences in frequencies between the SAS initiator phy, SAS target phy, and expander phys. The frequency tolerance for a phy is specified in 5.3.2.

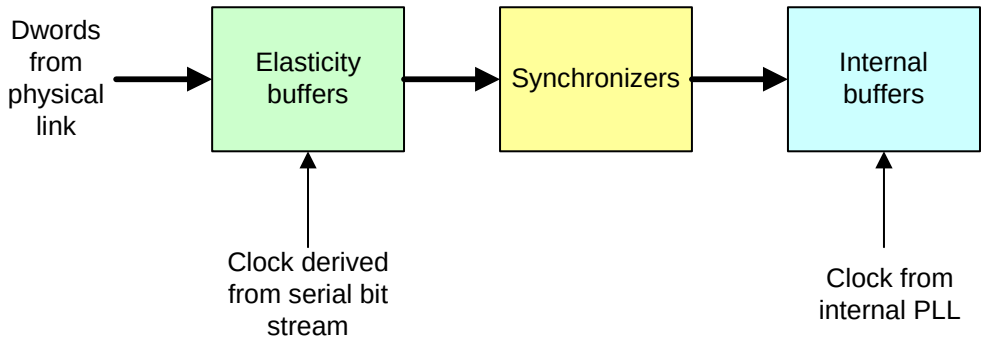


Figure 74 — Elasticity buffers

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A phy that is the original source for the dword stream (i.e., a phy that is not an expander phy forwarding dwords from another expander phy) shall periodically insert ALIGNs or NOTIFYs into the dword stream as shown in table 64.

Table 64 — Clock skew management ALIGN or NOTIFY insertion requirements

Original source of dword stream	Clock skew management ALIGN or NOTIFY requirements
Either: a) SSP initiator phy or SSP target phy in SSP connection; b) SMP initiator phy or SMP target phy in SMP connection; c) STP target phy in an STP connection; or d) any phy outside connections.	One ALIGN or NOTIFY within every 2 048 dwords
STP initiator phy in an STP connection	Two consecutive ALIGNs or NOTIFYs within each 256 dwords plus one ALIGN or NOTIFY within each 2 048 dwords

See 7.2.5.2 for details on rotating through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3). NOTIFYs may also be used in place of ALIGNs (see 7.2.5.9) on SAS physical links.

An expander device that is forwarding dwords (i.e., is not the original source) is allowed to insert or delete as many ALIGNs or NOTIFYs as required to match the transmit and receive connection rates (e.g., it is not required to ensure that it transmits one ALIGN or NOTIFY within every 2 048 dwords when forwarding to a SAS physical link).

NOTE 12 - One possible implementation for expander devices forwarding dwords is for the expander device to delete all ALIGNs and NOTIFYs received and to insert ALIGNs at the transmit port whenever its elasticity buffer is empty.

The STP target port of an STP/SATA bridge is allowed to insert or delete as many ALIGNs or NOTIFYs as required to match the transmit and receive connection rates (e.g., it is not required to ensure that it transmits one ALIGN or NOTIFY within every 2 048 dwords when forwarding to a SAS physical link). The STP target port in an STP/SATA bridge is not required to insert ALIGNs or NOTIFYs in pairs when transmitting dwords.

NOTE 13 - Due to clock skew ALIGN and NOTIFY removal, the STP target port may not receive a pair of ALIGNs or NOTIFYs every 256 dwords, even though the STP initiator port transmitted at least one pair. However, the rate of the dword stream allows for ALIGN or NOTIFY insertion by the STP/SATA bridge. One possible implementation is for the STP/SATA bridge to delete all ALIGNs and NOTIFYs 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.

7.4 Idle physical links

Idle dwords are vendor-specific data dwords.

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.

While an STP connection is open, STP phys transmit SATA_SYNC between frames (see ATA/ATAPI-7 V3). After transmitting two SATA_SYNCs, STP phys shall transmit SATA_CONT and start transmitting idle dwords.

NOTE 14 - SATA devices are allowed but not required to transmit SATA_CONT.

Idle dwords are scrambled (see 7.6).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.5 CRC

7.5.1 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 (see ATA/ATAPI-7 V3). 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 65 defines the CRC polynomials.

Table 65 — CRC polynomials

Function	Definition
F(x)	A polynomial of degree k-1 that is used to represent the k bits of the frame covered by the CRC. For the purposes of the CRC, the coefficient of the highest order term shall be the first bit transmitted.
L(x)	A degree 31 polynomial with all of the coefficients set to one: $L(x) = x^{31} + x^{30} + x^{29} + \ldots + x^2 + x^1 + 1$ (i.e., L(x) = FFFFFFFFh)
G(x)	The standard generator 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.
P(x)	The remainder polynomial on the receive checking side, which is of degree less than 32.
Q(x)	The greatest multiple of G(x) in $(x^{32} \times F(x)) + (x^k \times L(x))$
Q'(x)	$x^{32} \times Q(x)$
M(x)	The sequence that is transmitted.
M'(x)	The sequence that is received.
C(x)	A unique polynomial remainder produced by the receiver upon reception of an error free sequence. This polynomial has the value: $C(x) = x^{32} \times \frac{L(x)}{G(x)}$ $C(x) = x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$ (i.e., C(x) = C704DD7Bh)

7.5.2 CRC generation

The equations that are used to generate the CRC from F(x) are as follows. All arithmetic is modulo 2.

CRC value in frame = L(x) + R(x) = one's complement of R(x)

NOTE 15 - Adding L(x) (all ones) to R(x) produces the one's complement of R(x); this equation is specifying that the R(x) is inverted before it is transmitted.

The CRC is calculated by the following equation:

$$\frac{(x^{32} \times F(x)) + (x^K \times L(x))}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The following equation specifies that the CRC is appended to the end of F(x):

$$M(x) = x^{32} \times F(x) + CRC$$

The bit order of F(x) presented to the CRC function 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). This order is shown in figure 75.

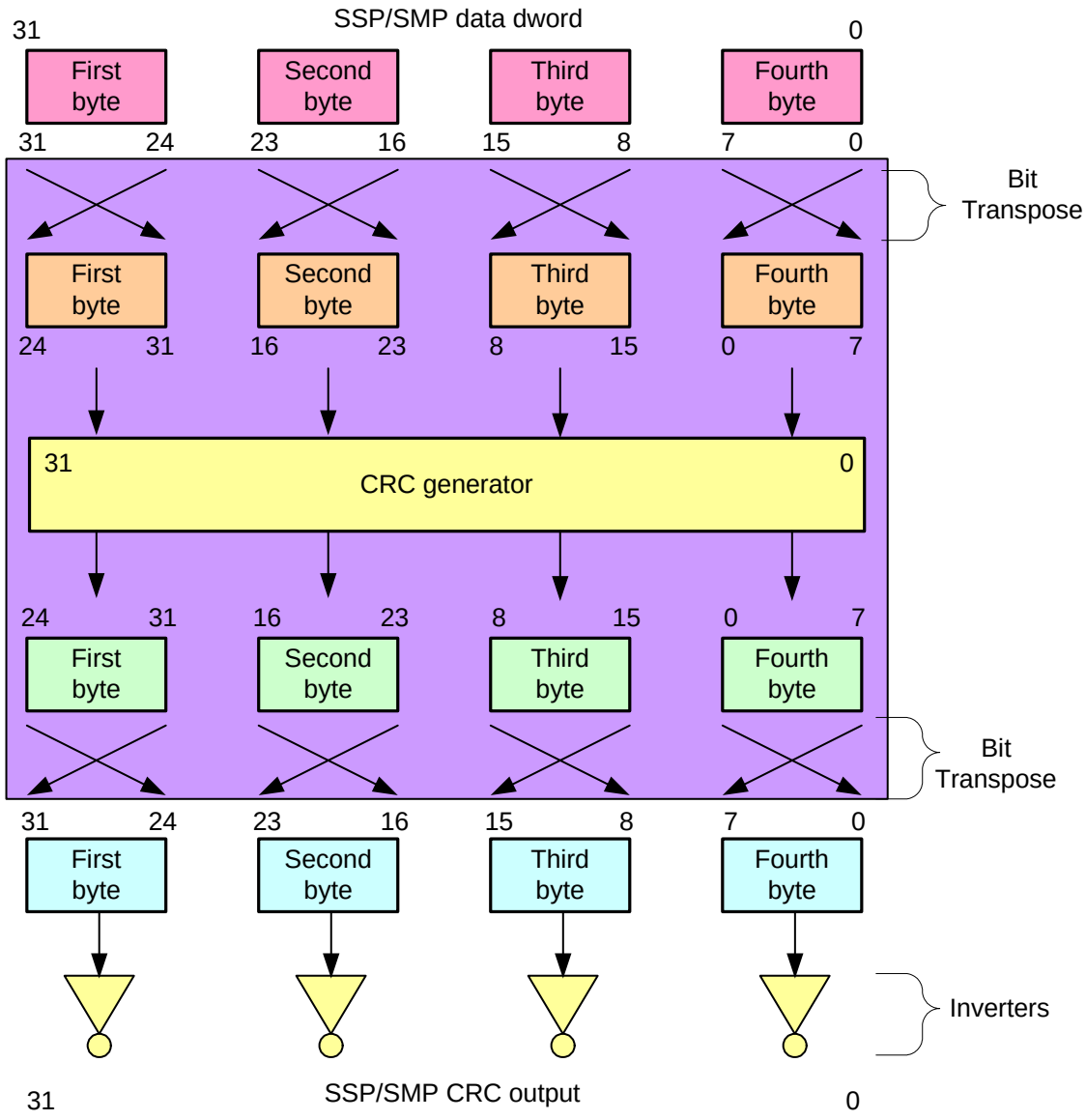


Figure 75 — CRC generator bit order

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Dwords in STP frames are little-endian and feed into the STP CRC generator without swapping bits within each byte and inverting the output like the SAS CRC generator. Figure 76 shows the STP CRC bit ordering.

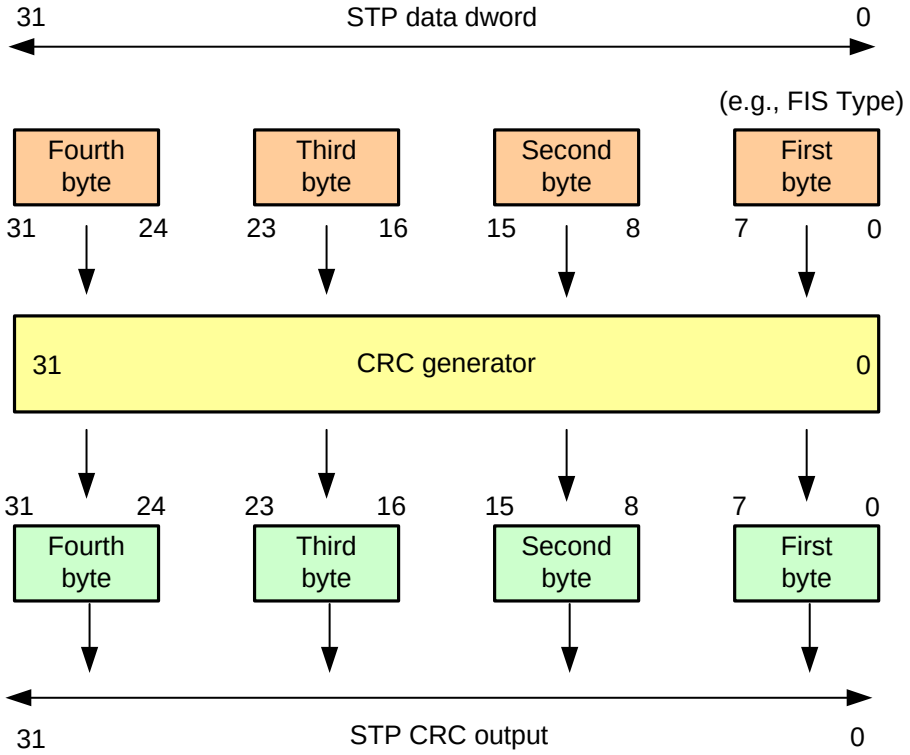


Figure 76 — STP CRC bit ordering

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. Thus, the first byte contains the least-significant bit. In SSP and SMP, the first byte contains the most-significant bit.

See 7.7 for details on how the CRC generator fits into the dword flow along with the scrambler.

7.5.3 CRC checking

The received sequence M'(x) may differ from the transmitted sequence M(x) if there are transmission errors. The process of checking the sequence for validity involves dividing the received sequence by G(x) and testing the remainder. Direct division, however, does not yield a unique remainder because of the possibility of leading zeros. Thus a term L(x) is prepended to M'(x) before it is divided. Mathematically, the received checking is shown by the following equation:

$$x^{32} \times \frac{M'(x) + (x^{K} \times L(x))}{G(x)} = Q'(x) + \frac{P(x)}{G(x)}$$

In the absence of errors, the unique remainder is the remainder of the division as shown by the following equation:

$$\frac{P(x)}{G(x)} = x^{32} \times \frac{L(x)}{G(x)} = C(x)$$

The bit order of F(x) presented to the CRC checking function is the same order as the CRC generation bit order (see figure 75). See 7.7 for details on where the CRC checker fits into the dword flow along with the descrambler.

7.6 Scrambling

Scrambling is used to reduce the probability of long strings of repeated patterns appearing on the physical link.

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

All data dwords are scrambled. Table 66 lists the scrambling for different types of data dwords.

Table 66 — 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.
	Repeated SATA primitive	After a SATA_CONT, vendor-specific scrambled data dwords shall be sent until a primitive other than ALIGN or NOTIFY 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 the 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; the next output of the generator is applied to the upper 16 bits (i.e., bits 31 through 16).

NOTE 16 - 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 SOF_CONT, see ATA/ATAPI-7 V3.

NOTE 17 - STP scrambling uses two linear feedback shift registers, since repeated SATA primitives may occur inside STP frames and they have independent scrambling patterns.

Annex E contains information on scrambling implementations.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.7 Bit order of CRC and scrambler

Figure 77 shows how data dwords and primitives are routed to the bit transmission logic in figure 55. Data dwords go through the CRC generator and scrambler.

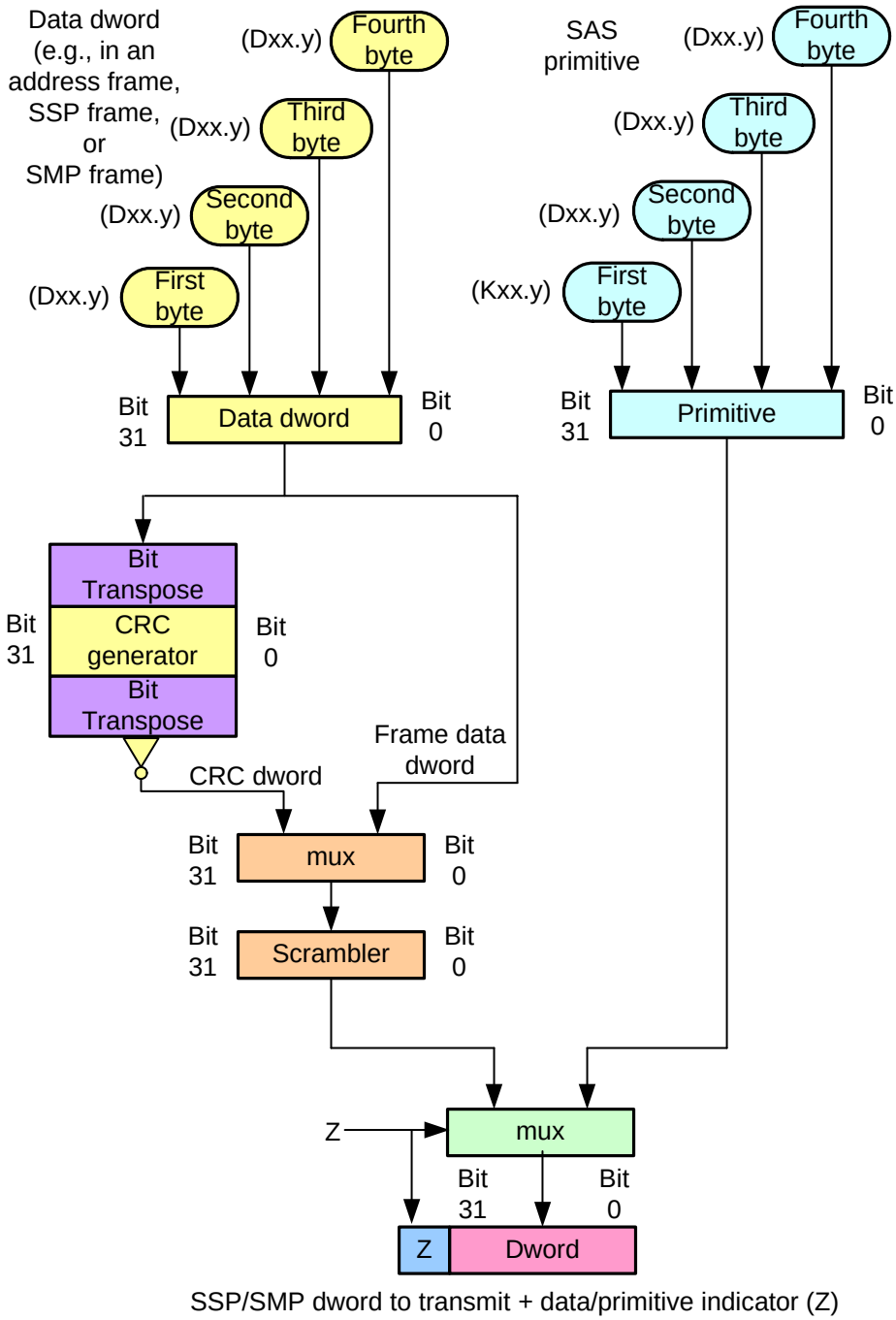


Figure 77 — Transmit path bit ordering

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 78 shows the routing of dwords received from the bit reception logic in figure 56.

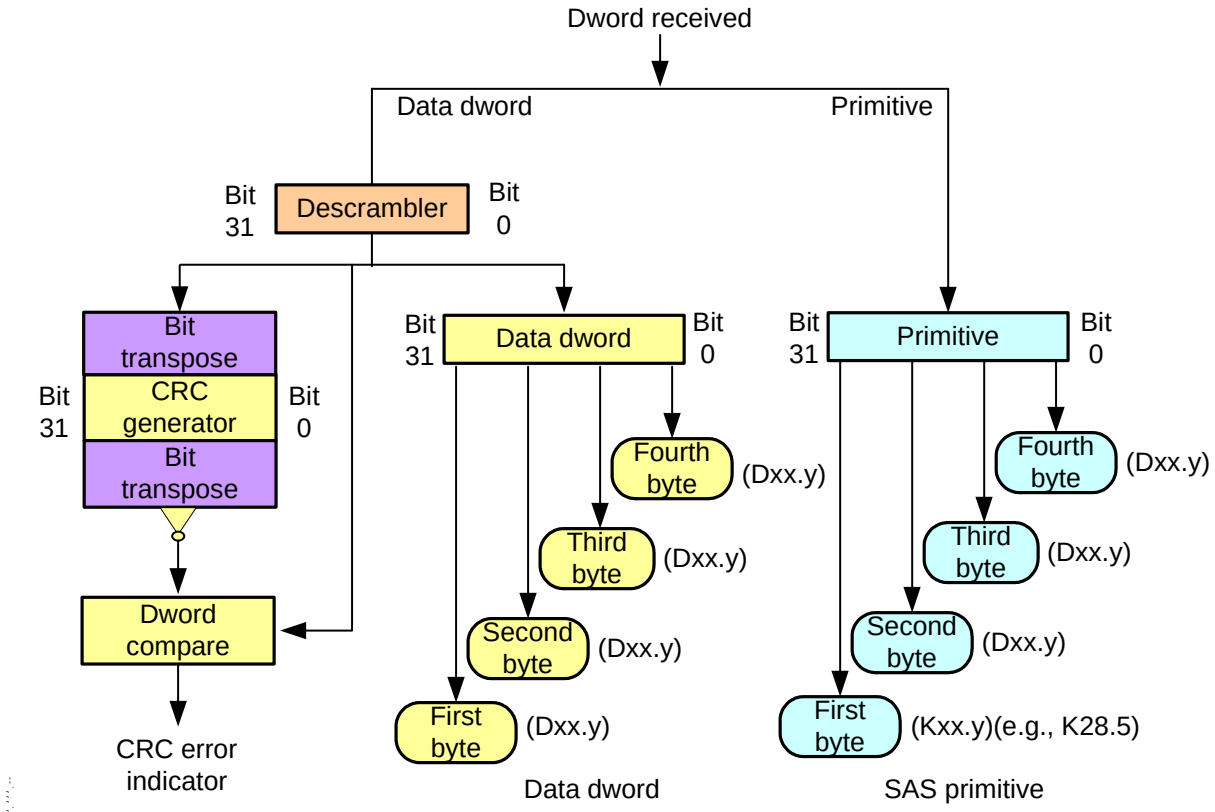


Figure 78 — Receive path bit ordering

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 79 shows the STP transmit path bit ordering.

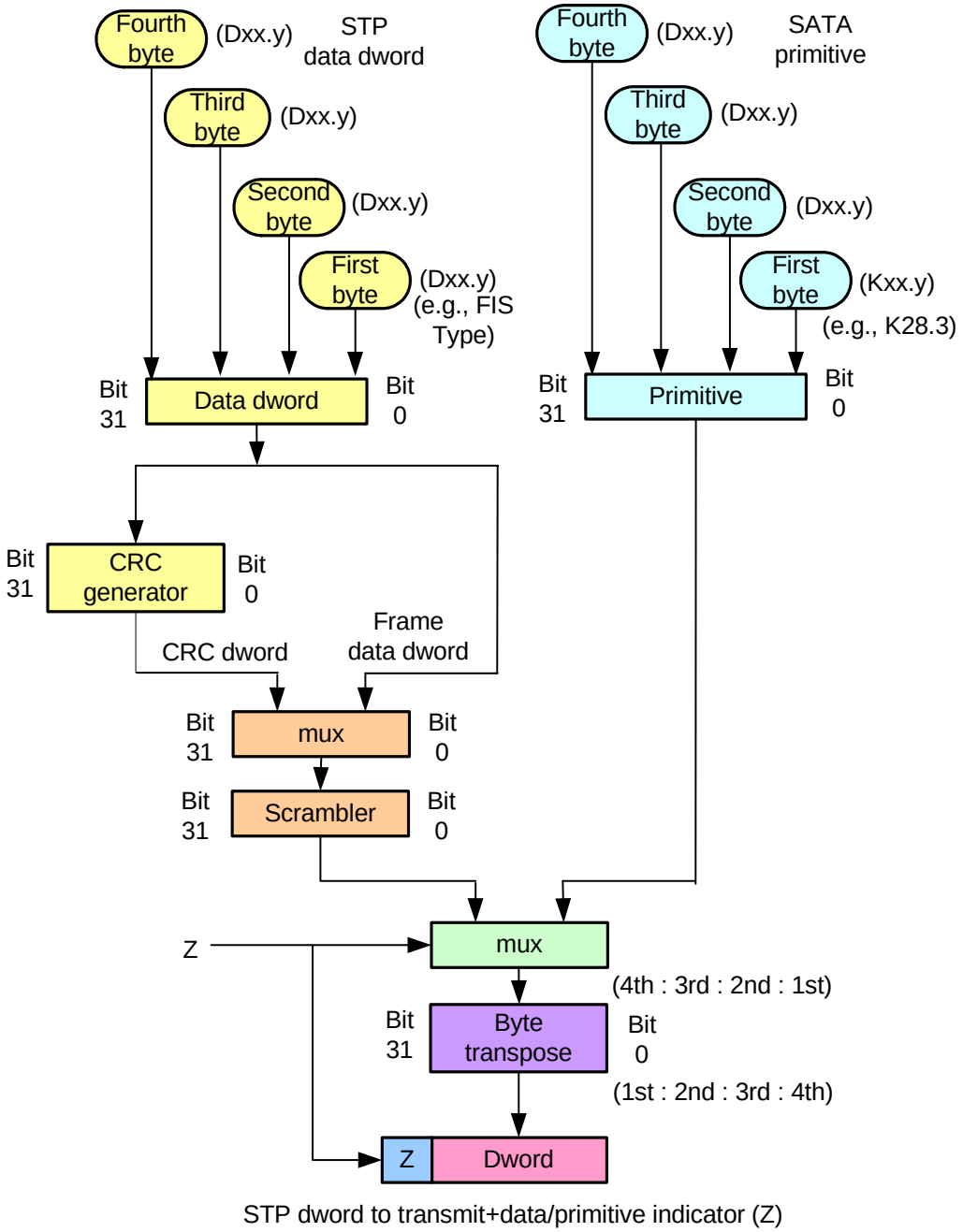


Figure 79 — STP transmit path bit ordering

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 80 shows the STP receive path bit ordering.

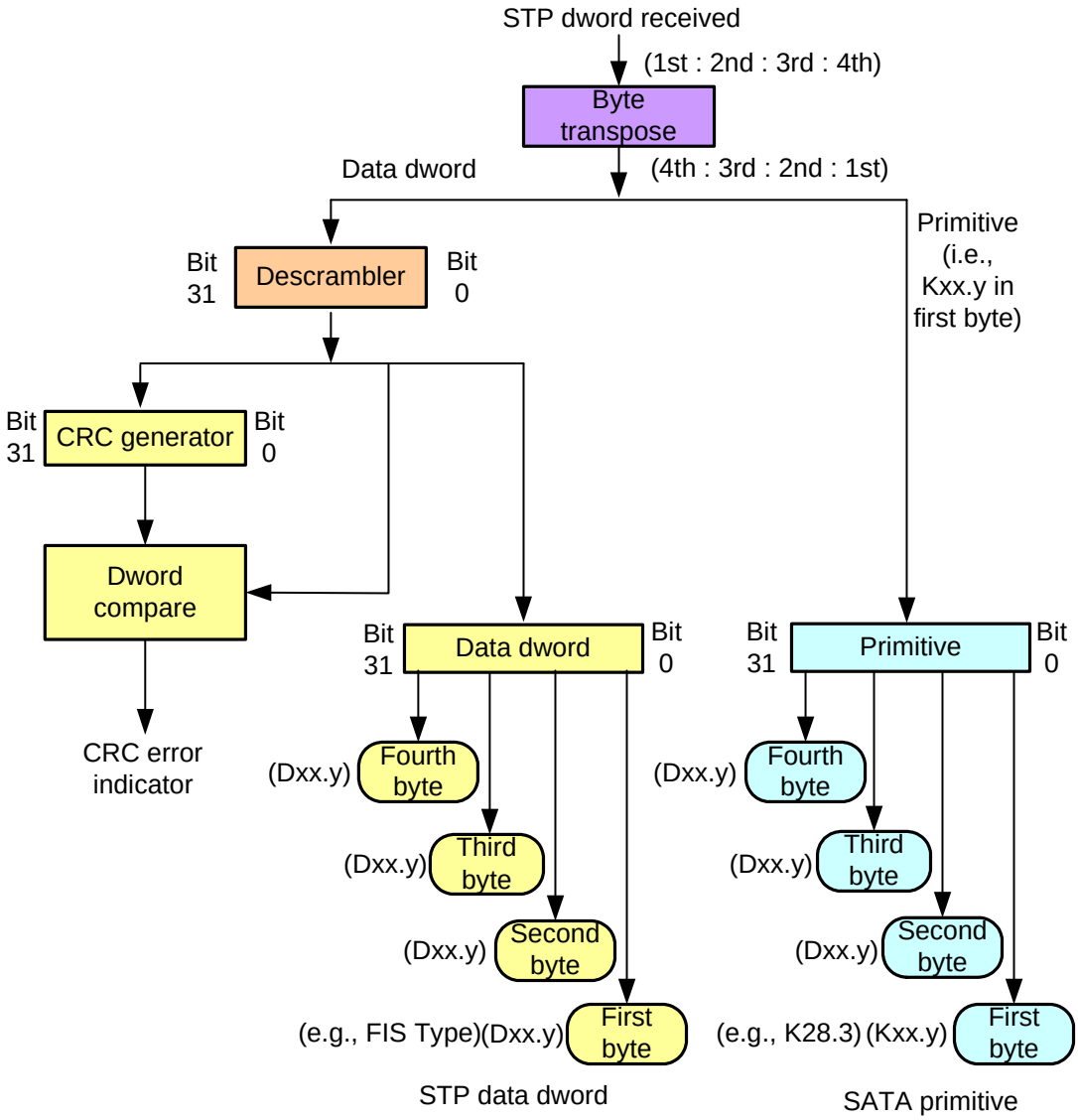


Figure 80 — STP receive path bit ordering

7.8 Address frames

7.8.1 Address frames overview

Address frames are used for the identification sequence and for connection requests. The address frame follows an SOAF and ends with an EOAF. Address frames shall only be sent outside connections. Address frames shall not be terminated early. All data dwords in an address frame shall be scrambled.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 67 defines the address frame format.

Table 67 — Address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0					ADDRESS FRAME TYPE			
1								
				Frame type dependent bytes				
27								
28	(MSB)							
				CRC				
31								(LSB)

The ADDRESS FRAME TYPE field indicates the type of address frame and is defined in table 68. This field determines the definition of the frame type dependent bytes.

Table 68 — Address frame types

Code	Frame type	Description
0h	Identify	Identification sequence
1h	Open	Connection request
All others	Reserved	

The CRC field contains a CRC value (see 7.5) 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.8.2 IDENTIFY address frame

Table 69 defines the IDENTIFY address frame format used for the identification sequence. The IDENTIFY address frame is sent after the phy reset sequence completes if the physical link is a SAS physical link.

Table 69 — IDENTIFY address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	Restricted (for OPEN address frame)	DEVICE TYPE			ADDRESS FRAME TYPE (0h)			
1	Restricted (for OPEN address frame)							
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	Restricted (for OPEN address frame)							
11								
12	SAS ADDRESS							
19								
20	PHY IDENTIFIER							
21	Reserved							
27								
28	(MSB)	CRC						
31								(LSB)

The DEVICE TYPE field indicates the type of device containing the phy, and is defined in table 70.

Table 70 — Device types

Code	Description
001b	End device
010b	Edge expander device
011b	Fanout expander device
All others	Reserved

The ADDRESS FRAME TYPE field shall be set to 0h.

--`,,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

An SSP INITIATOR PORT bit set to one indicates the presence of an SSP initiator port. An SSP INITIATOR PORT bit set to zero indicates 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 the presence of an STP initiator port. An STP INITIATOR PORT bit set to zero indicates 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 the presence of an SMP initiator port. An SMP INITIATOR PORT bit set to zero indicates 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 the presence of an SSP target port. An SSP TARGET PORT bit set to zero indicates 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 the presence of an STP target port. An STP TARGET PORT bit set to zero indicates 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 the presence of an SMP target port. An SMP TARGET PORT bit set to zero indicates an SMP target port is not present. Expander devices shall set the SMP TARGET PORT bit to one.

For SAS ports, the SAS ADDRESS field indicates the port identifier of the SAS port transmitting the IDENTIFY address frame. For expander ports, the SAS ADDRESS field indicates the device name of the expander device transmitting the IDENTIFY address frame.

The PHY IDENTIFIER field indicates the phy identifier of the phy transmitting the IDENTIFY address frame.

A wide port shall set the DEVICE TYPE field, SSP INITIATOR PORT bit, STP INITIATOR PORT bit, SMP INITIATOR PORT bit, SSP TARGET PORT bit, STP TARGET PORT bit, and SMP TARGET PORT bit to the same set of values on each phy in the wide port. Recipient ports need not check the consistency of these fields across phys.

The CRC field is defined in 7.8.1.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.8.3 OPEN address frame

Table 71 defines the OPEN address frame format used for connection requests.

Table 71 — 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						
3								(LSB)
4		DESTINATION SAS ADDRESS						
11								
12		SOURCE SAS ADDRESS						
19								
20	COMPATIBLE FEATURES							
21	PATHWAY BLOCKED COUNT							
22	(MSB)	ARBITRATION WAIT TIME						
23								(LSB)
24		MORE COMPATIBLE FEATURES						
27								
28	(MSB)	CRC						
31								(LSB)

An INITIATOR PORT bit set to one indicates the source port is acting as a SAS initiator port. An INITIATOR PORT bit set to zero indicates the source port is acting as a SAS target port. If a SAS target/initiator port sets the INITIATOR PORT bit to one, it shall operate only in its initiator role during the connection. If a target/initiator port sets the INITIATOR PORT bit to zero, it shall operate only in its target role during the connection.

If a SAS target/initiator 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 target/initiator 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The PROTOCOL field indicates the protocol for the connection being requested and is defined in table 72.

Table 72 — Protocol

Code	Description
000b	SMP
001b	SSP
010b	STP
All others	Reserved

The ADDRESS FRAME TYPE field shall be set to 1h.

The FEATURES field shall be set to zero.

The CONNECTION RATE field indicates the connection rate (see 4.1.10) being requested between the source and destination, and is defined in table 73.

Table 73 — Connection rate

Code	Description
8h	1,5 Gbps
9h	3,0 Gbps
All others	Reserved

When requesting a connection to a SAS target port, a SAS initiator port shall set the CONNECTION RATE field to a value supported by at least one potential pathway.

When requesting an SSP connection to an SSP initiator port, an SSP target port shall set the CONNECTION RATE field to the connection rate in effect when the command was received unless it has received an OPEN_REJECT (CONNECTION RATE NOT SUPPORTED). See 7.12.2.2 for details on handling OPEN_REJECT (CONNECTION RATE NOT SUPPORTED).

The SAS target port should send frames in a connection to the SAS initiator port regardless of whether the saved connection rate for that command matches the current connection rate; the SAS target port should not close the connection just to reopen the connection at the saved connection rate.

When requesting an STP connection to an STP initiator port, an STP target port shall set the CONNECTION RATE field to the last value received in a connection request from the STP initiator port unless the STP target port has received an OPEN_REJECT (CONNECTION RATE NOT SUPPORTED).

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. SSP or STP initiator ports shall set the INITIATOR CONNECTION TAG field to FFFFh if they do not require the field be provided by the SAS target port. If they do require the field to be provided, an SSP 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 use the same INITIATOR CONNECTION TAG field value for all connection requests to the same SAS target port, and shall only change the INITIATOR CONNECTION TAG field value when it has no commands 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 indicates the port identifier of the SAS port to which a connection is being requested.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The SOURCE SAS ADDRESS field indicates the port identifier of the SAS port that originated the OPEN address frame.

The COMPATIBLE FEATURES field shall be set to zero. The destination device shall ignore the COMPATIBLE FEATURES field.

The PATHWAY BLOCKED COUNT field indicates the number of times the port has retried this connection request due to receiving OPEN_REJECT (PATHWAY BLOCKED). The port shall not increment the PATHWAY BLOCKED COUNT value past FFh. If the port changes connection requests, it shall use a PATHWAY BLOCKED COUNT of 00h.

The ARBITRATION WAIT TIME field indicate how long the port transmitting the OPEN address frame has been waiting for a connection request to be accepted. This time is maintained by the port layer in an Arbitration Wait Time timer (see 8.2.2). 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 74 describes several values of the ARBITRATION WAIT TIME field. See 7.12.3 for details on arbitration fairness.

Table 74 — Arbitration wait time

Code	Description
0000h	0 µs
0001h	1 µs
...	...
7FFFh	32 767 µs
8000h	0 ms + 32 768 µs
8001h	1 ms + 32 768 µs
...	...
FFFFh	32 767 ms + 32 768 µs

The MORE COMPATIBLE FEATURES field shall be set to zero. The destination device shall ignore the MORE COMPATIBLE FEATURES field.

The CRC field is defined in 7.8.1.

7.9 Identification and hard reset sequence

7.9.1 Identification and hard reset sequence overview

After the phy reset sequence has been completed indicating the physical link is using SAS rather than SATA, each phy transmits either:

a) an IDENTIFY address frame (see 7.8.2); or
b) a HARD_RESET.

Each phy receives an IDENTIFY address frame or a HARD_RESET from the phy to which it is attached. The combination of a phy reset sequence, an optional hard reset sequence, and an identification sequence is called a link reset sequence (see 4.4.1).

If a device supports more than one phy, it shall transmit the same SAS address on all phys for which it is capable of sharing within a port.

If a device detects the same SAS address incoming on different phys, it shall consider those phys part of the same wide port.

If a device detects different SAS addresses incoming on different physical links, it shall consider those physical links as independent physical links and consider those phys part of different ports.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If a device does not receive a valid IDENTIFY address frame within 1 ms of phy reset sequence completion, it shall restart the phy reset sequence.

If a device receives an additional IDENTIFY address frame after receiving the first one, without an intervening phy reset sequence, it shall ignore the additional IDENTIFY address frame.

If a phy receives a HARD_RESET, it shall be considered a reset event and cause a hard reset (see 4.4.2) of the port containing that phy.

7.9.2 SAS initiator device rules

After a link reset sequence, or after receiving a BROADCAST (CHANGE), a management application client behind an SMP initiator port should perform a discover process (see 4.6.7.4).

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.

The discover information may be used to select connection rates for connection requests.

7.9.3 Fanout expander device rules

After completing the identification sequence on a phy and completing internal initialization, the ECM within a fanout 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 for connection requests.

After a link reset sequence, or after receiving a BROADCAST (CHANGE), the management application client behind an SMP initiator port in a fanout expander device that does not have a configurable expander route table shall follow the SAS initiator device rules (see 7.9.2) to perform a discover process.

The ECM of a fanout expander device that has a configurable expander route table is dependent on the completion of the discover process (see 4.6.7.4) for routing connection requests using the table routing method.

7.9.4 Edge expander device rules

After completing the identification sequence on a phy and completing internal initialization, the ECM within an edge 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 for connection requests.

The ECM of an edge expander device that has a configurable expander route table is dependent on the completion of the discover process (see 4.6.7.4) for routing connection requests using the table routing method.

7.9.5 SL_IR (link layer identification and hard reset) state machines

7.9.5.1 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) state machine (see 7.9.5.3);
b) SL_IR_RIF (receive IDENTIFY address frame) state machine (see 7.9.5.4); and
c) SL_IR_IRC (identification and hard reset control) state machine (see 7.9.5.5).

The SL_IR state machines send the following messages to the SL state machines (see 7.14) in SAS devices or the XL (see 7.15) state machine in expander devices:

a) Enable Disable SAS Link (Enable); and
b) Enable Disable SAS Link (Disable).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The SL_IR state machines shall maintain the timers listed in table 75.

Table 75 — SL_IR timers

Timer	Initial value
Receive Identify Timeout timer	1 ms

--`,,,`,,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 81 shows the SL_IR state machines.

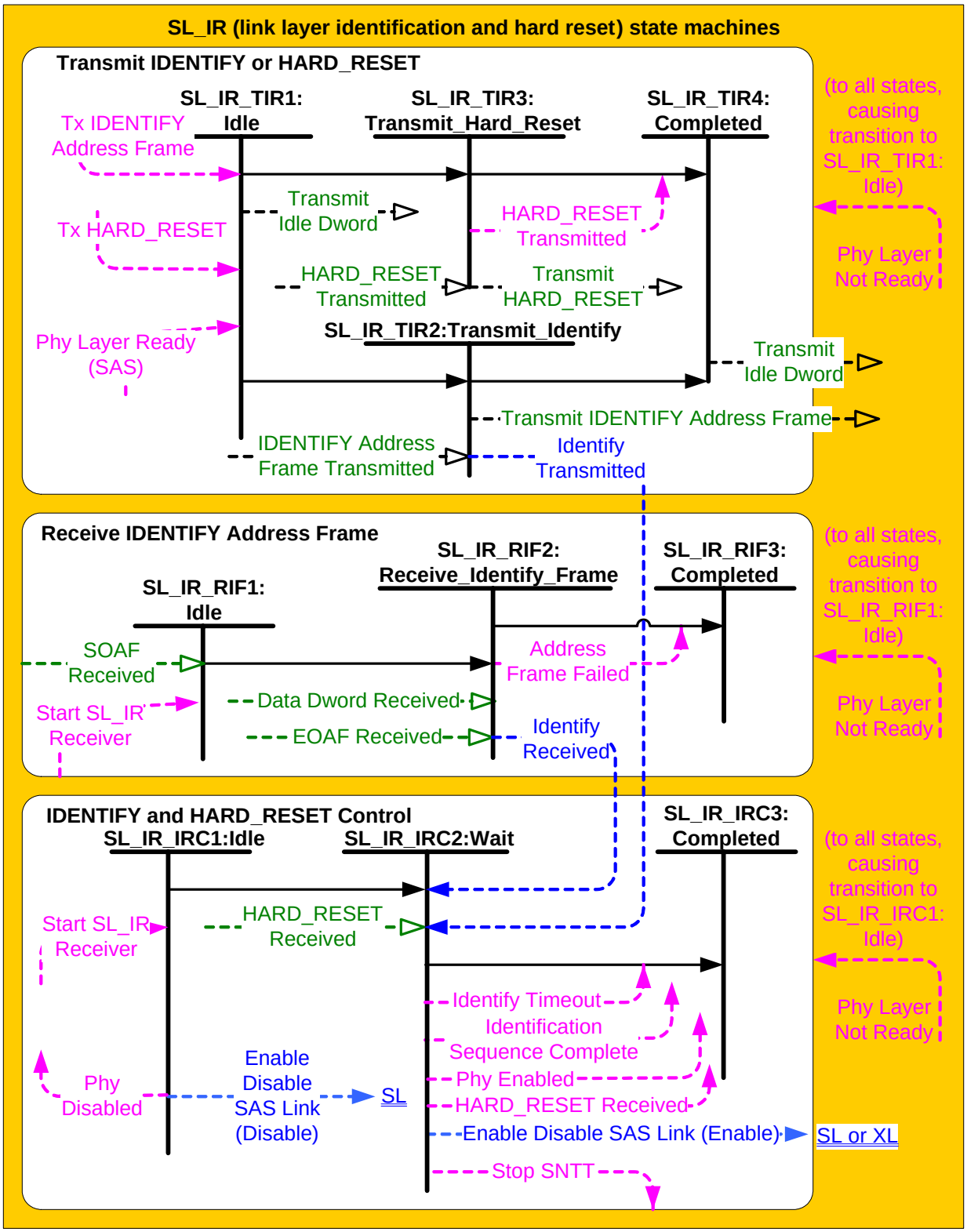


Figure 81 — SL_IR (link layer identification and hard reset) state machines

--`,,`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.9.5.2 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.

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:

a) SOAF Received;
b) Data Dword Received;
c) EOAF Received; and
d) HARD_RESET Received.

The SL_IR receiver shall ignore all other dwords.

7.9.5.3 SL_IR_TIR (transmit IDENTIFY or HARD_RESET) state machine

7.9.5.3.1 SL_IR_TIR state machine overview

The SL_IR_TIR state machine's function is to transmit a single IDENTIFY address frame or HARD_RESET after the phy layer enables the link layer. This state machine consists of the following states:

a) SL_IR_TIR1:Idle (see 7.9.5.3.2)(initial state);
b) SL_IR_TIR2:Transmit_Identify (see 7.9.5.3.3);
c) SL_IR_TIR3:Transmit_Hard_Reset (see 7.9.5.3.4); and
d) SL_IR_TIR4:Completed (see 7.9.5.3.5).

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.

7.9.5.3.2 SL_IR_TIR1:Idle state

7.9.5.3.2.1 State description

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IR transmitter.

7.9.5.3.2.2 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 Tx IDENTIFY Address Frame request is received.

7.9.5.3.2.3 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 Tx HARD_RESET request is received.

7.9.5.3.3 SL_IR_TIR2:Transmit_Identify state

7.9.5.3.3.1 State description

Upon entry into this state, this state shall send a Transmit IDENTIFY Address Frame message to the SL_IR transmitter.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

After this state receives an IDENTIFY Address Frame Transmitted message, this state shall send an Identify Transmitted message to the SL_IR_IRC state machine.

7.9.5.3.3.2 Transition SL_IR_TIR2:Transmit_Identify to SL_IR_TIR4:Completed

This transition shall occur after sending an Identify Transmitted message to the SL_IR_IRC state machine.

7.9.5.3.4 SL_IR_TIR3:Transmit_Hard_Reset state

7.9.5.3.4.1 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.

7.9.5.3.4.2 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.

7.9.5.3.5 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.

7.9.5.4 SL_IR_RIF (receive IDENTIFY address frame) state machine

7.9.5.4.1 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 7.9.5.4.2)(initial state);
b) SL_IR_RIF2:Receive_Identify_Frame (see 7.9.5.4.3); and
c) SL_IR_RIF3:Completed (see 7.9.5.4.4).

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.

7.9.5.4.2 SL_IR_RIF1:Idle state

7.9.5.4.2.1 State description

This state waits for an SOAF to be received from the physical link, indicating an address frame is arriving.

7.9.5.4.2.2 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.

7.9.5.4.3 SL_IR_RIF2:Receive_Identify_Frame state

7.9.5.4.3.1 State description

This state receives the dwords of an address frame and the EOAF.

This state shall ignore all primitives (e.g., BREAK and HARD_RESET) except SOAF.

--`,,,``,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If this state receives an SOAF, 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 IDENTIFY address frame was received.

If this state receives more than eight data dwords after an SOAF and before an EOAF, 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 IDENTIFY address frame was received.

After receiving an EOAF, this state shall check if it the IDENTIFY address frame is valid.

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 Identify;
b) the number of bytes between the SOAF and EOAF is 32; and
c) the CRC field contains a valid CRC.

Otherwise, this state shall discard the IDENTIFY address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

7.9.5.4.3.2 Transition SL_IR_RIF2:Receive_Identify_Frame to SL_IR_RIF3:Completed

This transition shall occur after sending an Identify Received message or Address Frame Failed confirmation.

7.9.5.4.4 SL_IR_RIF3:Completed state

This state waits for a Phy Layer Not Ready confirmation.

7.9.5.5 SL_IR_IRC (identification and hard reset control) state machine

7.9.5.5.1 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 is received before an IDENTIFY address frame has been received.

This state machine consists of the following states:

a) SL_IR_IRC1:Idle (see 7.9.5.5.2)(initial state);
b) SL_IR_IRC2:Wait (see 7.9.5.5.3); and
c) SL_IR_IRC3:Completed (see 7.9.5.5.4).

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.

7.9.5.5.2 SL_IR_IRC1:Idle state

7.9.5.5.2.1 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 SL_CC state machines (see 7.14) or XL state machine (see 7.15) 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.

7.9.5.5.2.2 Transition SL_IR_IRC1:Idle to SL_IR_IRC2:Wait

This transition shall occur after a Start SL_IR Receiver confirmation is received.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.9.5.5.3 SL_IR_IRC2:Wait state

7.9.5.5.3.1 State description

This state ensures that an IDENTIFY address frame has been received by the SL_IR_RIF state machine and that a 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.

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, 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 7.14) in a SAS phy or the XL state machine (see 7.15) in an expander 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, 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 before an Identify Received message is received, this state shall send a HARD_RESET Received confirmation to the port layer and a Stop SNTT request to the phy layer.

If this state receives a HARD_RESET after an Identify Received message is received, the HARD_RESET shall be ignored.

7.9.5.5.3.2 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 an Phy Enabled confirmation.

7.9.5.5.4 SL_IR_IRC3:Completed state

This state waits for a Phy Layer Not Ready confirmation.

7.10 Power management

SATA interface power management is not supported in SAS.

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, it shall reply with SATA_PMNAK.

If an expander device receives SATA_PMREQ_P or SATA_PMREQ_S from a SATA device while an STP connection is not open, it shall not forward it to any STP initiator port and shall reply with a SATA_PMNAK. If one of these primitives arrives while an STP connection is open, it may forward the primitive to the STP initiator port.

SCSI idle and standby power conditions, implemented with the START STOP UNIT command (see SBC-2) and the Power Condition mode page (see SPC-3), may be supported by SSP initiator ports and SSP target ports as described in 10.2.8.

ATA idle and standby power modes, implemented with the IDLE, IDLE IMMEDIATE, STANDBY, STANDBY IMMEDIATE, and CHECK POWER MODE commands (see ATA/ATAPI-7 V1), may be supported by STP initiator ports. The ATA sleep power mode, implemented with the SLEEP command, shall not be used.

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.11 SAS domain changes

After power on or receiving BROADCAST (CHANGE), an application client in each SAS initiator port should scan the SAS domain using the discover process (see 4.6.7.4) to search for SAS initiator devices, SAS target devices, and expander devices.

The expander device shall transmit BROADCAST (CHANGE) from at least one phy in each expander port other than the expander port that is the cause for transmitting BROADCAST (CHANGE).

Expander devices shall transmit BROADCAST (CHANGE) for the following reasons:

a) after an expander phy has lost dword synchronization;
b) after the link reset sequence completes; and
c) after the expander device receives BROADCAST (CHANGE).

BROADCAST (CHANGE) may be sent by SAS initiator ports to force other SAS initiator ports and expander ports to re-run the discover process, but should not be sent by SAS target ports.

A SAS initiator port that detects BROADCAST (CHANGE) shall follow the SAS initiator device rules (see 7.9.2) to discover the topology.

A fanout expander device that detects BROADCAST (CHANGE) shall follow the fanout device rules (see 7.9.3) to discover the topology.

An edge expander device that detects BROADCAST (CHANGE) shall follow the edge device rules (see 7.9.4).

See 10.4.3.3 for details on counting BROADCAST (CHANGE) generation in an expander device.

7.12 Connections

7.12.1 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, or transfer data. SSP target ports open SSP connections to transfer data or transmit status.

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. AIP, OPEN_ACCEPT and OPEN_REJECT are the responses to an OPEN address frame. BREAK is used to abort connection requests and to unilaterally break a connection. CLOSE is used for orderly closing a connection.

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 in an expander device attached to a SATA device. The SATA device is not aware of SAS connection management.

A wide port may have separate connections on each of its phys.

7.12.2 Opening a connection

7.12.2.1 Connection request

The OPEN address frame (see 7.8.3) is used to open a connection from a source port to a destination port using one source phy and one destination phy.

To make a connection request, the source port shall transmit an OPEN address frame through an available phy. The source phy shall transmit idle dwords after the OPEN address frame until it receives a response or abort the connection request with BREAK.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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, the source phy may assume the destination port does not exist and shall transmit BREAK to abort the connection request.

The OPEN address frame flows through expander devices onto intermediate physical links. If an expander device on the pathway is unable to forward the connect request because none of the prospective physical links support the requested connection rate, the expander device shall return OPEN_REJECT (CONNECTION RATE NOT SUPPORTED). If the OPEN address frame reaches the destination, it shall return either OPEN_ACCEPT or OPEN_REJECT. Rate matching shall be used on any physical links in the pathway with negotiated physical link rates that are faster than the requested connection rate (see 7.13).

7.12.2.2 Connection responses

Table 76 lists the responses to an OPEN address frame being transmitted.

Table 76 — Connection responses

Response	Description
AIP	Arbitration in progress. When an expander device is trying to open a connection to the selected destination port, it returns an AIP to the source phy. The source phy shall reinitialize and restart its Open Timeout timer when it receives an AIP. AIP is sent by an expander device while it is internally arbitrating for access to an expander port.
OPEN_ACCEPT	Connection request accepted. This is sent by the destination phy.
OPEN_REJECT	Connection request rejected. This is sent in response by the destination phy or by an expander device. The different versions are described in 7.2.5.11.
OPEN address frame	If AIP has been previously detected, this indicates an overriding connection request. If AIP has not yet been detected, this indicates two connection requests crossing on the physical link. Arbitration fairness determines which one wins (see 7.12.3).
BREAK	The destination port or expander port may reply with BREAK indicating the connection is not being established.
Open Timeout timer expires	The source phy shall abort the connection request by transmitting BREAK (see 7.12.6).

After an OPEN_REJECT (CONNECTION RATE NOT SUPPORTED) has been received by a SAS target port, the SAS target device shall set the connection rate for future requests for that I_T_L_Q nexus to:

a) the last value received in a connection request from the SAS initiator port;
b) 1,5 Gbps; or
c) the connection rate in effect when the command was received.

7.12.3 Arbitration fairness

SAS supports least-recently used arbitration fairness.

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 its request is granted. The Arbitration Wait Time timer shall count in microseconds from 0 μs to 32 767 μs and in milliseconds from 32 768 μs to 32 767 ms + 32 768 μs. The Arbitration Wait Time timer shall stop incrementing when its value reaches 32 767 ms + 32 768 μs.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SAS ports (i.e., SAS initiator ports and SAS target ports) shall start the Arbitration Wait Time timer (see 8.2.2) when they transmit the first OPEN address frame (see 7.8.3) 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.

SAS ports should set the Arbitration Wait Time timer to zero when they transmit 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 7.8.3) to a higher value than its Arbitration Wait Time timer indicates. However, unfair SAS ports shall not set the ARBITRATION WAIT TIME field to a value greater than or equal to 8000h; this limits the amount of unfairness and helps prevent livelocks.

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 port shall stop the Arbitration Wait Time timer and set it to zero when it wins arbitration (i.e., it receives either OPEN_ACCEPT or OPEN_REJECT from the destination SAS port rather than from an intermediate expander device). A port shall stop the Arbitration Wait Time timer when it loses arbitration to a connection request that satisfies its arbitration request (i.e., it receives an OPEN address frame from the destination SAS port with the INITIATOR PORT bit set to the opposite value and a matching PROTOCOL field).

When arbitrating for access to an outgoing expander port, the expander device shall select the connection request from the expander port with the largest Arbitration Wait Time timer value. If the largest Arbitration Wait timer values are identical, then the connection request with the largest SOURCE SAS ADDRESS shall win arbitration.

If two connection requests pass on a physical link, the winner shall be determined by comparing OPEN address frame field contents using the arbitration priority described in table 77.

Table 77 — Arbitration priority for OPENs passing on a physical link

Bits 79-64 (79 is MSB)	Bits 63-0 (0 is LSB)
ARBITRATION WAIT TIME field value	SOURCE SAS ADDRESS field value

See 7.8.3 for details on the OPEN address frame and the ARBITRATION WAIT TIME field.

7.12.4 Arbitration and resource management in an expander device

7.12.4.1 Arbitration overview

The ECM shall arbitrate and assign or deny path resources for connection attempts requested by each expander phy in response to receiving valid OPEN address frames.

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 responds to connection requests by returning arbitration won, lost, and reject confirmations to the requesting expander phy.

Each path request contains the Arbitration Wait Time and the Source SAS Address arguments from the received OPEN address frame.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If two or more path requests contend, the winner shall be selected by comparing the OPEN address frame contents using the arbitration priority described in table 78.

Table 78 — Arbitration priority for contending path requests in the ECM

Bits 83-68 (83 is MSB)	Bits 67-5	Bits 3-0 (0 is LSB)
ARBITRATION WAIT TIME field value	SOURCE SAS ADDRESS field value	CONNECTION RATE field value

The ECM shall generate the arbitration reject confirmation when any of the following conditions are met:

a) the connection request does not map to a valid expander phy;
b) the connection request specifies an unsupported connection rate; or
c) the connection request specifies a destination port that contains at least one partial pathway and pathway recovery rules require this connection request to release path resources.

The ECM shall generate the arbitration lost confirmation when all of the following conditions are met:

a) the connection request maps to an available expander phy at a supported connection rate; and
b) the destination expander phy of this connection request has received a higher priority OPEN address frame with this expander phy as its destination (i.e., when two expander phys both receive an OPEN address frame destined for each other, the ECM shall provide arbitration lost confirmation to the expander phy that received the lowest priority OPEN address frame).

The ECM shall generate the arbitration won confirmation when all of the following conditions are met:

a) the connection request maps to an available expander phy at a supported connection rate; and
b) no higher priority connection requests are present with this expander phy as the destination.

7.12.4.2 Arbitration status

Arbitration status shall be conveyed between expander devices and by expander devices to SAS endpoints using AIP primitives. 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 phy is set to the last type of AIP received.

7.12.4.3 Partial Pathway Timeout timer

Each expander 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 phy shall initialize the Partial Pathway Timeout timer to the partial pathway timeout value it reports in the SMP DISCOVER function (see 10.4.3.5) and run the Partial Pathway Timeout timer whenever the ECM provides confirmation to an expander phy that all expander phys within the requested destination port are blocked waiting on partial pathways.

NOTE 18 - The partial pathway timeout value allows flexibility in specifying how long an expander device waits before attempting pathway recovery. The recommended default value (see 10.4.3.5) 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 it retries 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 phy shall initialize and start the Partial Pathway Timeout timer when all of the following conditions are met:

a) there are no unallocated expander phys within a requested destination port available to complete the connection; and
b) at least one expander phy within the requested destination port contains a blocked partial pathway.

When one of the conditions above are not met, the expander phy shall stop the Partial Pathway Timeout timer. If the timer expires, pathway recovery shall occur (see 7.12.4.4).

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.12.4.4 Pathway recovery

Pathway recovery provides a means to abort connection requests in order to prevent deadlock using pathway recovery priority comparisons. Pathway recovery priority compares the OPEN address frames of the blocked connection requests as described in table 79.

Table 79 — Pathway recovery priority

Bits 75-68 (75 is MSB)	Bits 67-5	Bits 3-0 (0 is LSB)
PATHWAY BLOCKED COUNT field value	SOURCE SAS ADDRESS field value	CONNECTION RATE field value

When the Partial Pathway Timeout timer for an arbitrating expander 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 instruct the arbitrating expander phy to reject the connection request by transmitting OPEN_REJECT (PATHWAY_BLOCKED) when the Partial Pathway Timeout timer expires and the pathway recovery priority of the arbitrating expander phy (i.e., the expander phy requesting the connection) is less than the pathway recovery priority of all expander phys within the destination port with an arbitration status of WAITING_ON_PARTIAL.

7.12.5 Expander devices and connection requests

7.12.5.1 All expander devices

Before an expander device transmits AIP, it may have transmitted an OPEN address frame on the same physical link. Arbitration fairness dictates which OPEN address frame wins (see 7.12.3).

After an expander device transmits an AIP, it shall not transmit an OPEN address frame unless it has higher arbitration priority than the incoming connection request.

Expander devices shall transmit no more than three consecutive AIPs without transmitting an idle dword. Expander devices shall transmit at least one AIP every 128 dwords.

Expander devices shall transmit an AIP within 128 dwords of receiving an OPEN address frame.

7.12.5.2 Edge expander devices

When an edge expander device receives a connection request, it shall compare the destination SAS address to the SAS addresses of the devices to which each of its phys is attached. For all phys which have table routing attributes (see 4.6.7.1) and are attached to edge expander devices, it shall compare the destination SAS address to all the enabled routed SAS addresses in the expander route table.

If it finds a match in one or more phys, then the expander device shall arbitrate for access to one of the matching phys and forward the connection request.

If it does not find a match, but at least one phy has the subtractive routing attribute and is attached to an expander device (either an edge expander device or a fanout expander device), and the request did not come from that expander device, the connection request shall be forwarded to the expander device through any of the subtractive routing phys.

If it does not find a match and no subtractive routing phy is available, the edge expander device shall reply with OPEN_REJECT (NO DESTINATION).

If the destination phy is in the same expander port as the source phy and the source phys are using the subtractive routing method, the edge expander device shall reply with OPEN_REJECT (NO DESTINATION). If the source phys are not using subtractive routing, the edge expander device shall reply with OPEN_REJECT (BAD DESTINATION).

7.12.5.3 Fanout expander devices

When a fanout expander device receives a connection request, it shall compare the destination SAS address to the SAS addresses of the devices to which each of its phys is attached. For all phys that are attached to

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

edge expander devices, the fanout expander device shall compare the destination SAS addresses to all the enabled SAS addresses in the expander route table.

If the fanout expander device finds a match in one or more phys, it shall arbitrate for access to one of the matching phys and forward the connection request.

If the fanout expander device does not find a match, it shall reply with OPEN_REJECT (NO DESTINATION). If the destination phy is in the same expander port as the source phy, it shall reply with OPEN_REJECT (BAD DESTINATION).

7.12.6 Aborting a connection request

BREAK may be used to abort a connection request. The source phy shall transmit a BREAK after the Open Timeout timer expires or if it chooses to abort its request for any other reason.

After transmitting BREAK, the source phy shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer.

Table 80 lists the responses to a BREAK being transmitted before a connection response has been received.

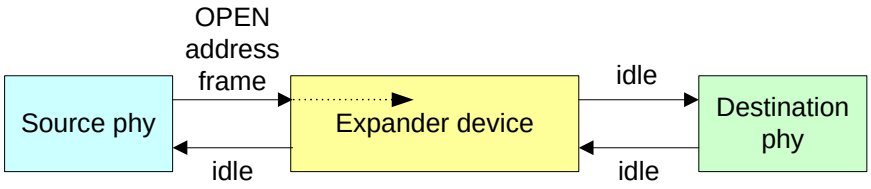
Table 80 — Abort connection responses

Response	Description
BREAK	This confirms that the connection request has been aborted.
Break Timeout timer expires	The originating phy shall assume the connection request has been aborted.

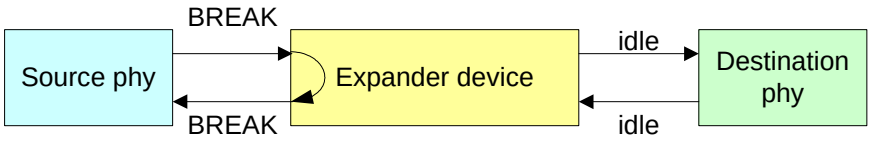
When a phy sourcing a BREAK is attached to an expander device, the BREAK response to the source phy is generated by the expander phy to which the source phy is attached, not the other SAS phy in the connection. If the expander device has transmitted a connection request to the destination, it shall also transmit BREAK to the destination. If the expander device has not transmitted a connection request to the destination, it shall not transmit BREAK to the destination. After transmitting BREAK back to the originating phy, the expander device shall ensure that an open response does not occur (i.e., the expander device shall not forward dwords from the destination any more). Figure 82 shows an example of BREAK usage.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

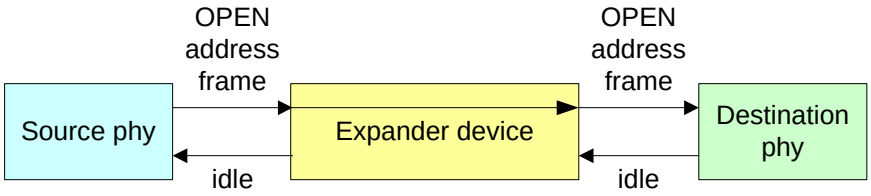
Case 1: OPEN address frame has not propogated through the expander device:



Case 1 result: BREAK only on Source device physical link



Case 2: OPEN address frame has propogated through the expander device:



Case 2 result: BREAK on Source device's physical link, then on destination device's physical link

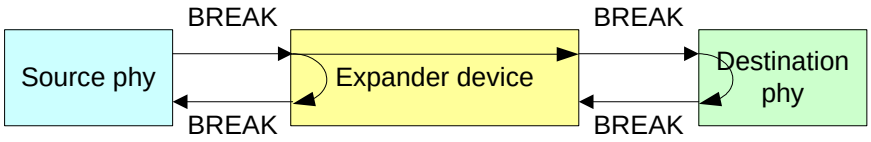


Figure 82 — Aborting a connection request with BREAK

Figure 83 shows the sequence for a connection request where the Open Timeout timer expires.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

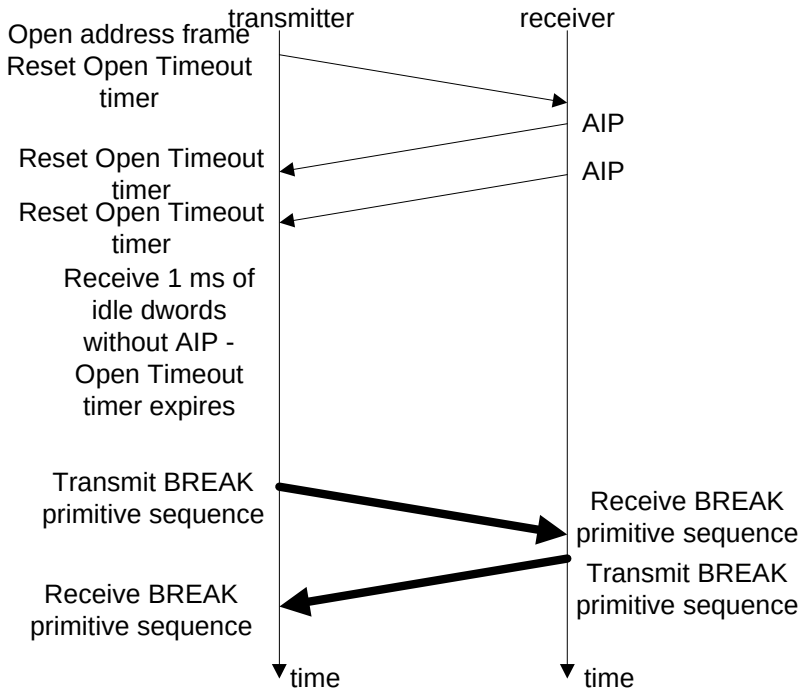


Figure 83 — Connection request timeout example

7.12.7 Closing a connection

CLOSE is used to close a connection of any protocol. See 7.16.6 for details on closing SSP connections, 7.17.5 for details on closing STP connections, and 7.18.3 for details on closing SMP connections.

After transmitting CLOSE, the source phy shall initialize a Close Timeout timer to 1 ms and start the Close Timeout timer.

Table 81 lists the responses to a CLOSE being transmitted.

Table 81 — Close connection responses

Response	Description
CLOSE	This confirms that the connection has been closed.
Close Timeout timer expires	The originating phy shall attempt to break the connection (see 7.12.8).

No additional dwords for the connection shall follow the CLOSE. Expander devices shall close the full-duplex connection upon detecting a CLOSE in each direction.

When a phy has both transmitted and received CLOSE, it shall consider the connection closed.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 84 shows example sequences for closing a connection.

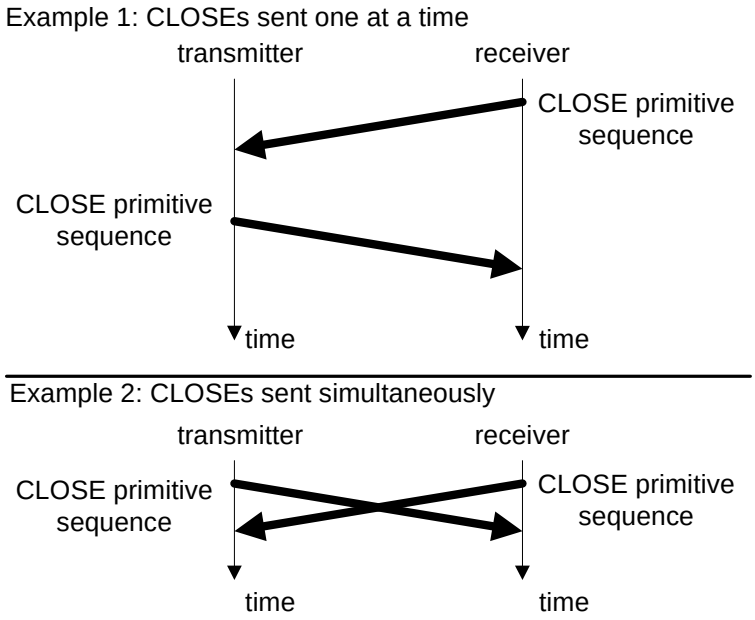


Figure 84 — Closing a connection example

7.12.8 Breaking a connection

In addition to aborting a connection request, BREAK may also be used to break a connection, in cases where CLOSE is not available. After transmitting BREAK, the originating phy shall ignore all incoming dwords except for BREAKs.

After transmitting BREAK, the source phy shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer.

Table 82 lists the responses to a BREAK being transmitted after a connection has been established.

Table 82 — Break connection responses

Response	Description
BREAK	This confirms that the connection has been broken.
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 due to loss of dword synchronization (see 6.8).

In addition to the actions described in this subclause and in 7.12.6, the following shall be the responses by an SSP phy to a broken connection:

a) 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;
b) Transmitted frames for which an ACK has been received prior to a broken connection shall be considered successfully transmitted; and
c) Transmitted frames for which an ACK or NAK has not been received prior to a broken connection shall be considered not successfully transmitted.

7.13 Rate matching

Each successful connection request contains the connection rate (see 4.1.10) of the pathway.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Every phy in the physical link shall insert ALIGNs or NOTIFYs between dwords to match the connection rate. Phys receiving ALIGNs and NOTIFYs delete them regardless of whether the ALIGNs and NOTIFYs were inserted for clock skew management (see 7.3) or for rate matching.

The faster phy shall rotate between ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) to reduce long strings of repeated patterns appearing on the physical link. NOTIFYs may be used to replace ALIGNs (see 7.2.5.9).

Figure 85 shows an example of rate matching between a 3,0 Gbps source phy and a 3,0 Gbps destination phy, with an intermediate 1,5 Gbps physical link in between.

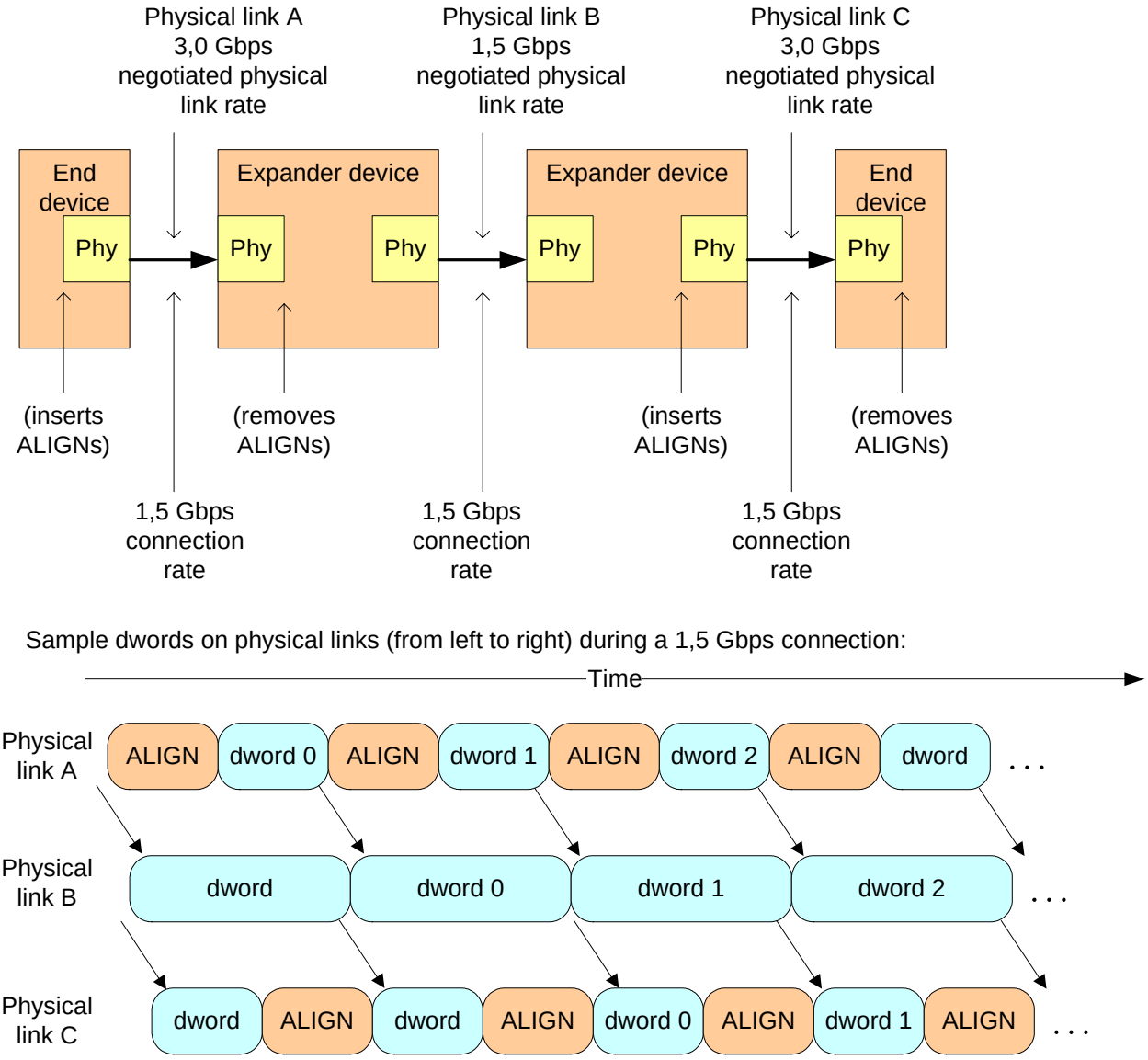


Figure 85 — Rate matching example

A phy shall start inserting ALIGNs and NOTIFYs for rate matching at the selected connection rate with the first dword following:

a) transmitting the EOAF for an OPEN address frame; or
b) transmitting an OPEN_ACCEPT.

The source phy transmits idle dwords including ALIGNs and NOTIFYs 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A phy shall stop inserting ALIGNs and NOTIFYs for rate matching after:

a) transmitting the first dword in a CLOSE;
b) transmitting the first dword in a BREAK;
c) receiving an OPEN_REJECT for a connection request; or
d) losing arbitration to a received OPEN address frame.

If an STP initiator port discovers a SATA device behind an STP/SATA bridge with a physical link rate greater than the maximum connection rate supported by the pathway from the STP initiator port, the STP initiator port should use the SMP PHY CONTROL function (see 10.4.3.10) to set the MAXIMUM PHYSICAL LINK RATE field of the expander phy attached to the SATA device to the maximum connection rate supported by the pathway.

7.14 SL (link layer for SAS phys) state machines

7.14.1 SL state machines overview

The SL (link layer for SAS phys) state machines controls connections, handling both connection requests (OPEN address frames), CLOSEs, and BREAKs. The SL state machines are as follows:

a) SL_RA (receive OPEN address frame) state machine (see 7.14.3); and
b) SL_CC (connection control) state machine (see 7.14.4).

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 the initial state is as indicated in the state machine description.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 86 shows the SL state machines.

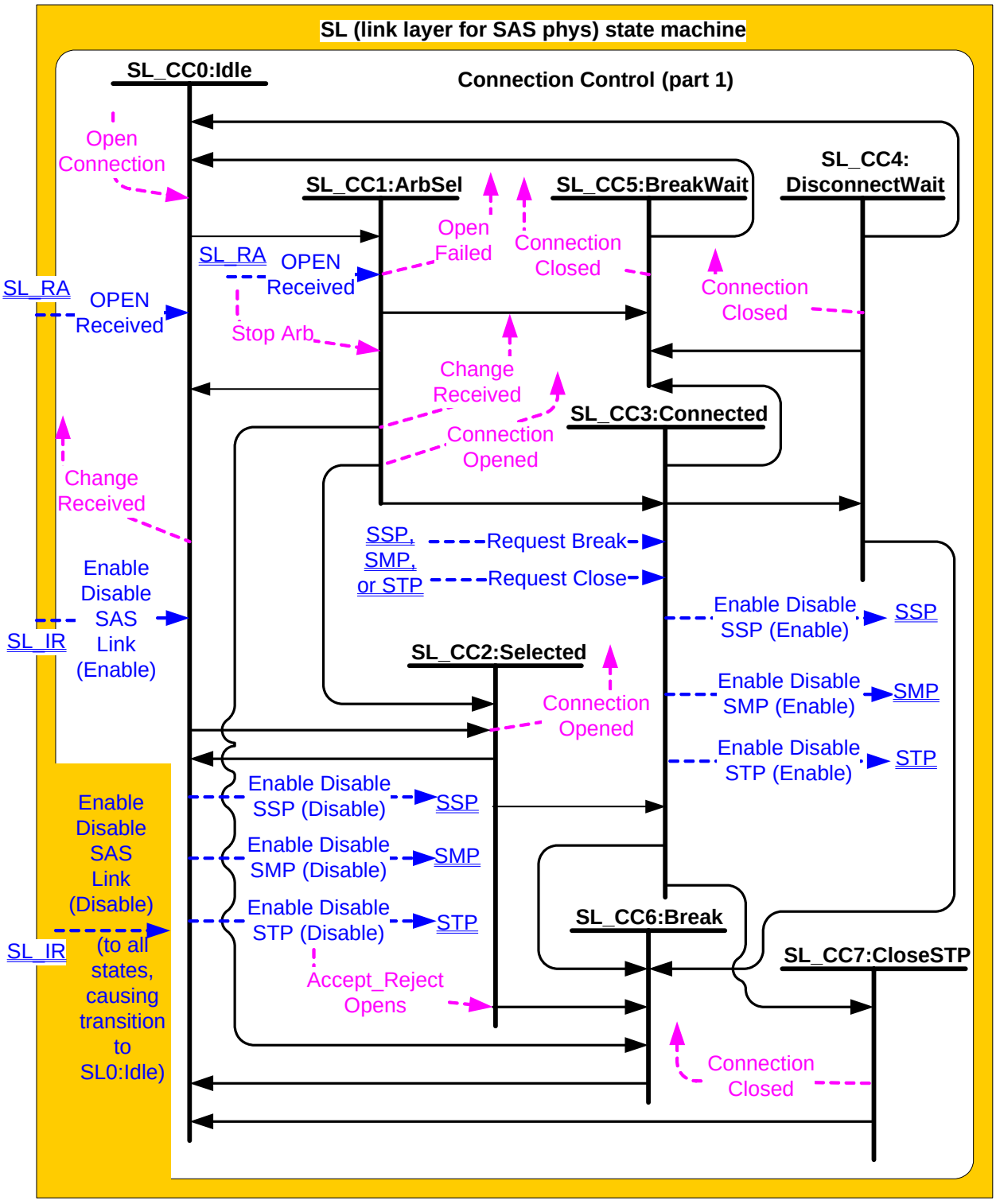


Figure 86 — SL (link layer for SAS phys) state machines (part 1)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 87 shows the messages sent to the SL transmitter and received from the SL receiver.

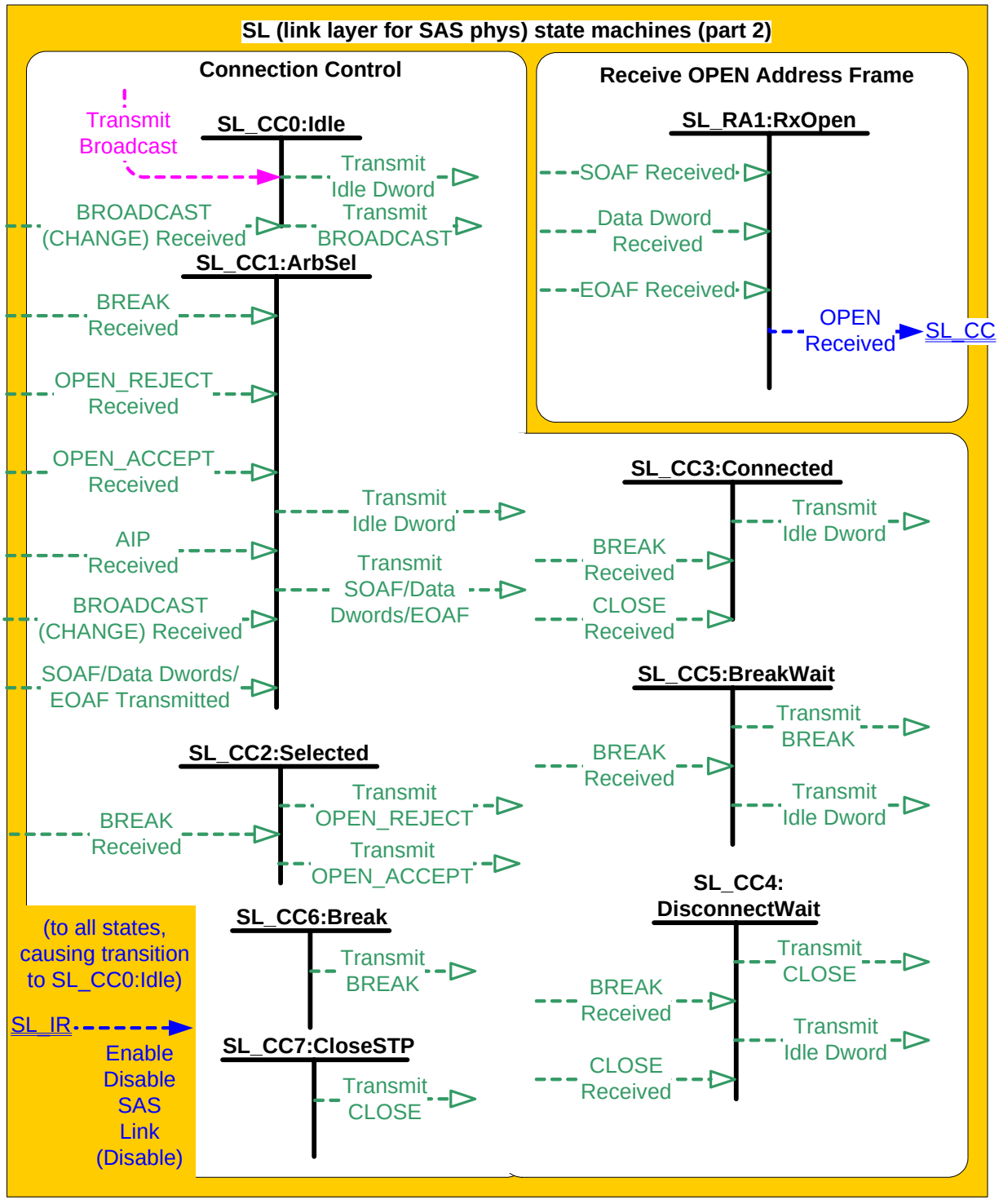


Figure 87 — SL (link layer for SAS phys) state machines (part 2)

7.14.2 SL transmitter and receiver

The SL transmitter receives the following messages from the SL state machines:

a) Transmit Idle Dword;
b) Transmit SOAF/Data Dwords/EOAF;
c) Transmit OPEN_ACCEPT;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

d) Transmit OPEN_REJECT with an argument indicating the specific type (e.g., Transmit OPEN_REJECT (Retry));
e) Transmit BREAK;
f) Transmit BROADCAST; and
g) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal)).

The SL transmitter sends the following messages from the SL state machines:

a) SOAF/Data Dwords/EOAF Transmitted.

The SL receiver sends the following messages to the SL state machines:

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) OPEN_ACCEPT Received;
g) OPEN_REJECT Received with an argument indicating the specific type (e.g., OPEN_REJECT Received (No Destination));
h) AIP Received; and
i) CLOSE Received with an argument indicating the specific type (e.g., CLOSE Received (Normal)).

When the SL transmitter is requested to transmit a dword from any state within any of the SL state machines, it shall transmit that dword. If there are multiple requests to transmit, the following priority should be followed when selecting the dword to transmit:

1) BREAK;
2) CLOSE;
3) OPEN_ACCEPT or OPEN_REJECT;
4) SOAF or data dword or EOAF; then
5) idle dword.

7.14.3 SL_RA (receive OPEN address frame) state machine

The SL_RA state machine's function is to receive address frames and determine if the received address frame is an OPEN address frame and whether or not it was received successfully. 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 primitives except SOAF and EOAF.

If this state machine receives a subsequent SOAF after receiving an SOAF but before receiving an EOAF, then this state machine shall discard the data dwords received before the subsequent SOAF.

If this state machine receives more than eight data dwords after an SOAF and before an EOAF, then this state machine shall discard the address frame.

After receiving an EOAF, this state machine shall check if the address frame is a valid OPEN address frame.

This state machine shall accept an address frame a valid OPEN address frame if:

a) the ADDRESS FRAME TYPE field is set to Open;
b) the number of data dwords between the SOAF and EOAF is 8; and
c) the CRC field contains a valid CRC.

Otherwise, this state machine shall discard the address frame. If the frame is not discarded then this state machine shall send a OPEN 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.

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.14.4 SL_CC (connection control) state machine

7.14.4.1 SL_CC state machine overview

The state machine consists of the following states:

a) SL_CC0:Idle (see 7.14.4.2)(initial state);
b) SL_CC1:ArbSel (see 7.14.4.3);
c) SL_CC2:Selected (see 7.14.4.4);
d) SL_CC3:Connected (see 7.14.4.5);
e) SL_CC4:DisconnectWait (see 7.14.4.6);
f) SL_CC5:BreakWait (see 7.14.4.7); and
g) SL_CC6:Break (see 7.14.4.8).

The 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 7.9.5).

The SL_CC state machine receives the following messages from the SSP link layer state machine (see 7.16.7), the STP link layer state machine, and SMP link layer state machine (see 7.18.4):

a) Request Break; and
b) Request Close.

The SL_CC 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;
b) Enable Disable STP; and
c) Enable Disable SMP.

The SL_CC state machine receives the following messages from the SL_IR state machines (see 7.9.5):

a) Enable Disable SAS Link (Enable); and
b) Enable Disable SAS Link (Disable).

Unless otherwise stated within the state description, all running disparity errors, illegal characters, and unexpected primitives (i.e., any primitive not described in the description of the SL_CC state) received within any SL state shall be ignored and idle dwords shall be transmitted.

Any detection of an internal error shall cause the SL_CC state machine to transition to the SL_CC5:BreakWait state.

The SL_CC state machine shall maintain the timers listed in table 83.

Table 83 — SL_CC timers

Timer	Initial value
Open Timeout timer	1 ms
Close Timeout timer	1 ms
Break Timeout timer	1 ms

7.14.4.2 SL_CC0:Idle state

7.14.4.2.1 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 Enable Disable SSP (Disable), Enable Disable SMP (Disable), and Enable Disable STP (Disable) messages to the SSP, SMP, and STP link layer state machines.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter (see 7.4).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If a BROADCAST Received (Change) message is received, this state shall send a Change Received confirmation to the management layer.

If a Transmit Broadcast request is received with any argument, this state shall send a Transmit BROADCAST message with the same argument to the SL transmitter.

7.14.4.2.2 Transition SL_CC0:Idle to SL_CC1:ArbSel

This transition shall occur after receiving both an Enable Disable SAS Link (Enable) confirmation and an Open Connection request. The Open Connection request includes these arguments:

a) source SAS address;
b) destination SAS address;
c) protocol;
d) arbitration wait time;
e) connection rate;
f) initiator port bit; and
g) initiator connection tag.

7.14.4.2.3 Transition SL_CC0:Idle to SL_CC2:Selected

This transition shall occur after receiving both an Enable Disable SAS Link (Enable) confirmation and an OPEN Address Frame Received message.

7.14.4.3 SL_CC1:ArbSel state

7.14.4.3.1 State description

This state is used to make a connection request.

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.

This state shall ignore incoming OPEN_REJECTs or OPEN_ACCEPTs 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 is received this state shall send a Change Received confirmation to the management layer.

If an AIP Received message is received after requesting the OPEN address frame be transmitted, 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 this state receives an OPEN_REJECT Received (No Destination) message after transmitting the OPEN address frame, this state shall send an Open Failed (No Destination) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Bad Destination) message after transmitting the OPEN address frame, this state shall send an Open Failed (Bad Destination) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Wrong Destination) message after transmitting the OPEN address frame, this state shall send an Open Failed (Wrong Destination) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Connection Rate Not Supported) message after transmitting the OPEN address frame, this state shall send an Open Failed (Connection Rate Not Supported) confirmation to the port layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If this state receives an OPEN_REJECT Received (Protocol Not Supported) message after transmitting the OPEN address frame, this state shall send an Open Failed (Protocol Not Supported) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Retry) message after transmitting the OPEN address frame, this state shall send an Open Failed (Retry) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Pathway Blocked) message after transmitting the OPEN address frame, this state shall send an Open Failed (Pathway Blocked) confirmation to the port layer.

7.14.4.3.2 Transition SL_CC1:ArbSel to SL_CC0:Idle

This transition shall occur after sending an Open Failed confirmation.

7.14.4.3.3 Transition SL_CC1:ArbSel to SL_CC2:Selected

This transition shall occur after transmitting the OPEN address frame if:

a) 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
b) no AIP Received messages have been received before an OPEN Address Frame Received message is received, and the arbitration fairness rules (see 7.12.3) indicate the received OPEN address frame overrides the outgoing OPEN address frame.

The arbitration fairness comparison shall use the value of the arbitration wait time argument to 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.

7.14.4.3.4 Transition SL_CC1:ArbSel to SL_CC3:Connected

This transition shall occur if this state receives an OPEN_ACCEPT Received message after transmitting the OPEN address frame.

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.

7.14.4.3.5 Transition SL_CC1:ArbSel to SL_CC5:BreakWait

This transition shall occur if a BREAK Received message has not been received and after:

a) a Stop Arb request is received and after sending an Open Failed (Port Layer Request) confirmation to the port layer; or
b) there is no response to the OPEN address frame before the Open Timeout timer expires and after sending an Open Failed (Open Timeout Occurred) confirmation to the port layer.

7.14.4.3.6 Transition SL_CC1:ArbSel to SL_CC6:Break

This transition shall occur after:

a) receiving a BREAK Received message; and
b) sending an Open Failed (Break Received) confirmation to the port layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.14.4.4 SL_CC2:Selected state

7.14.4.4.1 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, this state shall send a Transmit OPEN_REJECT (Wrong Destination) message to the SL transmitter;
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), this state shall send a Transmit OPEN_REJECT (Protocol Not Supported) message to the SL transmitter;
3) If the OPEN address frame CONNECTION RATE field is set to a connection rate that is not supported, this state shall send a Transmit OPEN_REJECT (Connection Rate Not Supported) message to the SL transmitter;
4) If the OPEN address frame PROTOCOL field is set to STP, the source SAS address is not that of the STP initiator port with an affiliation established or the source SAS address is not that of an STP initiator port with task file register set resources (see 7.17.3), this state shall send a Transmit OPEN_REJECT (STP Resources Busy) message to the SL transmitter;
5) If an Accept_Reject Opens (Reject SSP) request, Accept_Reject Opens (Reject SMP) request, or Accept_Reject Opens (Reject STP) request is received and the requested protocol is the corresponding protocol, this state shall send a Transmit OPEN_REJECT (Retry) message to the SL transmitter;
6) If the requested protocol is SSP and this state has not received an Accept_Reject Opens (Reject SSP) request 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;
7) If the requested protocol is SMP and this state has not received an Accept_Reject Opens (Reject SMP) request 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; or
8) If the requested protocol is STP and this state has not received an Accept_Reject Opens (Reject STP) request 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.

7.14.4.4.2 Transition SL_CC2:Selected to SL_CC0:Idle

This transition shall occur after this state sends a Transmit OPEN_REJECT message to the SL transmitter.

7.14.4.4.3 Transition SL_CC2:Selected to SL_CC3:Connected

This transition shall occur after sending a Connection Opened confirmation.

This transition shall include an Open SSP Connection, Open STP Connection, or Open SMP Connection argument based on the requested protocol.

7.14.4.4.4 Transition SL_CC2:Selected to SL_CC6:Break

This transition shall occur after a BREAK Received message is received.

7.14.4.5 SL_CC3:Connected state

7.14.4.5.1 State description

This state enables the SSP, STP, or SMP link layer state machine to transmit dwords during a connection. See 7.13 for details on rate matching during the connection.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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 7.18.4).

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 7.16.7).

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 7.17.7).

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) is received during an STP connection, this state shall clear any affiliation (see 7.17.3).

7.14.4.5.2 Transition SL_CC3:Connected to SL_CC4:DisconnectWait

This transition shall occur if a Request Close message is received.

7.14.4.5.3 Transition SL_CC3:Connected to SL_CC5:BreakWait

This transition shall occur if:

a) a Request Break message is received;
b) a BREAK Received message has not been received; and
c) after sending a Connection Closed (Break Received) confirmation to the port layer.

7.14.4.5.4 Transition SL_CC3:Connected to SL_CC6:Break

This transition shall occur if a BREAK Received message is received and after sending a Connection Closed (Break Received) confirmation to the port layer.

7.14.4.5.5 Transition SL_CC3:Connected to SL_CC7:CloseSTP

This transition shall occur if a CLOSE Received message is received during an STP connection.

7.14.4.6 SL_CC4:DisconnectWait state

7.14.4.6.1 State description

This state closes the connection and releases all resources associated with the connection.

This state shall:

1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter; 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) is received during an STP connection, this state shall clear any affiliation (see 7.17.3).

7.14.4.6.2 Transition SL_CC4:DisconnectWait to SL_CC0:Idle

This transition shall occur after:

a) sending a Transmit CLOSE message to the SL transmitter;
b) receiving a CLOSE Received message; and
c) sending a Connection Closed (Normal) confirmation to the port layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.14.4.6.3 Transition SL_CC4:DisconnectWait to SL_CC5:BreakWait

This transition shall occur if:

a) a BREAK Received message has not been received;
b) no CLOSE Received message is received in response to a Transmit CLOSE message before the Close Timeout timer expires; and
c) after sending a Connection Closed (Close Timeout) confirmation to the port layer.

7.14.4.6.4 Transition SL_CC4:DisconnectWait to SL_CC6:Break

This transition shall occur after receiving a BREAK Received message and after sending a Connection Closed (Break Received) confirmation to the port layer.

7.14.4.7 SL_CC5:BreakWait state

7.14.4.7.1 State description

This state closes the connection if one is established and releases all resources associated with the connection.

This state shall:

1) send a Transmit BREAK message to the SL transmitter; and
2) initialize and start the Break Timeout timer.

7.14.4.7.2 Transition SL_CC5:BreakWait to SL_CC0:Idle

This transition shall occur after receiving a BREAK Received message or if the Break Timeout timer expires. If a BREAK Received message is not received before the Break Timeout timer expires, this state shall send a Connection Closed (Break Timeout) confirmation to the port layer before making this transition.

7.14.4.8 SL_CC6:Break state

7.14.4.8.1 State description

This state closes any connection and releases all resources associated with this connection.

This state shall send a Transmit BREAK message to the SL transmitter.

While in this state all primitives received shall be ignored.

7.14.4.8.2 Transition SL_CC6:Break to SL_CC0:Idle

This transition shall occur after sending a Transmit BREAK message to the SL transmitter.

7.14.4.9 SL_CC7:CloseSTP state

7.14.4.9.1 State description

This state closes an STP connection and releases all resources associated with the connection.

This state shall:

1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter; and
2) send a Connection Closed (Normal) confirmation to the port layer.

While in this state all primitives received shall be ignored.

7.14.4.9.2 Transition SL_CC7:CloseSTP to SL_CC0:Idle

This transition shall occur after sending a Connection Closed (Normal) confirmation to the port layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.15 XL (link layer for expander phys) state machine

7.15.1 XL state machine overview

The XL state machine controls the flow of dwords on the physical link and establishes and maintains connections with another XL state machine as facilitated by the expander function - specifically the ECM and ECR.

This state machine consists of the following states:

a) XL0:Idle (see 7.15.3)(initial state);
b) XL1:Request_Path (see 7.15.4);
c) XL2:Request_Open (see 7.15.5);
d) XL3:Open_Confirm_Wait (see 7.15.6);
e) XL4:Open_Reject (see 7.15.7);
f) XL5:Forward_Open (see 7.15.8);
g) XL6:Open_Response_Wait (see 7.15.9);
h) XL7:Connected (see 7.15.10);
i) XL8:Close_Wait(see 7.15.11);
j) XL9:Break (see 7.15.12); and
k) XL10:Break_Wait (see 7.15.13).

The XL 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 7.9.5).

The XL state machine receives the following messages from the SL_IR state machine:

a) Enable Disable SAS Link (Enable); and
b) Enable Disable SAS Link (Disable).

Unless otherwise stated within a state description, all running disparity errors, illegal characters, and unexpected primitives received within any XL state shall be ignored.

The XL state machine shall maintain the timers listed in table 84.

Table 84 — XL timers

Timer	Initial value
Partial Pathway Timeout timer	Partial pathway timeout value (see 7.12.4.3)
Break Timeout timer	1 ms

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 88 shows several states in the XL state machine.

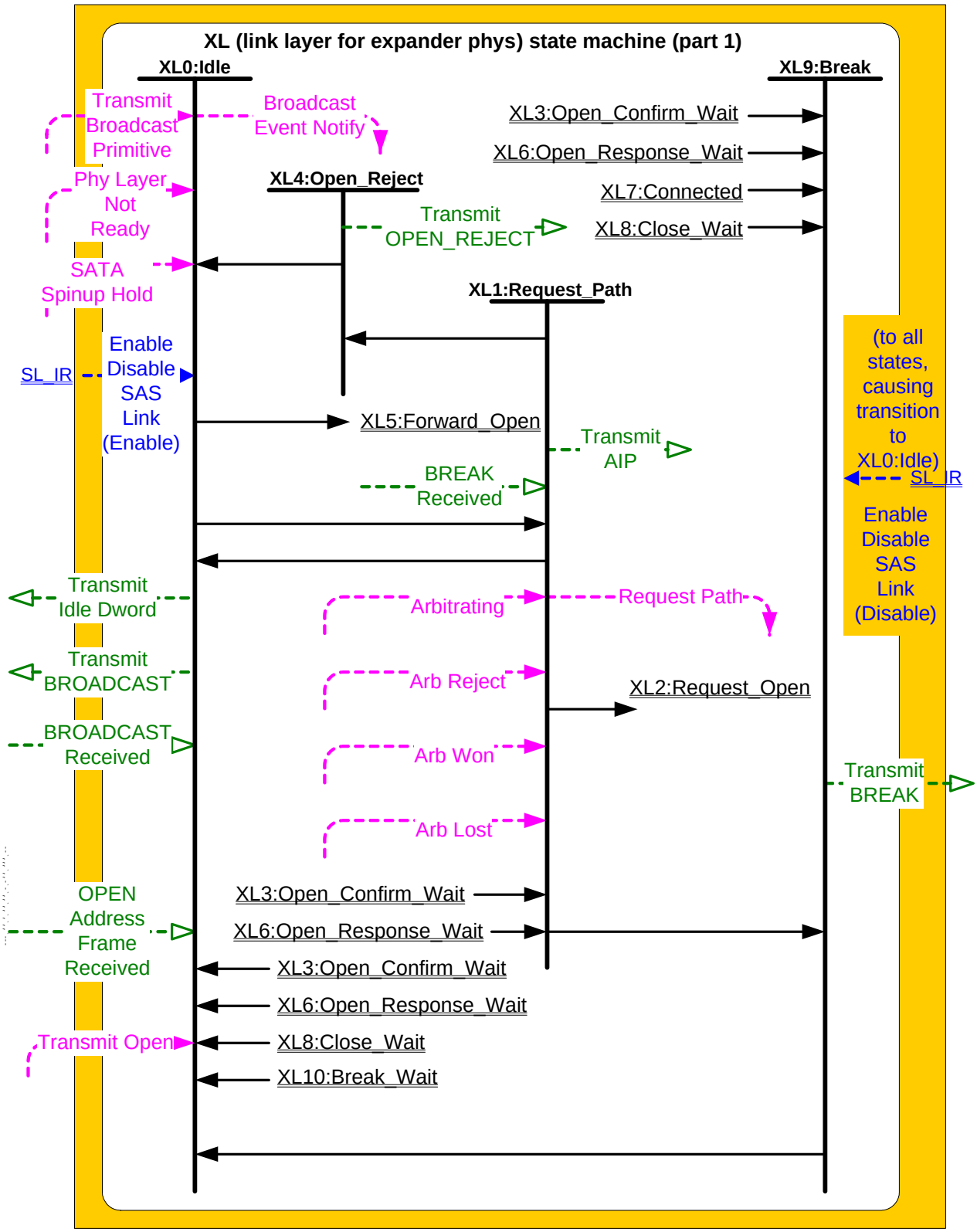


Figure 88 — XL (link layer for expander phys) state machine (part 1)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 89 shows additional states in the XL state machine.

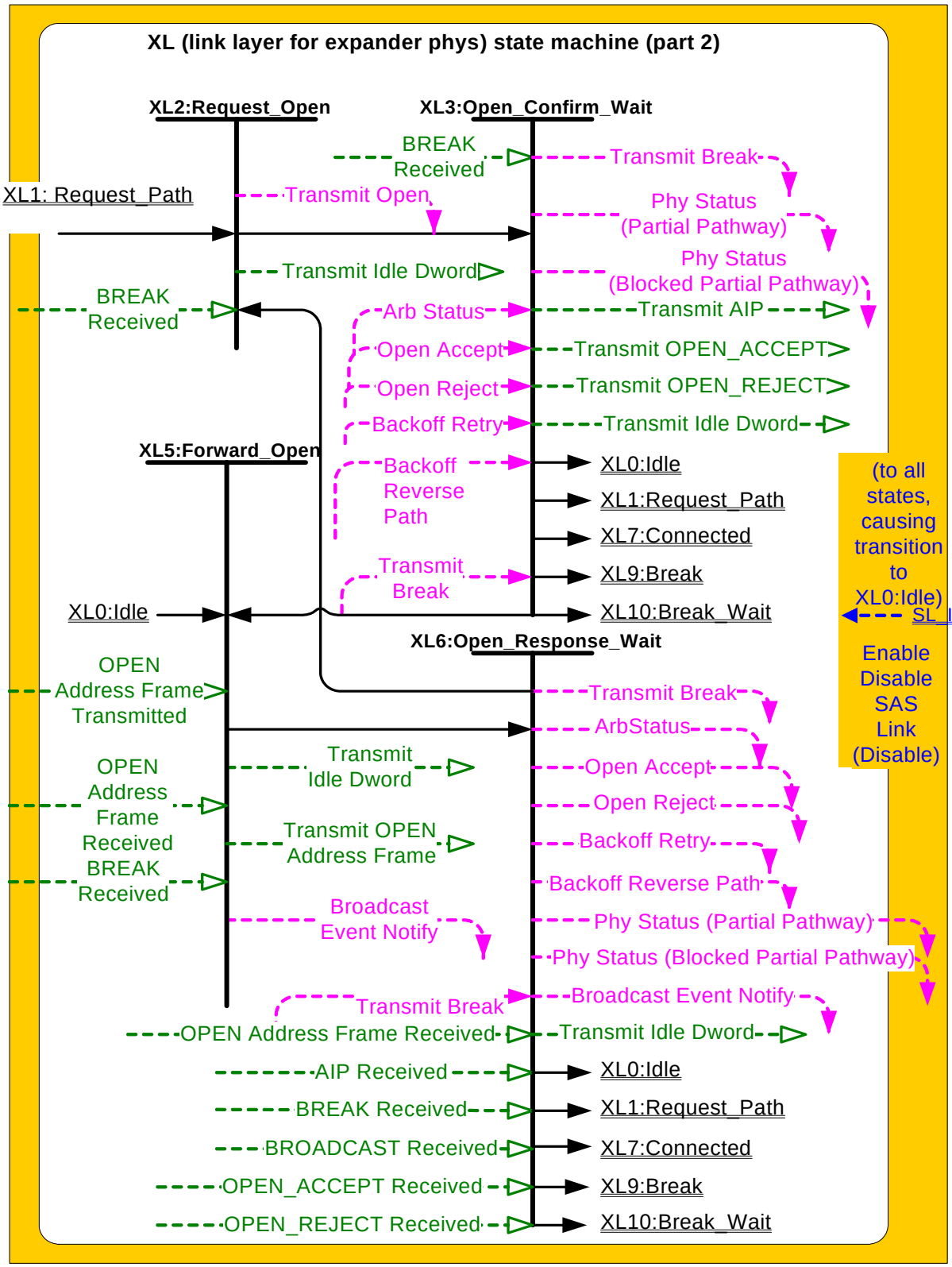


Figure 89 — XL (link layer for expander phys) state machine (part 2)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 90 shows additional states in the XL state machine.

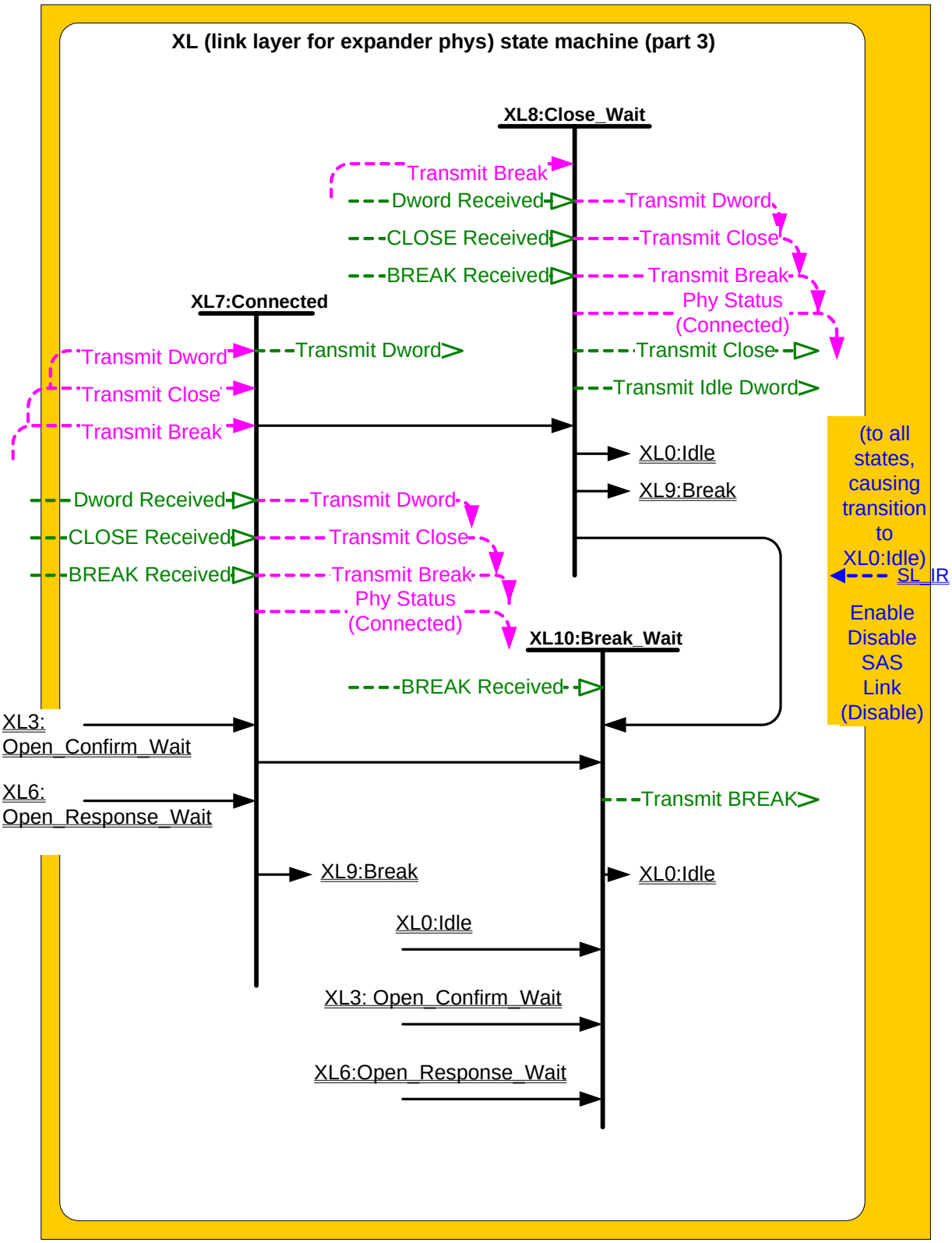


Figure 90 — XL (link layer for expander phys) state machine (part 3)

--`,,`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.15.2 XL transmitter and receiver

The XL transmitter receives the following messages from the XL state machine indicating 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 BROADCAST with an argument indicating the specific type (e.g., Transmit BROADCAST (Change));
e) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal));
f) Transmit OPEN_ACCEPT;
g) Transmit OPEN_REJECT, with an argument indicating the specific type (e.g., Transmit OPEN_REJECT (No Destination));
h) Transmit OPEN Address Frame; and
i) Transmit Dword.

The XL transmitter sends the following messages to the XL state machine:

a) OPEN Address Frame Transmitted.

The XL transmitter shall insert ALIGNs and NOTIFYs needed for rate matching (see 7.13).

The XL receiver sends the following messages to the XL state machine indicating primitive sequences, frames, and dwords received:

a) AIP Received with an argument indicating the specific type (e.g., AIP Received (Normal));
b) BREAK Received;
c) BROADCAST Received;
d) CLOSE Received;
e) OPEN_ACCEPT Received;
f) OPEN_REJECT Received;
g) OPEN Address Frame Received; and
h) Dword Received.

The XL receiver shall ignore all other dwords.

7.15.3 XL0:Idle state

7.15.3.1 State description

This state is the initial state and is the state that is used when there is no connection pending or established.

If a Phy Layer Not Ready confirmation is received, this state shall send a Broadcast Event Notify (Phy Not Ready) request to the BPP.

If a SATA Spinup Hold confirmation is received, 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, this state shall send a Broadcast Event Notify (Identification Sequence Complete) request to the BPP.

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 received (e.g., CHANGE Received).

If a Transmit Broadcast indication is received, 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.

7.15.3.2 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 Transmit Open indication is not being received; and

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

c) an OPEN Address Frame Received message is received.

7.15.3.3 Transition XL0:Idle to XL5:Forward_Open

This transition shall occur if:

a) an Enable Disable SAS Link (Enable) message has been received; and
b) a Transmit Open indication is received.

This transition shall include a set of arguments containing the arguments received in the Transmit Open indication.

7.15.4 XL1:Request_Path state

7.15.4.1 State description

This state is used to arbitrate for connection resources and to specify the destination of the connection.

Upon entry into this state, this state shall repeatedly send a Transmit AIP (Normal) parameter to the XL transmitter.

If an Arbitrating (Waiting On Partial) or Arbitrating (Blocked On Partial) confirmation is received, this state shall repeatedly send a Transmit AIP (Waiting On Partial) parameter to the XL transmitter.

If an Arbitrating (Waiting On Connection) confirmation is received, this state shall repeatedly send a Transmit AIP (Waiting On Connection) parameter to the XL transmitter.

Upon entry into this state, this state shall send a Request Path request to the ECM with the following arguments:

a) destination SAS address;
b) source SAS address;
c) protocol;
d) connection rate;
e) arbitration wait time;
f) initiator port bit;
g) initiator connection tag;
h) pathway blocked count; and
i) partial pathway timeout status.

This state maintains the Partial Pathway Timeout timer.

If the Partial Pathway Timeout timer is not already running, 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, 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, timeout status is conveyed to the expander connection manager via the partial pathway timeout status argument in the Request Path request.

7.15.4.2 Transition XL1:Request_Path to XL2:Request_Open

This transition shall occur after receiving an Arb Won confirmation.

7.15.4.3 Transition XL1:Request_Path to XL4:Open_Reject

This transition shall occur after receiving an Arb Reject confirmation. This transition shall include an Arb Reject argument corresponding to the Arb Reject confirmation.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.15.4.4 Transition XL1:Request_Path to XL0:Idle

This transition shall occur after receiving an Arb Lost confirmation.

7.15.4.5 Transition XL1:Request_Path to XL9:Break

This transition shall occur receiving a BREAK Received message.

7.15.5 XL2:Request_Open state

7.15.5.1 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 Transmit Open request to the ECR, received by the destination phy as a Transmit Open indication. The arguments to the Transmit Open request are:

a) destination SAS address;
b) source SAS address;
c) protocol;
d) connection rate;
e) arbitration wait time;
f) initiator port bit;
g) initiator connection tag;
h) features; and
i) pathway blocked count.

7.15.5.2 Transition XL2:Request_Open to XL3:Open_Confirm_Wait

This transition shall occur after sending a Transmit Open request.

If a BREAK Received message is received, this state shall include a BREAK Received argument with the transition.

7.15.6 XL3:Open_Confirm_Wait state

7.15.6.1 State description

This state waits for confirmation to an OPEN address frame sent on a destination phy.

This state shall send the following messages to the XL transmitter:

a) Transmit AIP (Normal) when an Arb Status (Normal) confirmation is received;
b) Transmit AIP (Waiting On Partial) when an Arb Status (Waiting On Partial) confirmation is received;
c) Transmit AIP (Waiting On Connection) when an Arb Status (Waiting On Connection) confirmation is received;
d) Transmit AIP (Waiting On Device) when an Arb Status (Waiting On Device) confirmation is received;
e) Transmit OPEN_ACCEPT when an Open Accept confirmation is received;
f) Transmit OPEN_REJECT when an Open Reject confirmation is received with the argument from the Open Reject confirmation, after releasing path resources; 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, this state shall release path resources.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM. 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.15.6.2 Transition XL3:Open_Confirm_Wait to XL0:Idle

This transition shall occur after sending a Transmit OPEN_REJECT message.

7.15.6.3 Transition XL3:Open_Confirm_Wait to XL1:Request_Path

This transition shall occur after receiving a Backoff Retry confirmation, after releasing path resources.

7.15.6.4 Transition XL3:Open_Confirm_Wait to XL5:Forward_Open

This transition shall occur after receiving a Backoff Reverse Path confirmation.

7.15.6.5 Transition XL3:Open_Confirm_Wait to XL7:Connected

This transition shall occur after sending a Transmit OPEN_ACCEPT message.

7.15.6.6 Transition XL3:Open_Confirm_Wait to XL9:Break

This transition shall occur after sending a Transmit Break request.

7.15.6.7 Transition XL3:Open_Confirm_Wait to XL10:Break_Wait

This transition shall occur after receiving a Transmit Break indication.

7.15.7 XL4:Open_Reject state

7.15.7.1 State description

This state is used to reject a connection request.

This state shall send one of the following messages to the XL transmitter:

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 (Bad Connection Rate) argument is received with the transition into this state; or
d) a Transmit OPEN_REJECT (Pathway Blocked) message when an Arb Reject (Pathway Blocked) argument is received with the transition into this state.

7.15.7.2 Transition XL4:Open_Reject to XL0:Idle

This transition shall occur after OPEN_REJECT has been transmitted.

7.15.8 XL5:Forward_Open state

7.15.8.1 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, this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive 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.

7.15.8.2 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 is received, this state shall include an OPEN Address Frame Received argument with the transition.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If a BREAK Received message is received, this state shall include a BREAK Received argument with the transition.

7.15.9 XL6:Open_Response_Wait state

7.15.9.1 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 request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

If a BROADCAST Received message is received before an AIP Received message is received this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive received (e.g., CHANGE Received).

This state shall send the following responses through the ECR to a source phy, received by the source phy as confirmations:

a) an Open Accept response when an OPEN_ACCEPT Received message is received;
b) an Open Reject response when an OPEN_REJECT Received message is received, after releasing any path resources;
c) a Backoff Retry response when an AIP Received message has not been received, 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 7.12.3), and 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, after releasing path resources;
d) a Backoff Retry response when an AIP Received message has been received and an OPEN Address Frame Received message is received or an OPEN Address Frame Received argument is included in the transition into this state, and 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, after releasing path resources;
e) a Backoff Reverse Path response an AIP Received message has not been received, 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 7.12.3), and 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 when an AIP Received message has been received, an OPEN Address Frame Received message is received or an OPEN Address Frame Received argument is included in the transition into this state (see 7.12.3), and 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.

This state shall send the following responses through the ECR to a source phy, received by the source phy as confirmations:

a) an Arb Status (Waiting On Device) response when an AIP Received message has not been received;
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, this state shall send a Transmit Break request to the ECR.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

This state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM. 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.

7.15.9.2 Transition XL6:Open_Response_Wait to XL0:Idle

This transition shall occur after sending an Open Reject response.

7.15.9.3 Transition XL6:Open_Response_Wait to XL1:Request_Path

This transition shall occur after sending a Backoff Retry response, after releasing path resources.

7.15.9.4 Transition XL6:Open_Response_Wait to XL2:Request_Open

This transition shall occur after sending a Backoff Reverse Path response.

7.15.9.5 Transition XL6:Open_Response_Wait to XL7:Connected

This transition shall occur after sending an Open Accept response.

7.15.9.6 Transition XL6:Open_Response_Wait to XL9:Break

This transition shall occur after sending a Transmit Break response.

7.15.9.7 Transition XL6:Open_Response_Wait to XL10:Break_Wait

This transition shall occur after receiving a Transmit Break indication.

7.15.10 XL7:Connected state

7.15.10.1 State description

This state provides a full-duplex circuit between two phys within an expander device.

This state shall send Transmit Dword messages to the XL transmitter to transmit all dwords received with Transmit Dword indications.

This state shall send Transmit Dword requests to the ECR containing each valid dword except BREAK and CLOSE primitives received with Dword Received messages.

If an invalid dword is received with the Dword Received message and the expander phy is forwarding to an expander phy attached to a SAS physical link, the expander phy shall send an ERROR primitive with the Transmit Dword request instead of the invalid dword.

If an invalid dword or an ERROR primitive is received with Dword Received message and the expander phy is forwarding to an expander phy attached to a SATA physical link, the expander phy shall send a SATA_ERROR primitive with the Transmit Dword request instead of the invalid dword or ERROR primitive.

If a CLOSE Received message is received, this state shall send a Transmit Close request to the ECR with the argument from the CLOSE Received message.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Connected) response to the ECM.

7.15.10.2 Transition XL7:Connected to XL8:Close_Wait

This transition shall occur after receiving a Transmit Close indication.

7.15.10.3 Transition XL7:Connected to XL9:Break

This transition shall occur after sending a Transmit Break request.

7.15.10.4 Transition XL7:Connected to XL10:Break_Wait

This transition shall occur after receiving a Transmit Break indication.

--`,,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.15.11 XL8:Close_Wait state

7.15.11.1 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 Transmit Close indication, then shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

If a Dword Received message is received containing a valid dword except a BREAK or CLOSE primitive, this state shall send Transmit Dword requests to the ECR containing that dword.

If an invalid dword is received with the Dword Received message and the expander phy is forwarding to an expander phy attached to a SAS physical link, the expander phy shall send an ERROR primitive with the Transmit Dword request instead of the invalid dword.

If an invalid dword or an ERROR primitive is received with Dword Received message and the expander phy is forwarding to an expander phy attached to a SATA physical link, the expander phy shall send a SATA_ERROR primitive with the Transmit Dword request instead of the invalid dword or ERROR primitive.

If a CLOSE Received message is received, this state shall release path resources and send a Transmit Close request to the ECR with the argument from the CLOSE Received message.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Connected) response to the ECM.

7.15.11.2 Transition XL8:Close_Wait to XL0:Idle

This transition shall occur after sending a Transmit Close request.

7.15.11.3 Transition XL8:Close_Wait to XL9:Break

This transition shall occur after sending a Transmit Break request.

7.15.11.4 Transition XL8:Close_Wait to XL10:Break_Wait

This transition shall occur after a Transmit Break indication is received.

7.15.12 XL9:Break state

7.15.12.1 State description

This state closes any connection and releases path resources.

This state shall send a Transmit BREAK message to the XL transmitter.

7.15.12.2 Transition XL9:Break to XL0:Idle

This transition shall occur after sending a Transmit BREAK message to the XL transmitter.

7.15.13 XL10:Break_Wait state

7.15.13.1 State description

This state closes any connection and releases path resources.

This state shall send a Transmit BREAK message to the XL transmitter. After transmitting the BREAK this state shall initialize and start the Break Timeout timer.

7.15.13.2 Transition XL10:Break_Wait to XL0:Idle

This transition shall occur after a BREAK Received message is received or after the Break Timeout timer expires.

--`,,`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.16 SSP link layer

7.16.1 Opening an SSP connection

An SSP phy that accepts 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, it shall respond with OPEN_REJECT (RETRY) and not accept the connection request.

7.16.2 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, it transmits a DONE to start closing the connection (see 8.2.2.3.5). The other direction may still be active, so the DONE may be followed by one or more CREDIT_BLOCKED, RRDY, ACK, or NAKs.

7.16.3 SSP frame transmission and reception

During an SSP connection, SSP frames are preceded by SOF and followed by EOF as shown in figure 91.

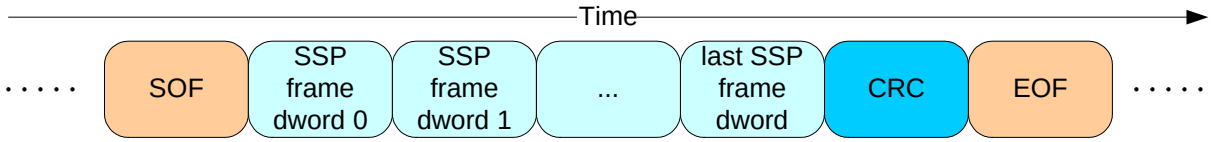


Figure 91 — SSP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5). The SSP link layer state machine checks that the frame is not too short and that the CRC is valid (see 7.16.7.7).

Receiving SSP phys shall acknowledge SSP frames within 1 ms if not discarded as described in 7.16.7.7 with either a positive acknowledgement (ACK) or a negative acknowledgement (NAK). ACK means the SSP frame was received into a frame buffer without errors. NAK (CRC ERROR) means the SSP frame was received with a CRC error.

Some interlocked frames which are NAKed may be retried by the transport layer (see 9.2.4). Non-interlocked frames which are NAKed shall not be retried by any layer. The SCSI command associated with a NAKed frame that is not successfully retried shall be terminated with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.2.3).

7.16.4 SSP flow control

SSP phys grant credit for permission to transmit frames with RRDY. 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 never refuse to provide credit by withholding RRDY because it needs to transmit a frame itself. It 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.16.5 Interlocked frames

Table 85 shows which SSP frames shall be interlocked and which are non-interlocked.

Table 85 — SSP frame interlock requirements

SSP frame type	Interlock requirement
COMMAND	Interlocked
TASK	Interlocked
XFER_RDY	Interlocked
DATA	Non-interlocked
RESPONSE	Interlocked
See 9.2 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 it 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 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 tag if credit is available. The phy shall not transmit:

a) a non-interlocked frame with a different 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., an SSP initiator phy could 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 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 indicate credit is not forthcoming) while waiting for an interlocked frame it transmitted to be acknowledged. These primitives may also be interspersed within an SSP frame.

Figure 92 shows an example of interlocked frame transmission.

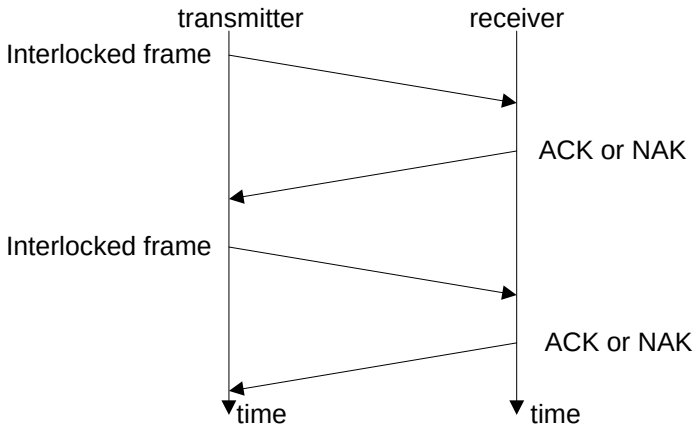


Figure 92 — Interlocked frames

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 93 shows an example of non-interlocked frame transmission with the same tags.

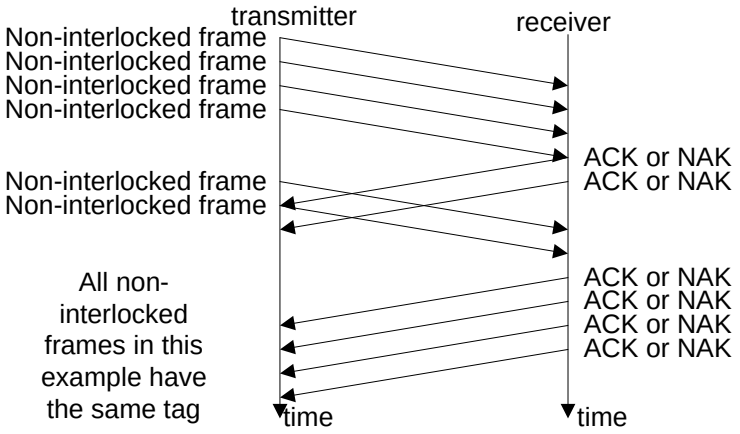


Figure 93 — Non-interlocked frames with the same tag

Figure 94 shows an example of non-interlocked frame transmission with different tags.

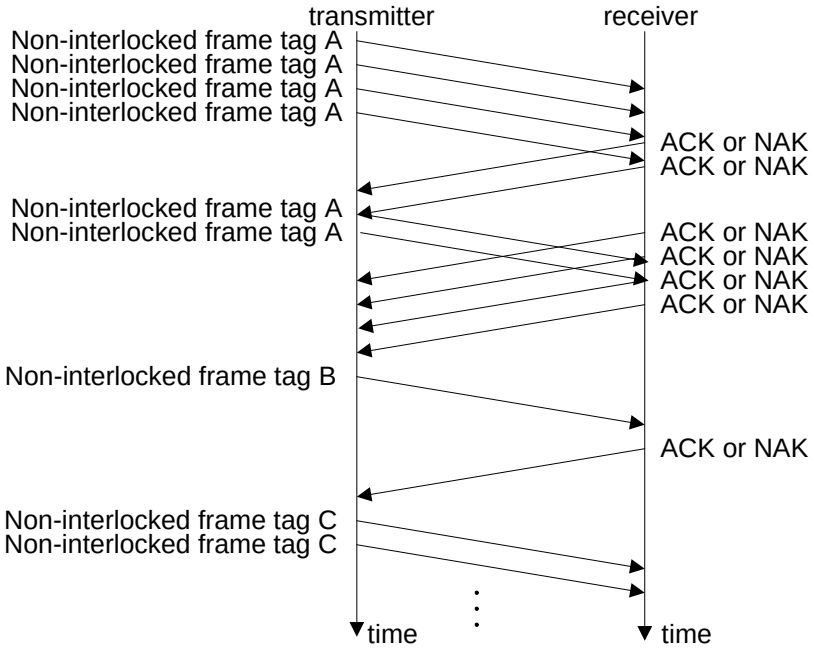


Figure 94 — Non-interlocked frames with different tags

7.16.6 Closing an SSP connection

DONE shall be exchanged prior to closing an SSP connection (see 8.2.2.3.5). There are several versions of the DONE primitive indicating additional information about why the SSP connection is being closed:

a) DONE (NORMAL) indicates normal completion; the transmitter has no more SSP frames to transmit;
b) DONE (CREDIT TIMEOUT) indicates the transmitter still has SSP frames to transmit, but did not receive an RRDY granting frame credit within 1 ms; and
c) DONE (ACK/NAK TIMEOUT) indicates 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 in 1 ms unless the recipient replies with DONE and the connection is closed.

After transmitting DONE, the transmitting phy initializes and starts a 1 ms DONE Timeout timer (see 7.16.7.5).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

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 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 CLOSE (NORMAL) (see 7.12.7).

Figure 95 shows the sequence for a closing an SSP connection.

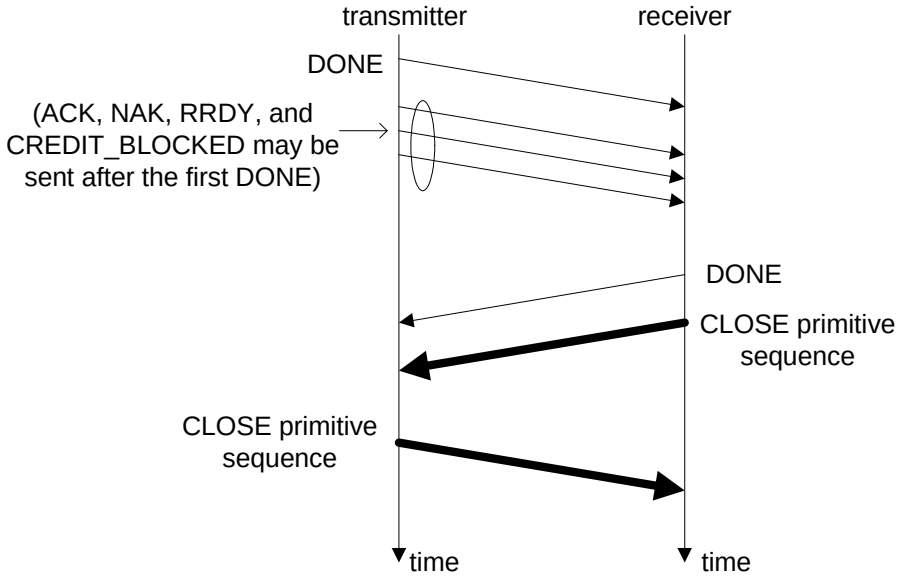


Figure 95 — Closing an SSP connection example

7.16.7 SSP (link layer for SSP phys) state machines

7.16.7.1 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:

a) SSP_TIM (transmit interlocked frame monitor) state machine (see 7.16.7.3);
b) SSP_TCM (transmit frame credit monitor) state machine (see 7.16.7.4);
c) SSP_D (DONE control) state machine (see 7.16.7.5);
d) SSP_TF (transmit frame control) state machine (see 7.16.7.6);
e) SSP_RF (receive frame control) state machine (see 7.16.7.7);
f) SSP_RCM (receive frame credit monitor) state machine (see 7.16.7.8);
g) SSP_RIM (receive interlocked frame monitor) state machine (see 7.16.7.9);
h) SSP_TC (transmit credit control) state machine (see 7.16.7.10); and
i) SSP_TAN (transmit ACK/NAK control) state machine (see 7.16.7.11).

All the SSP state machines shall start after receiving an Enable Disable SSP (Enable) message from the SL state machines (see 7.14).

All the SSP state machines shall terminate after:

a) receiving an Enable Disable SSP (Disable) message from the SL state machines;
b) receiving a Request Close message from the SSP_D state machine indicating that the connection has been closed; or
c) receiving a Request Break message from the SSP_D state machine indicating that a BREAK has been transmitted.

If a state machine consists of multiple states the initial state is as indicated in the state machine description in this subclause.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The SSP state machines shall maintain the timers listed in table 86.

Table 86 — SSP link layer timers

Timer	Initial value
ACK/NAK Timeout timer	1 ms
DONE Timeout timer	1 ms
Credit Timeout timer	1 ms

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 96 shows the SSP state machines and states related to frame transmission.

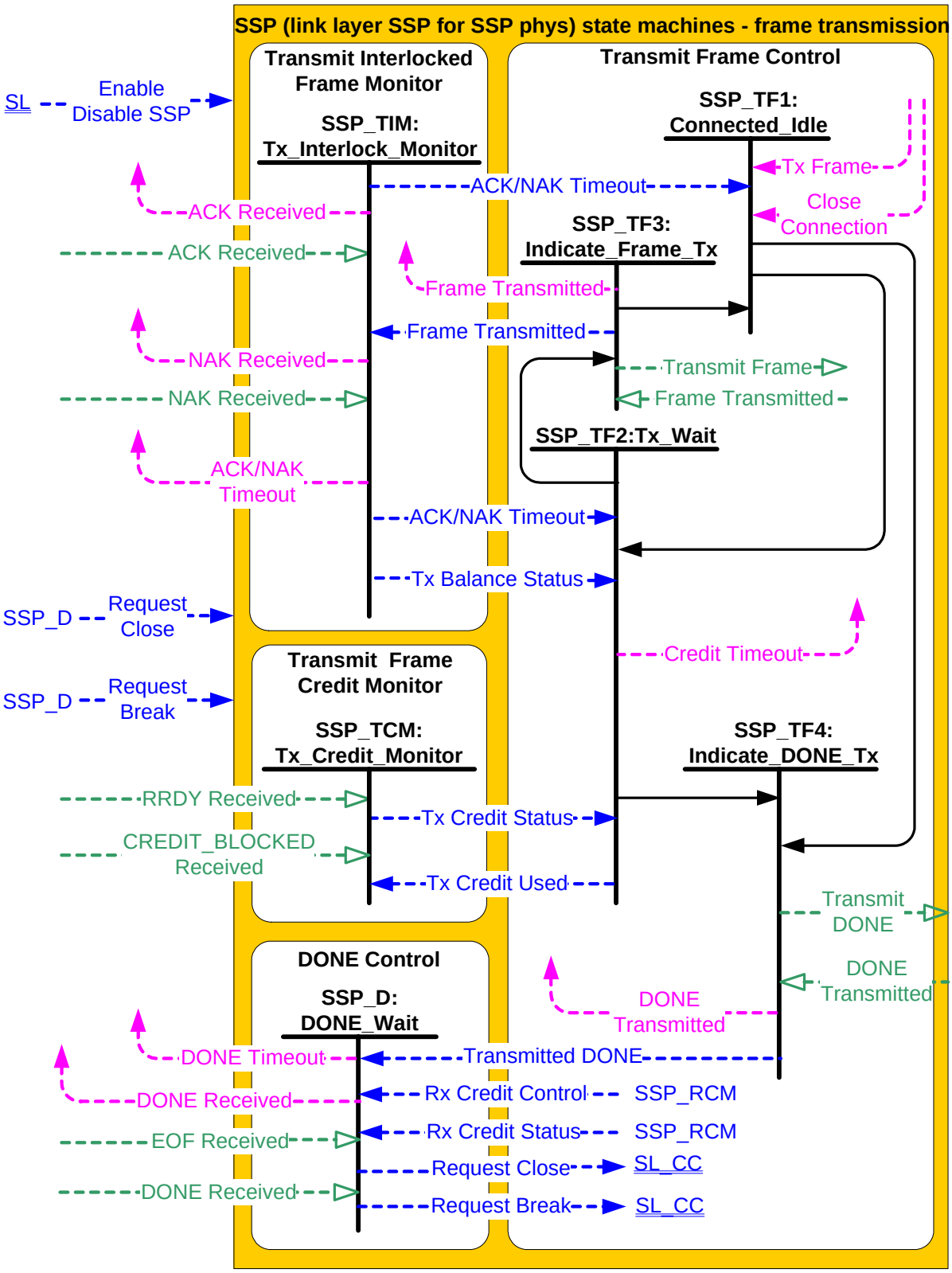


Figure 96 — SSP (link layer for SSP phys) state machines (part 1 - frame transmission)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 97 shows the SSP state machines and states related to frame reception.

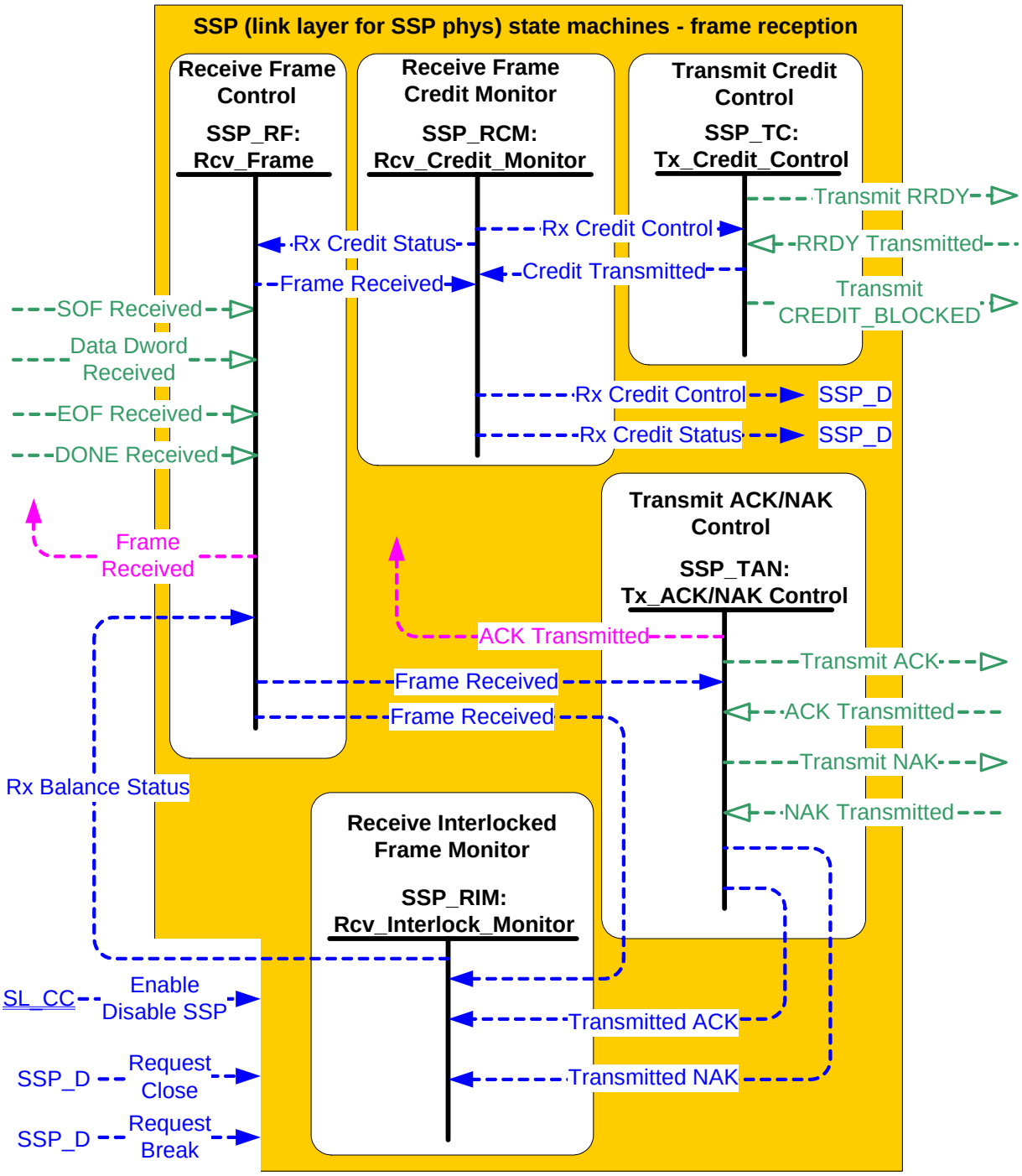


Figure 97 — SSP (link layer for SSP phys) state machines (part 2 - frame reception)

7.16.7.2 SSP transmitter and receiver

The SSP transmitter receives the following messages from the SSP state machines indicating primitive sequences and frames to transmit:

a) Transmit RRDY with an argument indicating the specific type (e.g., Transmit RRDY (Normal));
b) Transmit CREDIT_BLOCKED;
c) Transmit ACK;
d) Transmit NAK with an argument indicating the specific type (e.g., Transmit NAK (CRC Error));

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

e) Transmit Frame (i.e., SOF/data dwords/EOF); and
f) Transmit DONE with an argument indicating the specific type (e.g., Transmit DONE (Normal)).

The SSP transmitter sends the following messages to the SSP state machines:

a) DONE Transmitted;
b) RRDY Transmitted;
c) CREDIT_BLOCKED Transmitted;
d) ACK Transmitted;
e) NAK Transmitted; and
f) Frame Transmitted.

When the SSP transmitter is not processing a message to transmit, it shall transmit idle dwords.

The SSP receiver sends the following messages to the SSP state machines indicating primitive sequences and dwords received:

a) ACK Received;
b) NAK Received;
c) RRDY Received;
d) CREDIT_BLOCKED Received;
e) EOF Received;
f) DONE Received with an argument indicating the specific type (e.g., DONE Received (Normal));
g) SOF Received;
h) Data Dword Received; and
i) EOF Received.

The SSP receiver shall ignore all other dwords.

7.16.7.3 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 and monitors the number of ACKs and NAKs received. This state machine ensures that an ACK or NAK is received for each frame transmitted and indicates a timeout if they are not.

The Frame Transmitted message shall be used by this state machine to count the number of frames transmitted.

When the number of Frame Transmitted messages received equals the number of ACK Received and NAK Received messages received then 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 Frame Transmitted messages received does not equal the number of ACK Received and NAK Received messages received then this 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.

If the ACK/NAK count is not balanced and an ACK Received message is received this state machine shall:

a) use the ACK Received message to count the number of ACKs and NAKs received; and
b) send an ACK Received confirmation to the port layer each time the ACK Received message is received.

If the ACK/NAK count is not balanced and an NAK Received message is received this state machine shall:

a) use the NAK Received message to count the number of ACKs and NAKs received; and
b) send an NAK Received confirmation to the port layer each time the NAK Received message is received.

If the ACK/NAK count is balanced, 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 re-initialized each time an ACK or NAK is counted. If the ACK/NAK

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Timeout timer expires, 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 shall be set to the number of ACKs and NAKs received.

7.16.7.4 SSP_TCM (transmit frame credit monitor) state machine

The SSP_TCM state machine's function is to ensure that credit is available from the originator 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 received versus the number of transmit frame credits used. This state machine adds transmit frame credit for each RRDY Received message received and subtracts transmit frame credit for each Tx Credit Used message received. This state machine shall remember any CREDIT_BLOCKED Received message that is received.

When transmit frame credit is available, this state machine shall send the Tx Credit Status (Credit Available) message to the SSP_TF2:Tx_Wait state.

When transmit frame credit is not available and credit is not blocked, this state machine shall send the Tx Credit Status (Credit Not Available) message to the SSP_TF2:Tx_Wait state.

When transmit frame credit is not available and credit is blocked, this state machine shall send the Tx Credit Status (Credit Blocked) message to the SSP_TF2:Tx_Wait state.

When this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message transmit frame credit shall be set to not available and credit shall not be blocked.

7.16.7.5 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 the DONE Received message has been received before the Transmitted DONE message is received, this state machine shall send the Request Close message to the SL_CC state machine (see 7.14) and all the SSP state machines after receiving the Transmitted DONE message.

If a DONE Received message, the Transmitted DONE (Normal) message, or the Transmitted DONE (Credit Timeout) message has not been received and the Rx Credit Status (Extended) message or the Rx Credit Control (Blocked) message has been received, then this state shall initialize and start the DONE Timeout timer after receiving the Transmitted DONE (Normal) message or the Transmitted DONE (Credit Timeout) message.

If the DONE Received message has not been received and the Transmitted DONE (Normal) message or the Transmitted DONE (Credit Timeout) message has been received, this state machine shall initialize and start the DONE Timeout timer each time:

a) the Rx Credit Status (Extended) message is received; or
b) the Rx Credit Control (Blocked) message is received.

If the Transmitted DONE (Normal) message or the Transmitted DONE (Credit Timeout) message has been received, the DONE Timeout timer shall be reinitialized each time the EOF Received message is received.

If the Transmitted DONE (Normal) message or the Transmitted DONE (Credit Timeout) message has been received, the DONE Timeout timer shall be stopped after:

a) the Rx Credit Status (Exhausted) message is received; and
b) the Rx Credit Control (Blocked) message has not been received.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

NOTE 19 - 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 the DONE Received message has not been received and the Transmitted DONE (ACK/NAK Timeout) message has been received:

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 the DONE Received message is received before the DONE Timeout timer expires, this state machine shall send the Request Close message to the SL_CC state machine and all the SSP state machines.

If the DONE Received message is not received before the DONE Timeout timer expires, this state machine shall:

a) send a DONE Timeout confirmation to the port layer; and
b) 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 (Close Connection) and DONE Received (Credit Timeout)) may be used by the application layer to decide when to reuse tags.

7.16.7.6 SSP_TF (transmit frame control) state machine

7.16.7.6.1 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 7.16.7.6.2)(initial state);
b) SSP_TF2:Tx_Wait (see 7.16.7.6.3);
c) SSP_TF3:Indicate_Frame_Tx (see 7.16.7.6.4); and
d) SSP_TF4:Indicate_DONE_Tx (see 7.16.7.6.5).

7.16.7.6.2 SSP_TF1:Connected_Idle state

7.16.7.6.2.1 State description

This state waits for a request to transmit a frame or to close the connection.

7.16.7.6.2.2 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 this transition shall include a Transmit Frame Balance Required argument.

If a Tx Frame (Balance Not Required) request was received this transition shall include a Transmit Frame Balance Not Required argument.

If a Close Connection request was received this transition shall include a Close Connection argument.

7.16.7.6.2.3 Transition SSP_TF1:Connected_Idle to SSP_TF4:Indicate_DONE_Tx

This transition shall occur if an ACK/NAK Timeout message is received. This transition shall include an ACK/NAK Timeout argument.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.16.7.6.3 SSP_TF2:Tx_Wait state

7.16.7.6.3.1 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 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 this state shall stop the Credit Timeout timer.

7.16.7.6.3.2 Transition SSP_TF2:Tx_Wait to SSP_TF3:Indicate_Frame_Tx

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 Credit 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 Credit Available.

This transition shall occur after sending a Tx Credit Used message to the SSP_TCM state machine.

7.16.7.6.3.3 Transition SSP_TF2:Tx_Wait to SSP_TF4:Indicate_DONE_Tx

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 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.

7.16.7.6.4 SSP_TF3:Indicate_Frame_Tx state

7.16.7.6.4.1 State description

This state shall request a frame transmission by sending a Transmit Frame message to the SSP transmitter. Each time a Transmit Frame message is sent to the SSP transmitter, one SSP frame (i.e., SOF, frame contents, and EOF) is transmitted.

In this state receiving a Frame Transmitted message indicates that the frame has been transmitted.

7.16.7.6.4.2 Transition SSP_TF3:Indicate_Frame_Tx to SSP_TF1:Connected_Idle

This transition shall occur after:

a) receiving a Frame Transmitted message;

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) sending an Frame Transmitted message to the SSP_TIM state machine; and
c) sending a Frame Transmitted confirmation to the port layer.

7.16.7.6.5 SSP_TF4:Indicate_DONE_Tx 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.

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.

7.16.7.7 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 successfully. 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.

The frame (i.e., all the dwords between an SOF and EOF) shall be discarded if any of the following conditions are true:

a) the number of data dwords between the SOF and EOF is less than 7;
b) the number of data dwords after the SOF is greater than 263 data dwords;
c) the Rx Credit Status (Credit Exhausted) message is received; or
d) the DONE Received message is received.

If consecutive SOF Received messages are received without an intervening 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 machine shall discard all dwords between those SOFs.

If the frame is discarded then no further action is taken by this state machine. If the frame is not discarded then this state machine shall:

a) send a Frame Received message to the SSP_RCM state machine; and
b) send a Frame Received message to the SSP_RIM state machine;

If the frame CRC is good and the frame contained no invalid data dwords, this state machine shall send a Frame Received (Successful) message to the SSP_TAN1:Idle state and:

a) if the last Rx Balance Status message received had an argument of Balanced, 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, send a Frame Received (ACK/NAK Not Balanced) confirmation to the port layer.

If the frame CRC is bad or the frame contained invalid data dwords, this state machine shall send a Frame Received (Unsuccessful) message to the SSP_TAN1:Idle state.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.16.7.8 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 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 indicating that no more credit is going to be sent during this connection. 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. The Rx Credit Control (Blocked) message should be sent to the SSP_TC state machine and the SSP_D state machine when no further credit is going to become available within a credit timeout (i.e., less than 1 ms).

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 5 frames the maximum number of Rx Credit Control requests with the Available argument outstanding is 5).

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 (Credit 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 (Credit 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.

7.16.7.9 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 versus the number of ACKs and NAKs transmitted.

This state machine shall use the ACK Transmitted message and the NAK Transmitted message to keep track of the number of ACKs and NAKs 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 the ACK Transmitted messages and the number of NAK Transmitted messages received equals the number of Frame Received messages received this state machine shall send an Rx Balance Status (Balanced) message to the SSP_RF state machine.

Any time the number of the ACK Transmitted messages and the number of NAK Transmitted messages received does not equal the number of Frame Received messages received this state machine shall send an Rx Balance Status (Not Balanced) message to the SSP_RF state machine.

When the SL state machines send the Enable Disable SSP (Enable) message the number of the ACKs and NAKs transmitted shall be set to the number of frames received.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.16.7.10 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 5 RRDYs are to be transmitted this state machine sends 5 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.

7.16.7.11 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 receives a NAK Transmitted argument it shall send a Transmitted NAK message to the SSP_RIM state machine.

7.17 STP link layer

7.17.1 STP frame transmission and reception

STP frame transmission is defined by SATA (see ATA/ATAPI-7 V3). During an STP connection, frames are preceded by SATA_SOF and followed by SATA_EOF as shown in figure 98.

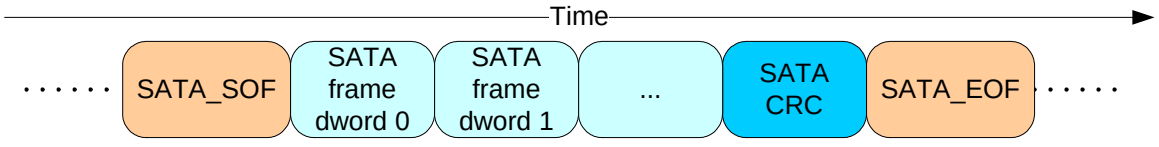


Figure 98 — STP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5).

Other primitives may be interspersed during the connection as defined by SATA.

STP encapsulates SATA with connection management.

7.17.2 STP flow control

Each STP port (i.e., STP initiator port and STP target port) and expander port through which the STP connection is routed shall implement the SATA flow control protocol on each physical link. The flow control primitives are not forwarded through expander devices like other dwords.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

When an STP port is receiving a frame and its buffer begins to fill up, it shall transmit SATA_HOLD. After transmitting SATA_HOLD, it shall accept the following number of data dwords for the frame:

a) 24 dwords at 1,5 Gbps; or
b) 28 dwords at 3,0 Gbps.

When an STP port is transmitting a frame and receives SATA_HOLD, it shall transmit no more than 20 data dwords for the frame and respond with SATA_HOLDA.

NOTE 20 - The receive buffer requirements are based on (20 + (4 × n)) where n is 1 for 1,5 Gbps and 2 for 3,0 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see ATA/ATAPI-7 V3). The (4 × n) portion of this equation is based on:
a) One-way propagation time on a 10 m cable = (5 ns/m propagation delay) × (10 m cable) = 50 ns;
b) Round-trip propagation time on a 10 m cable = 100 ns (e.g., time to send SATA_HOLD and receive SATA_HOLDA);
c) Time to transmit a 1,5 Gbps dword = (0,667 ns/bit unit interval) × (40 bits/dword) = 26,667 ns; and
d) Number of 1,5 Gbps dwords on the wire during round-trip propagation time = (100 ns / 26,667 ns) = 3,75.
Receivers may support longer cables by providing larger buffer sizes.

When a SATA host port in an STP/SATA bridge is receiving a frame from a SATA physical link, it shall transmit a SATA_HOLD when it is only capable of receiving 21 more dwords.

NOTE 21 - SATA requires that frame transmission cease and SATA_HOLDA be transmitted within 20 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 port in an STP/SATA bridge is transmitting a frame to a SATA physical link, it shall transmit no more than 19 data dwords after receiving SATA_HOLD.

NOTE 22 - SATA assumes that once a SATA_HOLD is transmitted, frame transmission ceases and SATA_HOLDA arrives within 20 dwords. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

Figure 99 shows STP flow control between:

a) an STP initiator port 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 port transmitting a frame.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

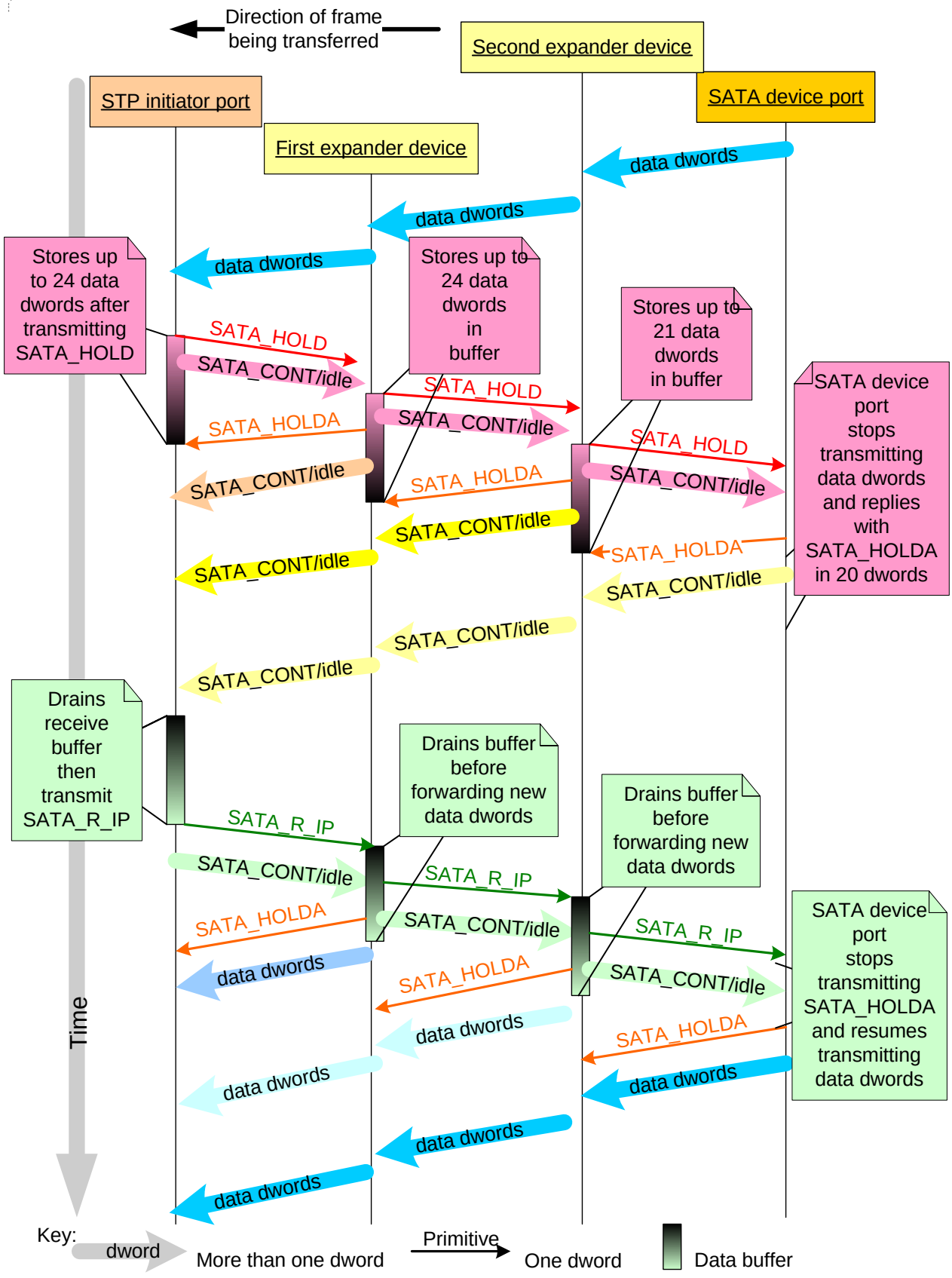


Figure 99 — STP flow control

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

After the STP initiator port transmits SATA_HOLD, it receives a SATA_HOLDA reply from the first expander device within 24 dwords (for a 1,5 Gbps physical link). The first expander device transmits SATA_HOLD to the second expander device and receives SATA_HOLDA within 24 dwords (for a 1,5 Gbps physical link), buffering data dwords it is no longer able to forward to the STP initiator port. The second expander device transmits SATA_HOLD to the SATA device port and receives SATA_HOLDA within 21 dwords (for a SATA physical link), buffering data dwords it is no longer able to forward to the first expander device. When the SATA device port stops transmitting data dwords, its previous data dwords are stored in the buffers in both expander devices and the STP initiator port.

After the STP initiator port drains its buffer and transmits SATA_R_IP, it receives data dwords from the first expander device's buffer, followed by data dwords from the second expander device's buffer, followed by data dwords from the SATA device port.

7.17.3 Affiliations

Coherent access to the SATA task file registers shall be provided for each STP initiator port. STP target ports that do not track all commands by the STP initiator ports' SAS addresses shall implement affiliations to provide coherency. STP target ports that track all commands by the STP initiator ports' SAS addresses shall not implement affiliations.

An affiliation is a state entered by an STP target port where it refuses to accept connection requests from STP initiator ports other than the one that has established an affiliation.

An STP target port that supports affiliations shall establish an affiliation whenever it accepts a connection request. When an affiliation is established, 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:

a) Power on;
b) the SAS target device receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of HARD RESET (see 10.4.3.10) from any SMP initiator port;
c) the SAS target device receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of CLEAR AFFILIATION from the same SAS initiator port that has the affiliation;
d) A connection to the phy with the affiliation is closed with CLOSE (CLEAR AFFILIATION); or
e) the STP target port is part of a STP/SATA bridge and a link reset sequence is begun on the SATA physical link.

An affiliation established when the command is transmitted shall be maintained until all frames for the command have been delivered. An STP initiator port implementing command queuing shall maintain an affiliation while any commands are outstanding. This avoids confusing the SATA device, which only knows about one SATA host. STP initiator ports may keep affiliations for longer tenures, but this is discouraged.

An STP target port that implements affiliations shall implement one affiliation per STP target port. Multiple phys on the same STP target port shall use the same affiliation. Support for affiliations is indicated in the SMP REPORT PHY SATA function response (see 10.4.3.7).

Only one connection between a wide STP initiator port and a wide STP target port shall be allowed at one time. The STP target port shall reject a second connection request from the same STP initiator port with OPEN_REJECT (STP RESOURCES BUSY).

7.17.4 Opening an STP connection

If no STP connection exists when the SATA host port in an STP/SATA bridge receives a SATA_X_RDY from the attached SATA device port, the STP target port in the STP/SATA bridge shall establish an STP connection to the appropriate STP initiator port before it transmits a SATA_R_RDY to the SATA device.

7.17.5 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

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SATA_R_RDY until after both sending and receiving SATA_SYNC. An STP port shall transmit CLOSE after receiving a CLOSE if it has not already transmitted CLOSE.

When an STP initiator port closes an STP connection, it shall transmit a CLOSE (NORMAL) or CLOSE (CLEAR AFFILIATION). When an STP target port closes an STP connection, it shall transmit a CLOSE (NORMAL).

An STP initiator port may issue CLOSE (CLEAR AFFILIATION) in place of a CLOSE (NORMAL) to cause the STP target port to clear the affiliation (see 7.17.3) along with closing the connection. If an STP target port receives CLOSE (CLEAR AFFILIATION), the STP target port shall clear the affiliation for the STP initiator port that sent the CLOSE (CLEAR AFFILIATION).

See 7.12.7 for additional details on closing connections.

An STP/SATA bridge shall break an STP connection if its SATA host phy loses dword synchronization (see 7.12.8).

7.17.6 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 7.17.2), while the connection is open.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 100 shows an STP initiator port opening a connection, transmitting a single SATA frame, and closing the connection.

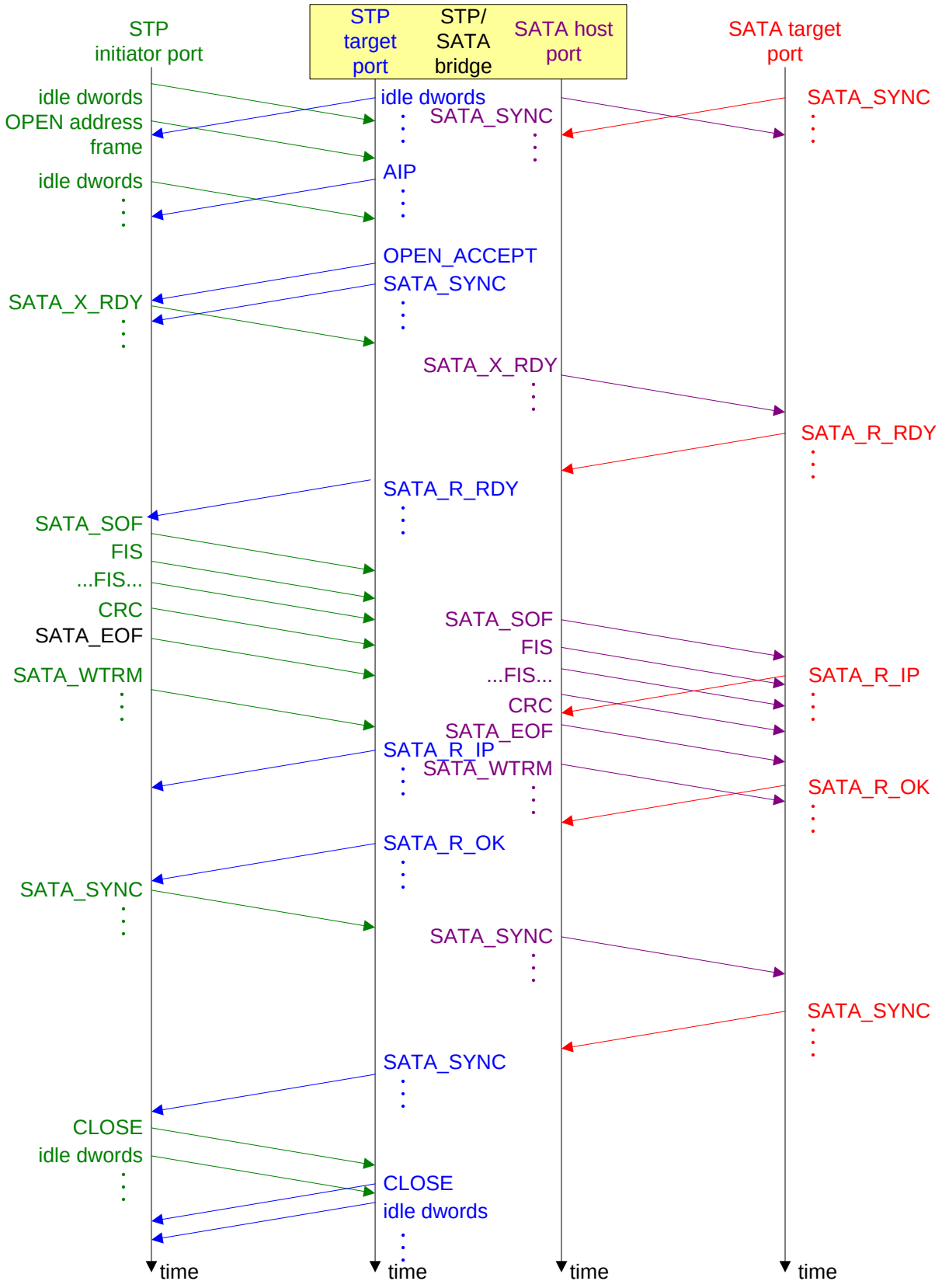


Figure 100 — STP initiator port opening an STP connection

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Figure 101 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.

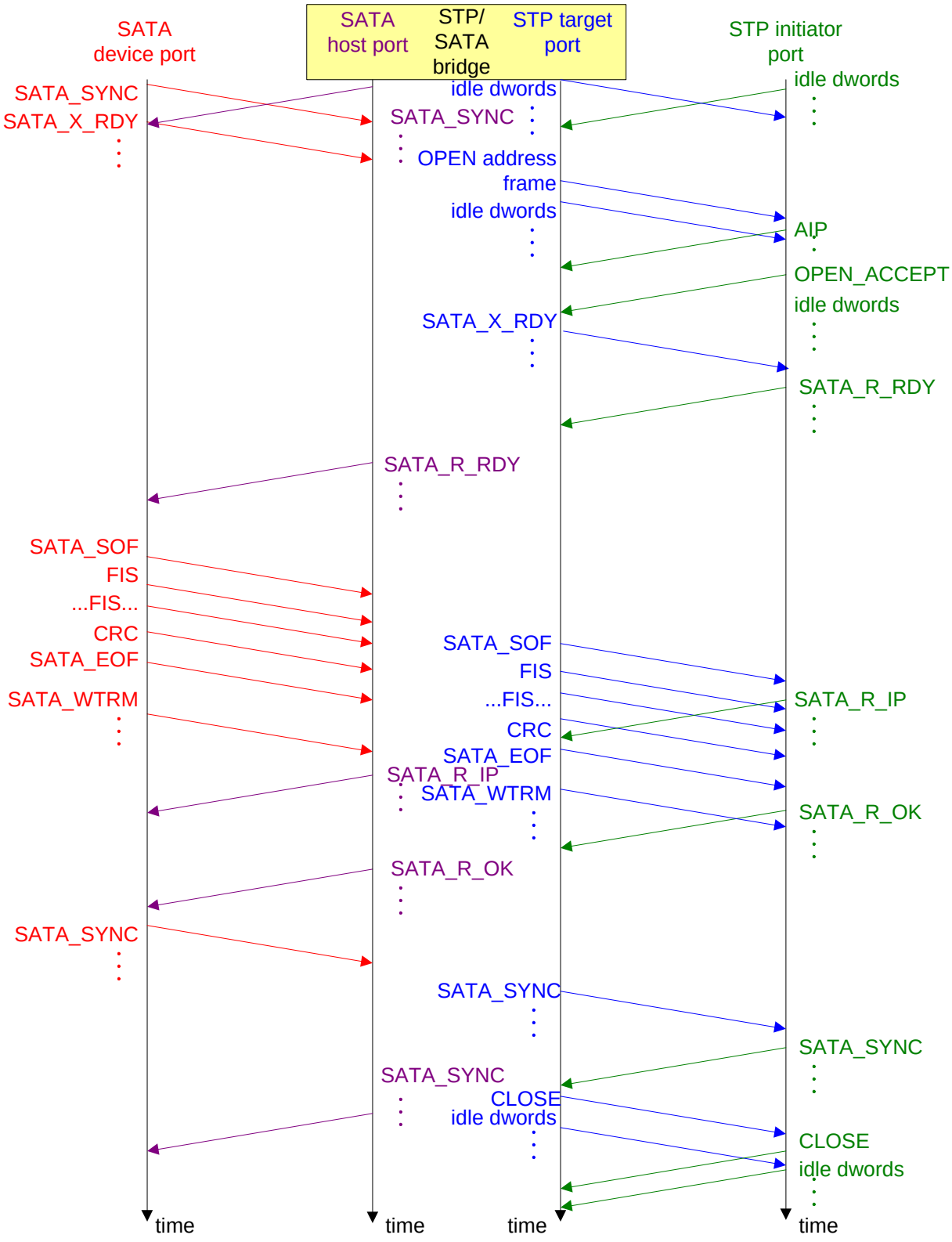


Figure 101 — STP target port opening an STP connection

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.17.7 STP (link layer for STP phys) state machines

The STP link layer uses the SATA link layer state machines (see ATA/ATAPI-7 V3), 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 or an OPEN_ACCEPT and the first SATA primitive); and
c) implement affiliations (see 7.17.3).

These modifications are not described in this standard.

7.17.8 SMP target port support

A SAS device that contains an STP target port shall also contain an SMP target port.

7.18 SMP link layer

7.18.1 SMP frame transmission and reception

Inside an SMP connection, the source device transmits a single SMP_REQUEST frame and the destination device responds with a single SMP_RESPONSE frame (see 9.4).

Frames are surrounded by SOF and EOF as shown in figure 102. There is no acknowledgement of SMP frames with ACK and NAK. There is no credit exchange with RRDY.

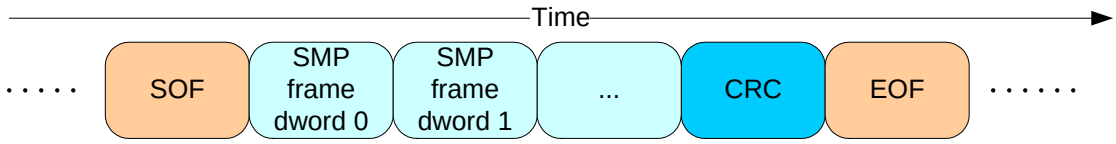


Figure 102 — SMP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5). The SMP link layer state machine checks that the frame is not too short and that the CRC is valid (see 7.18.4).

7.18.2 SMP flow control

By accepting an SMP connection, the destination device indicates it is ready to receive one SMP_REQUEST frame.

After the source device transmits one SMP_REQUEST frame, it shall be ready to receive one SMP_RESPONSE frame.

7.18.3 Closing an SMP connection

After receiving the SMP_RESPONSE frame, the source device shall transmit a CLOSE (NORMAL) to close the connection.

After transmitting the SMP_RESPONSE frame, the destination device shall reply with a CLOSE (NORMAL).

See 7.12.7 for additional details on closing connections.

7.18.4 SMP (link layer for SMP phys) state machines

7.18.4.1 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 7.18.4.3); and
b) SMP_TP (link layer for SMP target phys) state machine (see 7.18.4.4).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.18.4.2 SMP transmitter and receiver

The SMP transmitter receives the following messages from the SMP state machines indicating dwords and frames to transmit:

a) Transmit Idle Dword; and
b) Transmit Frame.

The SMP transmitter sends the following messages to the SMP state machines:

a) Frame Transmitted.

The SMP receiver sends the following messages to the SMP state machines indicating primitive sequences and dwords received:

a) SOF Received;
b) Dword Received; and
c) EOF Received;

The SMP receiver shall ignore all other dwords.

7.18.4.3 SMP_IP (link layer for SMP initiator phys) state machine

7.18.4.3.1 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 7.18.4.3.2)(initial state);
b) SMP_IP2:Transmit_Frame (see 7.18.4.3.3); and
c) SMP_IP3:Receive_Frame (see 7.18.4.3.4).

The SMP_IP 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 7.14).

The SMP_IP state machine shall terminate after receiving an Enable Disable SMP (Disable) message from the SL state machines.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 103 shows the SMP_IP state machine.

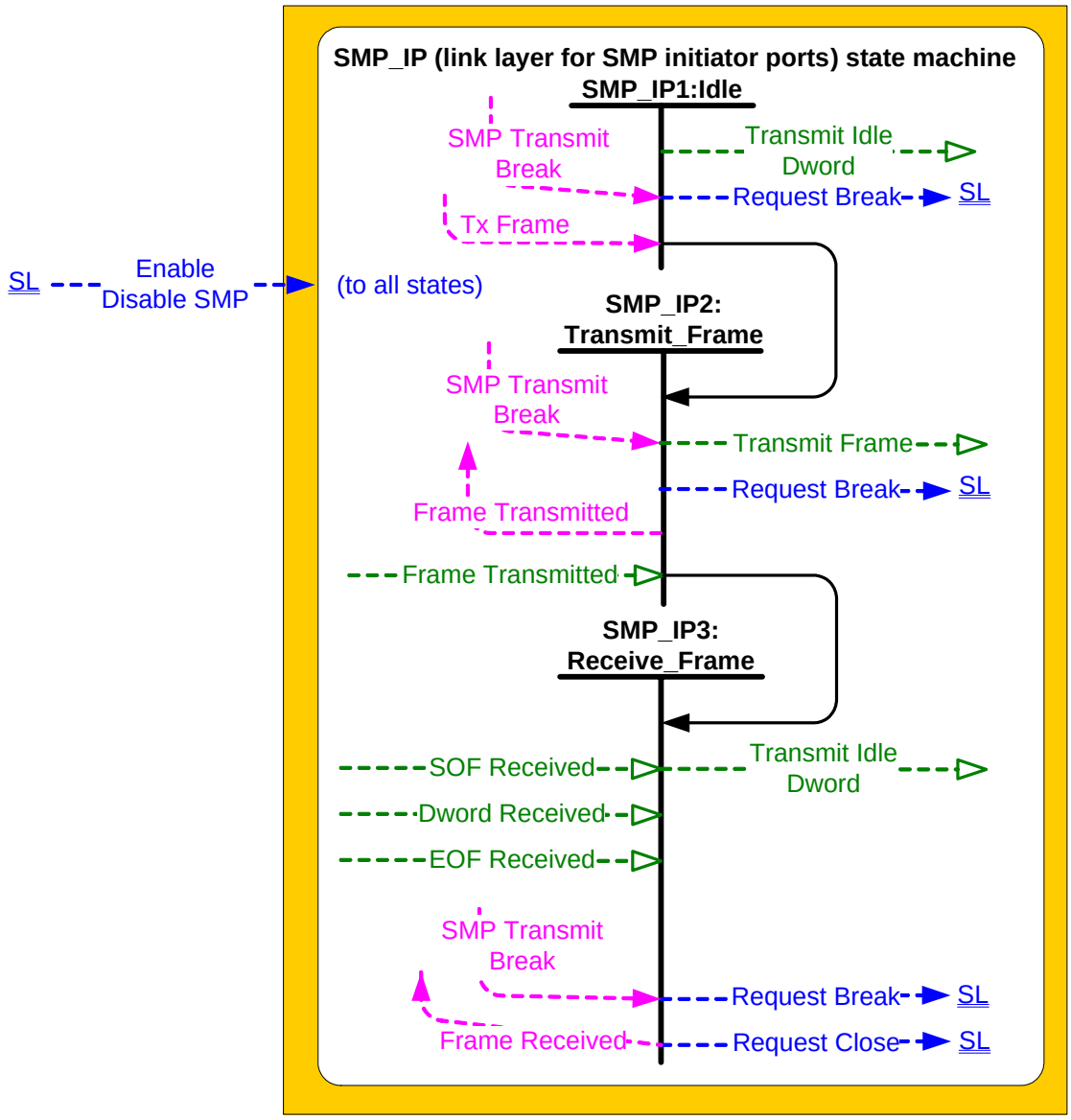


Figure 103 — SMP_IP (link layer for SMP initiator phys) state machine

7.18.4.3.2 SMP_IP1:Idle state

7.18.4.3.2.1 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, this state shall send a Request Break message to the SL state machines (see 7.14).

7.18.4.3.2.2 Transition SMP_IP1:Idle to SMP_IP2:Transmit_Frame

This transition shall occur after a Tx Frame request is received.

--`,,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.18.4.3.3 SMP_IP2:Transmit_Frame state

7.18.4.3.3.1 State description

This state shall send a Transmit Frame message to the SMP transmitter.

If an SMP Transmit Break request is received, this state shall send a Request Break message to the SL state machines (see 7.14) and terminate.

After the Frame Transmitted message is received, this state shall send a Frame Transmitted confirmation to the port layer.

7.18.4.3.3.2 Transition SMP_IP2:Transmit_Frame to SMP_IP3:Receive_Frame

This transition shall occur after sending a Frame Transmitted confirmation to the port layer.

7.18.4.3.4 SMP_IP3:Receive_Frame state

This state checks the SMP response frame and determines if the SMP response frame was successfully received (e.g., no CRC error).

If the SMP response frame is received with a CRC error, this state shall send a Frame Received (SMP Failure) confirmation to the port layer.

If the number of dwords between the SOF and EOF of the SMP response frame is less than 2, or the number of dwords after an SOF is greater than 258, this state shall send a Frame Received (SMP Failure) confirmation to the port layer. If the SMP response frame is received with no CRC error and the SMP response frame is valid, this state shall:

a) send a Frame Received confirmation to the port layer; and
b) send a Request Close message to the SL state machines (see 7.14).

If an SMP Transmit Break request is received, this state shall send a Request Break message to the SL state machines and terminate.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

7.18.4.4 SMP_TP (link layer for SMP target ports) state machine

7.18.4.4.1 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:

a) SMP_TP1:Receive_Frame (see 7.18.4.4.2)(initial state); and
b) SMP_TP2:Transmit_Frame (see 7.18.4.4.3).

The SMP_TP 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 7.14).

The SMP_TP state machine shall terminate after receiving an Enable Disable SMP (Disable) message from the SL state machines.

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 104 shows the SMP_TP state machine.

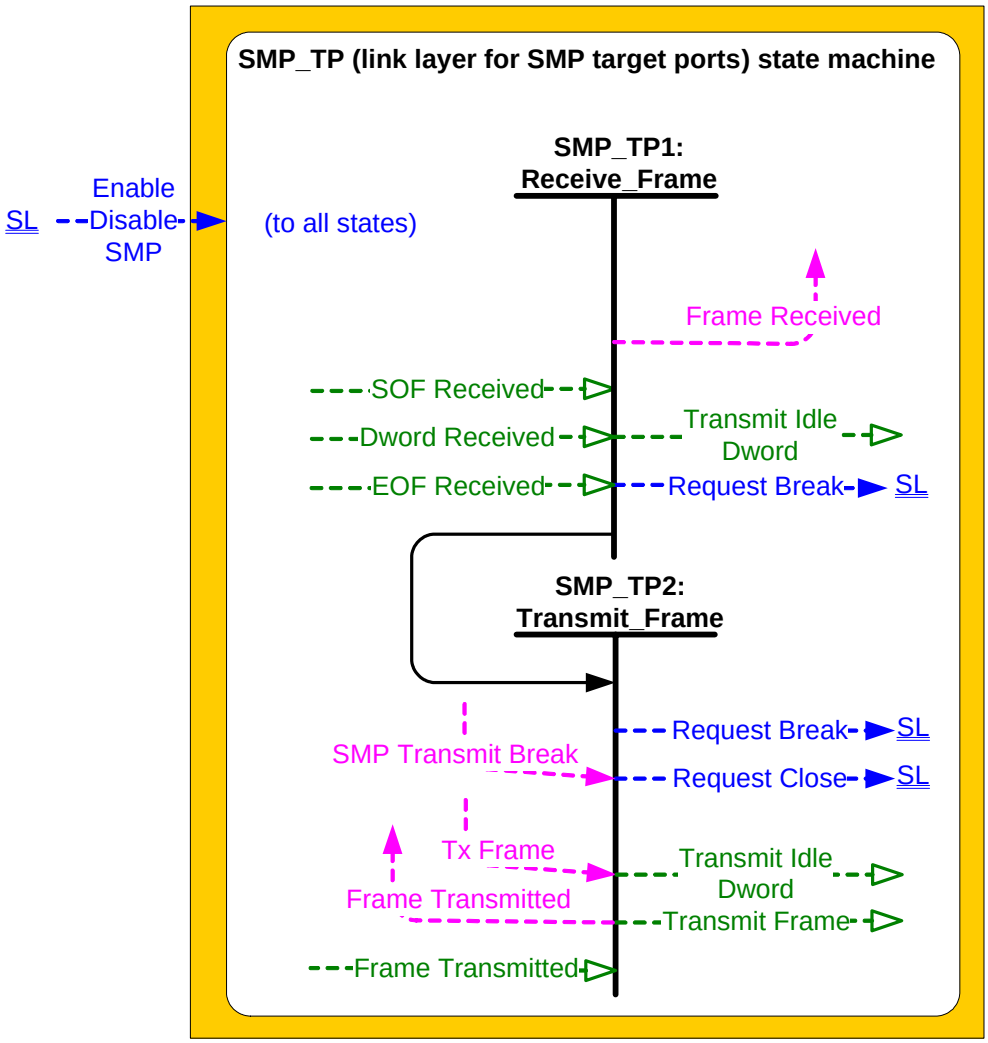


Figure 104 — SMP_TP (link layer for SMP target phys) state machine

7.18.4.4.2 SMP_TP1:Receive_Frame state

7.18.4.4.2.1 State description

This state waits for an SMP frame and determines if the SMP frame was successfully received (e.g., no CRC error).

If an SMP frame is received, this state shall send a Request Break message to the SL state machines (see 7.14) and terminate if:

a) the SMP frame has a CRC error;
b) the number of data dwords between the SOF and EOF is less than 2; or
c) the number of data dwords after the SOF is greater than 258.

Otherwise, this state shall send a Frame Received confirmation to the port layer.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

7.18.4.4.2.2 Transition SMP_TP1:Receive_Frame to SMP_TP2:Transmit_Frame

This transition shall occur after sending a Frame Received confirmation to the port layer.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.18.4.4.3 SMP_TP2:Transmit_Frame state

If this state receives an SMP Transmit Break request, this state shall send a Request Break message to the SL state machines and terminate.

If this state receives a Tx Frame request, this state shall send a Transmit Frame message to the SMP transmitter; then wait for a Frame Transmitted message. After receiving a Frame Transmitted message, this state shall send a Request Close message to the SL state machines (see 7.14) and terminate.

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.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8 Port layer

8.1 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.

8.2 PL (port layer) state machines

8.2.1 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 8.2.2); and
b) PL_PM (port layer phy manager) state machines (see 8.2.3).

There is one PL_OC state machine per port. There is one PL_PM state machine for each phy contained in the port. Phys are assigned to ports by the management application layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 105 shows examples of the port layer state machines and their interaction with the transport and link layers.

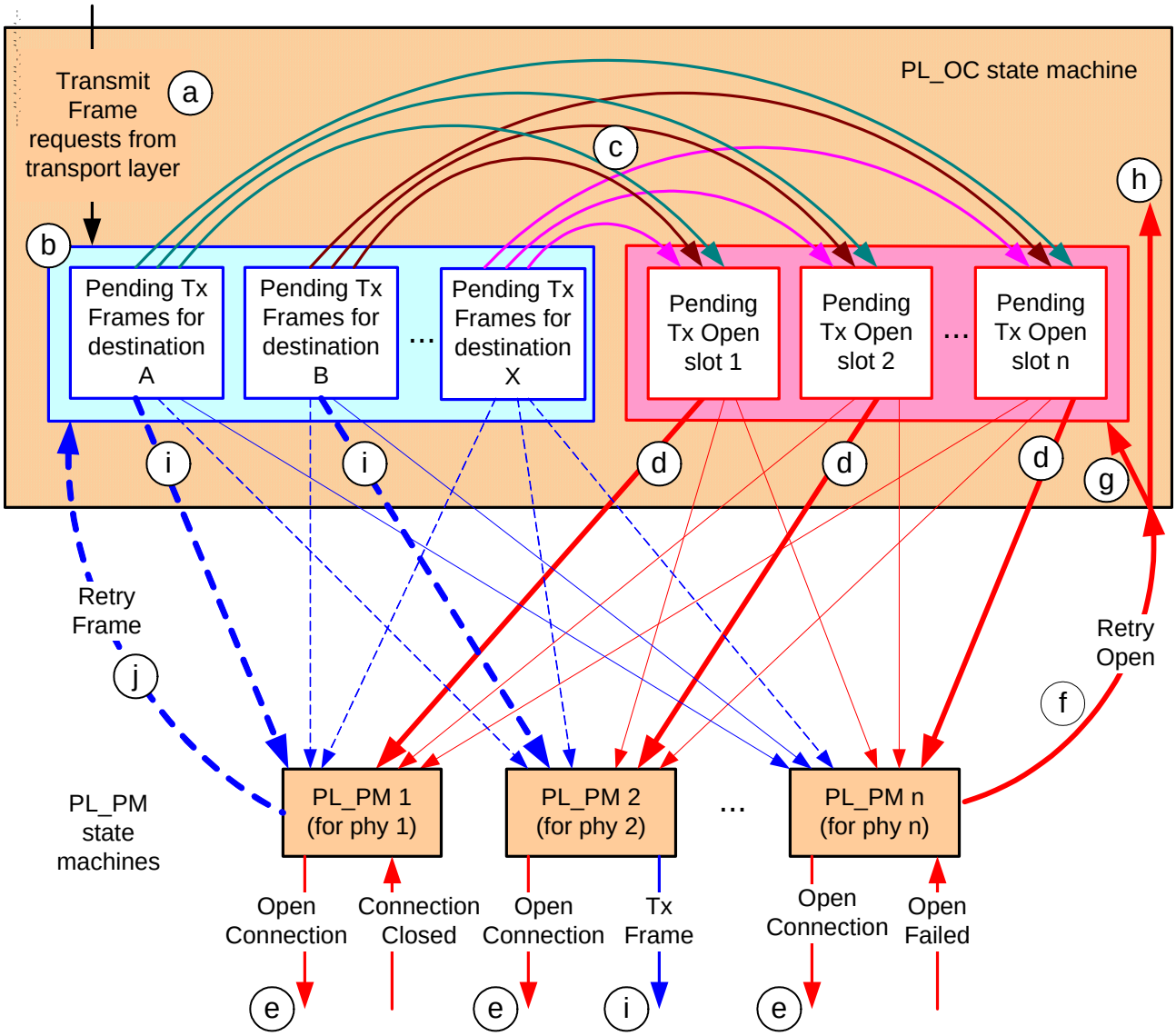


Figure 105 — Port layer examples

The following is a description of the example processes in figure 105. 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 phys);
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;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

g) the PL_OC state machine converts a Retry Open message to a pending Tx Open message if there is a pending Tx Open slot available; if the PL_OC state machine converts a Retry Open message into a pending Tx Open message, then the pathway blocked count and arbitration wait time context from the Retry Open message are applied to the pending Tx Open message;
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. 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 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) tag;
b) destination SAS address; and
c) source SAS address.

8.2.2 PL_OC (port layer overall control) state machine

8.2.2.1 PL_OC state machine overview

A 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; and
i) has I_T Nexus Loss timers.

This state machine consists of the following states:

a) PL_OC1:Idle (see 8.2.2.2) (initial state); and
b) PL_OC2:Overall_Control (see 8.2.2.3).

After power on this state machine shall start in the PL_OC1:Idle state.

This state machine shall maintain the timers listed in table 87.

Table 87 — PL_OC state machine timers

Timer	Initial value
I_T Nexus Loss timer	The value in the I_T NEXUS LOSS TIME field in the Protocol-Specific Port Control mode page (see 10.2.6.2).
Arbitration Wait Time timer	0000h, a vendor specific value less than 8000h (see 7.12.3), or the value received with a Retry Open message.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 106 shows the PL_OC state machine.

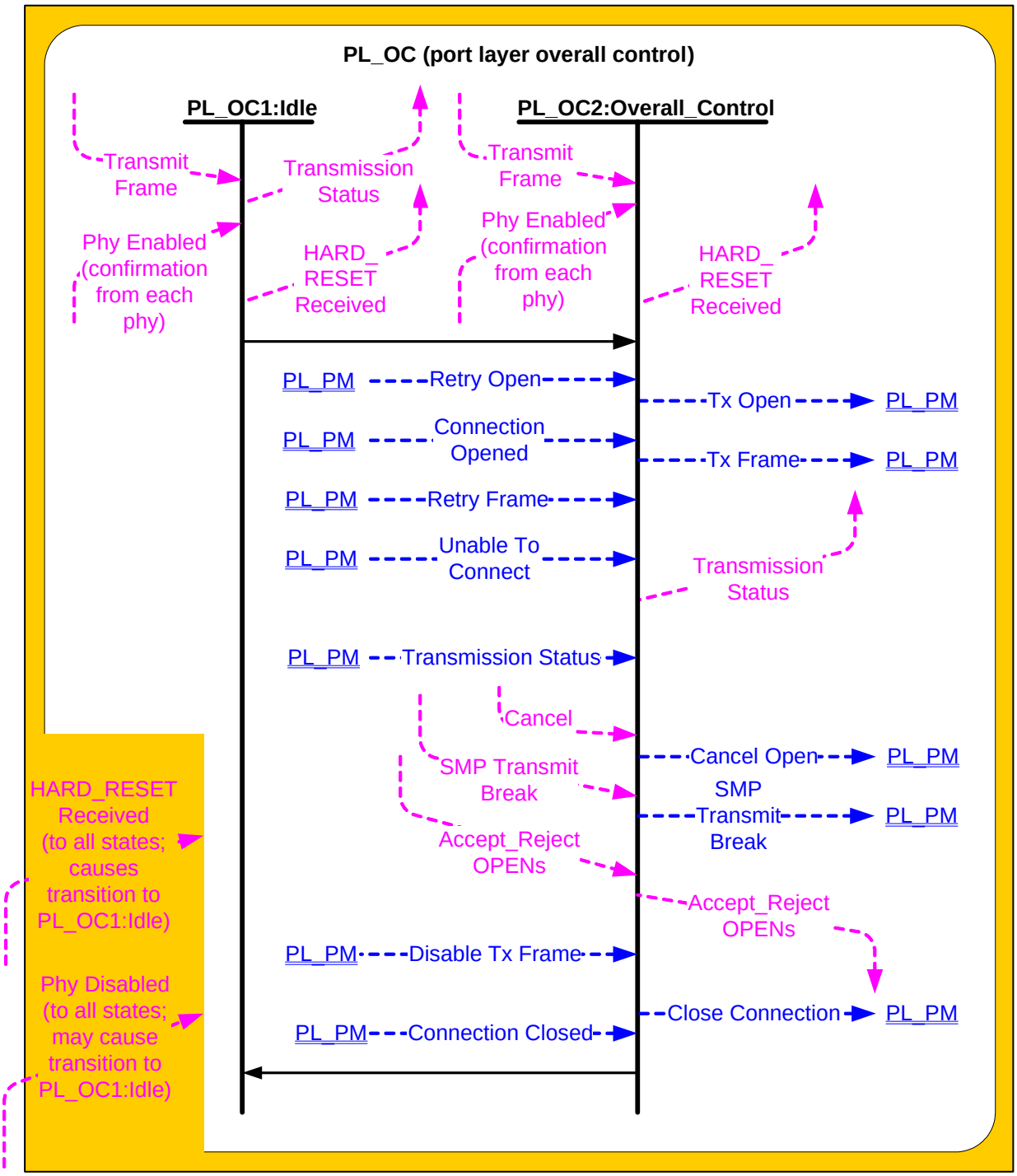


Figure 106 — PL_OC (port layer overall control) state machine

8.2.2.2 PL_OC1:Idle state

8.2.2.2.1 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

If this state receives a Transmit Frame request, then this state shall send a Transmission Status (No Phys In Port) confirmation to the transport layer.

8.2.2.2.2 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.

8.2.2.3 PL_OC2:Overall_Control state

8.2.2.3.1 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;
b) this state machine is in an SSP target port;
c) the Protocol-Specific Port Control mode page is implemented by the SSP target port; and
d) the I_T nexus loss time is not 0000h.

When this state creates an I_T Nexus Loss timer it shall:

a) initialize the I_T Nexus Loss timer; and
b) 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 and 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 Control mode page for the SSP target port (see 10.2.6.2) as the initial value for its I_T Nexus Loss timer.

Other SAS ports may detect an I_T nexus loss in a vendor-specific manner.

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 discard all pending Tx Frame messages and delete all I_T Nexus Loss timers and send a HARD_RESET Received confirmation to the transport layer.

8.2.2.3.2 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.

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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.

If this state receives a Retry Open (Retry) message, then this state shall process the Retry Open message.

If this state receives a Retry Open (No Destination) 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 Control 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 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), Retry Open (Open Timeout Occurred), or 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 this state receives a Retry Open (No Destination) or a Retry Open (Open Timeout Occurred) message, an I_T Nexus Loss timer has been created for the destination SAS address, and 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 and the I_T nexus loss time is not set to FFFFh, then this state shall start the timer;
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 8.2.2.3.4).

If this state receives a Retry Open (Pathway Blocked) message, an I_T Nexus Loss timer has been created for the destination SAS address, and 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 8.2.2.3.4).

If this state receives a Retry Open (Retry) 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) this state has fewer pending Tx Open messages than the number of PL_PM state machines (i.e., the number of phys in the port);
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) this state has fewer pending Tx Open messages than the number of PL_PM state machines.

This state shall have no more pending Tx Open messages than the number of PL_PM state machines.

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., the current number of pending Tx Open messages equals the number of phys), then this state shall discard the Retry

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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 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, then, if this state has a pending Tx Open slot available, this state may create a pending Tx Open message from the Retry Open message.

NOTE 23 - 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, this state should send a Tx Open message for the pending Tx Open message that has been pending for the longest time first.

If this state creates a pending Tx Open message from one of the following messages:

a) a Retry Open (Opened By Destination);
b) a Retry Open (Opened By Other); or
c) a Retry Open (Pathway Blocked),

then this state shall:

a) create an Arbitration Wait Time timer for the pending Tx Open message;
b) set the Arbitration Wait Time timer for the pending Tx Open message to the arbitration wait time argument from the Retry Open message; and
c) 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 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; or
d) a Retry Open (Retry) message,

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 7.12.3).

If a pending Tx Open message was created as the result this state receiving a Retry Open (Pathway Blocked) message, then this state shall set the pathway blocked count argument in the Tx Open message to the value of the pathway blocked count argument received with the message plus one, unless the pathway blocked count received with the argument is FFh.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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; or
c) a Retry Open (Pathway Blocked) message;

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 messages 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 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.

8.2.2.3.3 PL_OC2:Overall_Control state connection established

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:

a) stop the I_T Nexus Loss timer for the SAS address (if the timer has been running); and
b) initialize the I_T Nexus Loss timer.

8.2.2.3.4 PL_OC2:Overall_Control state unable to establish a connection

If this state receives a Retry Open (No Destination), Retry Open (Open Timeout Occurred), or Retry Open (Pathway Blocked) message and the I_T Nexus Loss timer for the SAS address has expired, then this state shall perform the following:

a) delete the I_T Nexus Loss timer for the SAS address;
b) 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 (Cancel Acknowledge) message from a PL_PM state machine in response to the Cancel Open message, 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.

If this state receives a Retry Open (No Destination), Retry Open (Open Timeout Occurred), or Retry Open (Pathway Blocked) message and processes it as an Unable To Connect message, or this state receives an Unable To Connect message, then this state shall send a Transmission Status confirmation as defined in table 88.

--`,,`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 88 — Confirmations from Unable To Connect or Retry Open messages

Message received	Confirmation to be sent to transport layer
Retry Open (No Destination)	Transmission Status (I_T Nexus Loss) if the I_T Nexus Loss timer for the SAS address has expired, or Transmission Status (No Destination) if it has not
Retry Open (Open Timeout Occurred)	Transmission Status (I_T Nexus Loss) if the I_T Nexus Loss timer for the SAS address has expired, or Transmission Status (Open Timeout Occurred) if it has not
Retry Open (Pathway Blocked)	Transmission Status (I_T Nexus Loss) if the I_T Nexus Loss timer for the SAS address has expired
Unable to Connect (Bad Destination)	Transmission Status (Bad Destination)
Unable To Connect (Break Received)	Transmission Status (Break Received)
Unable To Connect (Connection Rate Not Supported)	Transmission Status (Connection Rate Not Supported)
Unable To Connect (Port Layer Request)	Transmission Status (Cancel Acknowledge)
Unable To Connect (Protocol Not Supported)	Transmission Status (Protocol Not Supported)
Unable To Connect (STP Resources Busy)	Transmission Status (STP Resources Busy)
Unable To Connect (Wrong Destination)	Transmission Status (Wrong Destination)

If this state receives an Unable To Connect (Connection Rate Not Supported), Unable To Connect (Protocol Not Supported), or Unable To Connect (STP Resources Busy) message and an I_T Nexus Loss timer is running for the 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 discard the pending Tx Frame message for which the Transmission Status confirmation was sent.

8.2.2.3.5 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 message 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, the PM_OC state shall ignore the request.

If this state receives a Connection Closed 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 state is in 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 state is in 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 a message from this state, then this state shall send a Close Connection message to the PL_PM state machine. If this state is

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

in 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.

8.2.2.3.6 PL_OC2:Overall_Control state frame transmission

In order to prevent livelocks, If this state is in 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 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 that has established a connection with the destination SAS address when the initiator and protocol arguments match those in the Tx Frame message.

This state may send a Tx Frame message containing a COMMAND frame for a destination SAS address to a PL_PM state machine while waiting for one of the following messages for Tx Frame messages containing COMMAND frames for the same destination SAS address from different PL_PM state machines:

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).

This state shall not send a Tx Frame message containing a TASK frame for a task that only affects an I_T_L_Q nexus (e.g., an ABORT TASK or QUERY TASK task management function (see SAM-3)) until this state has received one of the following messages for each Tx Frame message with 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).

This state shall not send a Tx Frame message containing a TASK frame for a task that only affects an I_T_L nexus (e.g., an ABORT TASK SET, CLEAR TASK SET, CLEAR ACA, or LOGICAL UNIT RESET task management function (see SAM-3)) until this state has received one of the following messages for each Tx Frame message with the same I_T_L 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).

This state shall not send a Tx Frame message containing a TASK frame for a task that only affects an I_T nexus until this state has received one of the following messages for each Tx Frame message with the same I_T 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).

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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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 tag value 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.

Once this state has sent a Tx Frame message containing a DATA frame to a PL_PM state machine, this state shall not send a Tx Frame message containing a DATA frame with the same I_T_L_Q to another PL_PM state machine until this state has received one of the following messages for each Tx Frame message containing a DATA frame 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).

Read DATA frames and write DATA frames for the same I_T_L_Q nexus may be transmitted and received simultaneously on the same or different phys.

If this state is in 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 state is in an SMP target 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. See 7.18 for additional information about SMP connections.

Characteristics of STP connections are defined by SATA (also see 7.17).

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.

8.2.2.3.7 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) the destination SAS address; and
b) the tag.

If this state receives a Cancel request and a Tx Frame message for the Transmit Frame request has not been sent to a PL_PM state machine, then this state shall:

a) discard the Transmit Frame request; and
b) send a Transmission Status (Cancel Acknowledge) confirmation to the transport layer.

If this state receives a Cancel request and a Tx Frame message for the Transmit Frame request has been sent to a PL_PM state machine, then this state shall discard the request.

8.2.2.3.8 Transition PL_OC2:Overall_Control to PL_OC1:Idle

This transition shall occur after:

a) sending a HARD_RESET Received confirmation to the link layer; or
b) a Phy Disabled confirmation is received from all of the link layers in the port;

8.2.3 PL_PM (port layer phy manager) state machine

8.2.3.1 PL_PM state machine overview

A PL_PM state machine:

a) receives messages from the PL_OC state machine;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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 8.2.3.2) (initial state);
b) PL_PM2:Req_Wait (see 8.2.3.3);
c) PL_PM3:Connected (see 8.2.3.4); and
d) PL_PM4:Wait_For_Close (see 8.2.3.5).

After power on this state machine shall start in the PL_PM1:Idle state.

This state machine shall maintain the timers listed in Table 89.

Table 89 — PL_PM state machine timers

Timer	Initial value
Bus Inactivity Time Limit timer	The value in the BUS INACTIVITY TIME LIMIT field in the Disconnect-Reconnect mode page (see 10.2.6.1).
Maximum Connect Time Limit timer	The value in the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 10.2.6.1).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 107 shows part 1 of the PL_PM state machine.

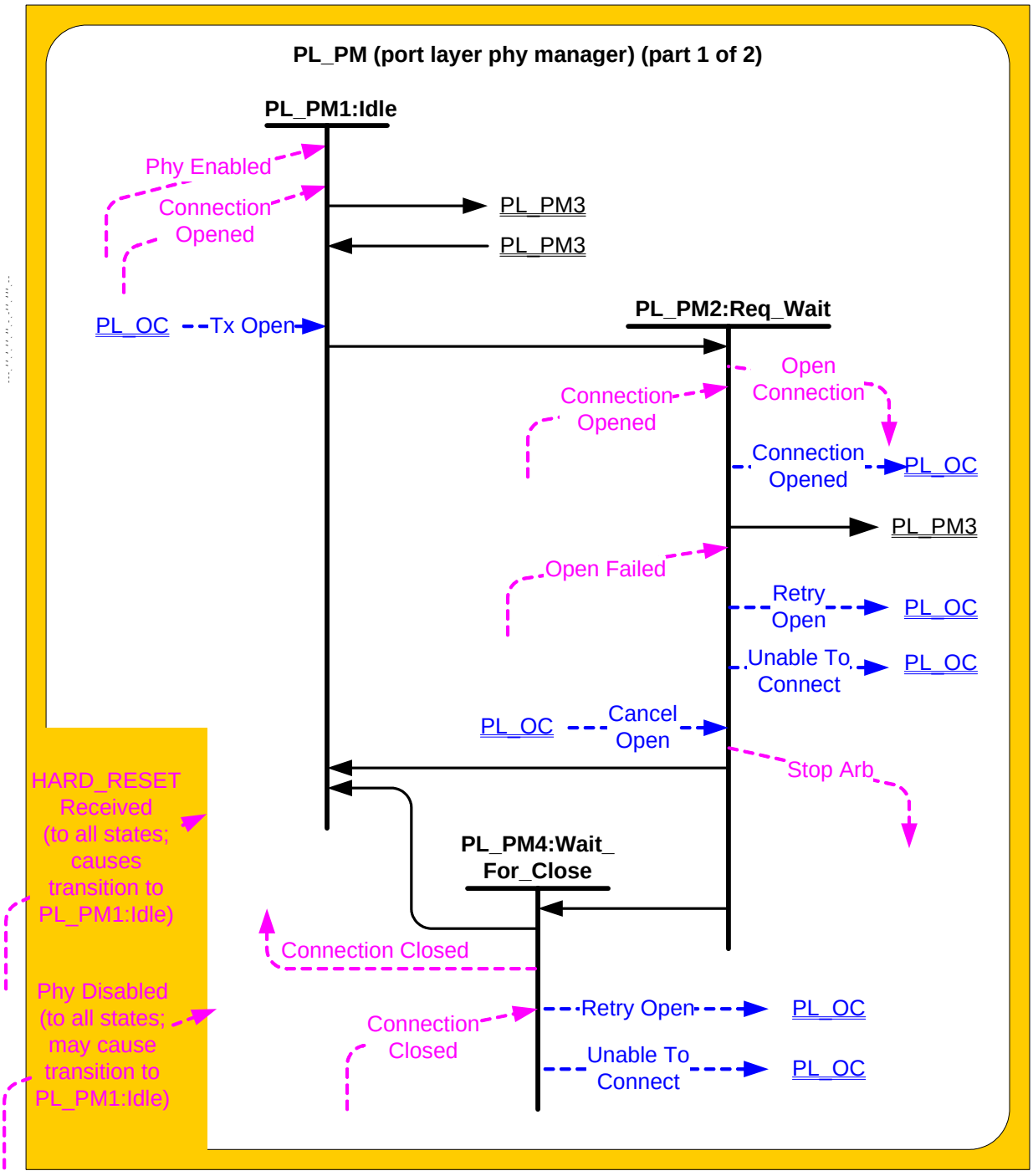


Figure 107 — PL_PM (port layer phy manager) state machine (part 1)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 108 shows part 2 of the PL_PM state machine.

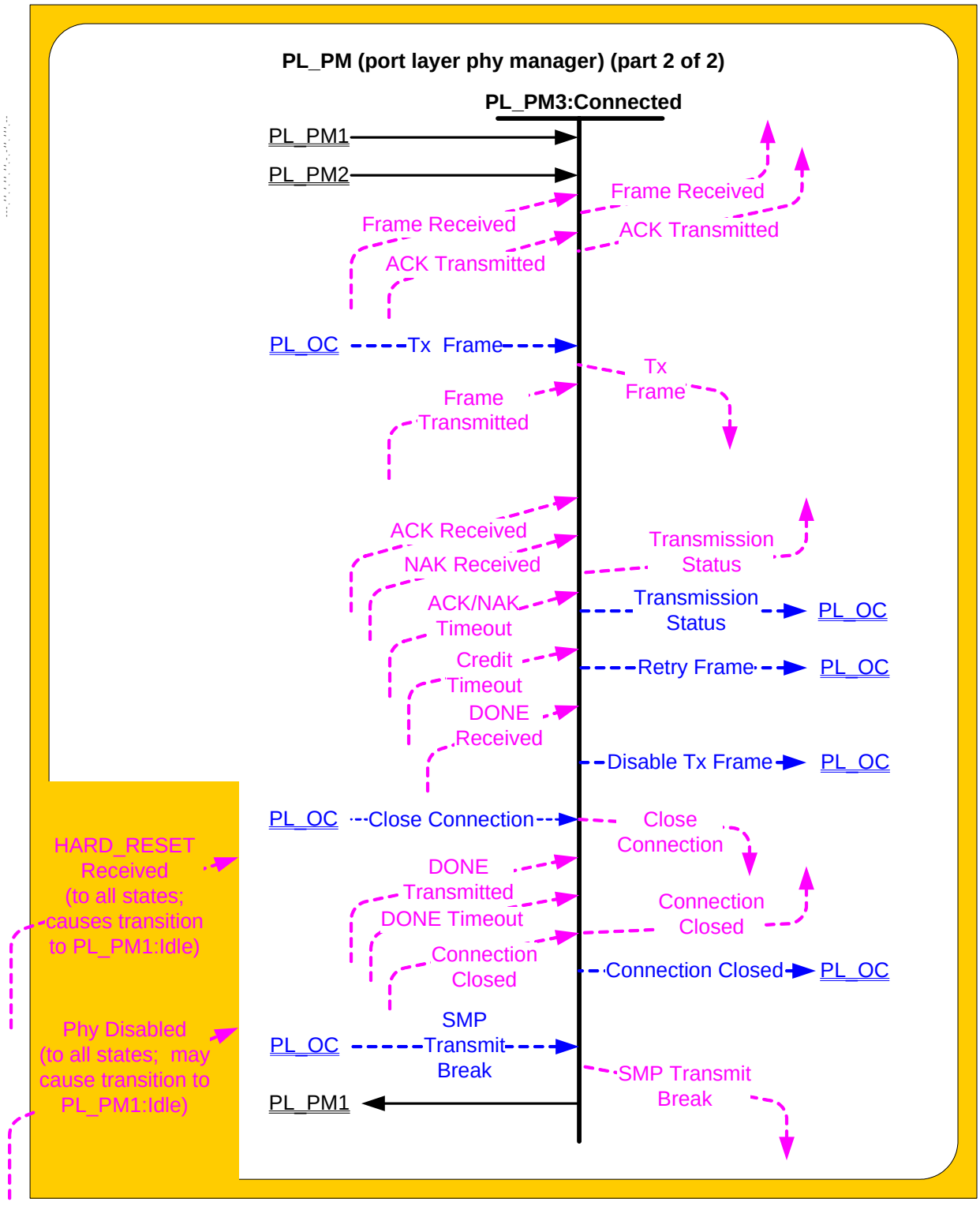


Figure 108 — PL_PM (port layer phy manager) state machine (part 2)

8.2.3.2 PL_PM1:Idle state

8.2.3.2.1 PL_PM1:Idle state description

This is the initial state of the PL_PM state machine.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

8.2.3.2.2 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.

8.2.3.2.3 Transition PL_PM1:Idle to PL_PM3:Connected

This transition shall occur after a Connection Opened confirmation is received.

8.2.3.3 PL_PM2:Req_Wait state

8.2.3.3.1 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 Received confirmation, then this state shall terminate all operations.

8.2.3.3.2 PL_PM2:Req_Wait establishing a connection

After this state receives a Tx Open message, this state shall:

a) create an Arbitration Wait Time timer;
b) initialize the Arbitration Wait Time timer to the arbitration wait time argument received with the Tx Open message;
c) start the Arbitration Wait Time timer; and
d) 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 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.

8.2.3.3.3 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 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;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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.

8.2.3.3.4 PL_PM2:Req_Wait unable to establish a connection

If this state receives an Open Failed confirmation, then this state shall send either a Retry Open message or an Unable To Connect message to the PL_OC state machine.

Table 90 defines the message to be sent for each Open Failed confirmation.

Table 90 — Messages from Open Failed confirmations

Confirmation received	Message to be sent to PL_OC
Open Failed (Pathway Blocked)	Retry Open (Pathway Blocked)
Open Failed (Retry)	Retry Open (Retry)
Open Failed (No Destination)	Retry Open (No Destination)
Open Failed (Bad Destination)	Unable To Connect (Bad Destination)
Open Failed (Break Received)	Unable To Connect (Break Received)
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 (STP Resources Busy)	Unable To Connect (STP Resources Busy)
Open Failed (Wrong Destination)	Unable To Connect (Wrong Destination)

A Retry Open message shall include the following arguments:

a) initiator 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.

8.2.3.3.5 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.

8.2.3.3.6 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 a HARD_RESET Received confirmation is received; or

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

d) a Phy Disabled confirmation is received.

8.2.3.3.7 Transition PL_PM2:Req_Wait to PL_PM3:Connected

This transition shall occur after a Connection Opened confirmation is received.

8.2.3.3.8 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); or
b) an Open Failed (Port Layer Request).

8.2.3.4 PL_PM3:Connected state

8.2.3.4.1 PL_PM3:Connected state description

If the protocol for the connection is SSP, and this state is in an SSP target port, and the value in the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 10.2.6.1) is not zero, then, upon entry into this state, this state shall:

a) create a Maximum Connect Time Limit timer;
b) initialize the Maximum Connect Time Limit timer; and
c) start the Maximum Connect Time Limit timer.

If the protocol for the connection is SSP, and this state is in an SSP target port, and the value in the MAXIMUM CONNECT TIME LIMIT field in the Disconnect-Reconnect mode page (see 10.2.6.1) is zero, then this state shall not create a Maximum Connect Time Limit timer (i.e., there is no maximum connect time limit).

Other SAS ports may implement a Maximum Connect Time Limit timer in a vendor-specific manner.

If the protocol for the connection is SSP, and this state is in an SSP target port, and the value in the BUS INACTIVITY TIME LIMIT field in the Disconnect-Reconnect mode page (see 10.2.6.1) is not zero, then, upon entry into this state, this state shall:

a) create a Bus Inactivity Time Limit timer;
b) initialize the Bus Inactivity Time Limit timer; and
c) not start the Bus Inactivity Time Limit timer.

If the protocol for the connection is SSP, and this state is in an SSP target port, and the value in the BUS INACTIVITY TIME LIMIT field in the Disconnect-Reconnect mode page (see 10.2.6.1) is zero, then this state shall not create a Bus Inactivity Time Limit timer (i.e., there is no maximum bus inactivity time limit).

Other SAS ports may implement a Bus Inactivity Time Limit timer in a vendor-specific manner.

If a Bus Inactivity Time Limit timer has been created and this state receives a Tx Frame message, then this state shall:

a) stop the Bus Inactivity Time Limit timer, if it is running; and
b) initialize the Bus Inactivity Time Limit timer.

If this state receives a Tx Frame message, 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 state is in 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 has been created and this state receives an ACK Received or NAK Received confirmation, then this state shall start the Bus Inactivity Time Limit timer. If the Bus Inactivity Time Limit timer expires before this state receives a Tx Frame message, then this state shall send a Close Connection request to the link layer.

If a Maximum Connect Time Limit timer has been created and this state receives an ACK Received or NAK Received confirmation, then this state shall check the Maximum Connect Time Limit timer. If the Maximum Connect Time Limit timer has expired, then this state shall send a Close Connection request to the link layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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.

If this state receives a Frame Transmitted confirmation, then this state shall send a Transmission Status (Frame Transmitted) confirmation to the transport layer.

If this state receives an ACK Received confirmation, then this state shall send a Transmission Status (ACK Received) confirmation to the transport layer.

If this state receives a NAK Received confirmation, then this state shall send a Transmission Status (NAK Received) confirmation to the transport layer.

If this state receives an ACK/NAK Timeout confirmation, then this state shall send a Transmission Status (ACK/NAK Timeout) confirmation to the transport layer.

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 a Connection Closed 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 Transmission Status (Connection Lost Without ACK/NAK) confirmation to the transport layer.

If this state receives a Connection Closed confirmation after sending a Transmit 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 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 bit;
b) protocol;
c) connection rate;
d) initiator connection tag;
e) destination SAS address;
f) source SAS address; and
g) frame.

If this state receives a DONE Received (ACK/NAK Timeout) or DONE Transmitted, then this state shall send a Disable Tx Frames message to the PL_OC state machine.

If this state receives a Connection Closed confirmation, then this state shall send a Connection Closed message to the PL_OC state machine. If this state receives a Connection Closed confirmation during an SMP connection, this state shall send a Connection Closed confirmation to the transport layer.

If this state receives a HARD_RESET Received confirmation, then this state machine shall terminate all operations.

8.2.3.4.2 Transition PL_PM3:Connected to PL_PM1:Idle

This transition shall occur after:

a) a Connection Closed message is sent to the PL_OC state machine; or
b) all operations are terminated after a HARD_RESET Received confirmation is received.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8.2.3.5 PL_PM4:Wait_For_Close state

8.2.3.5.1 PL_PM4:Wait_For_Close state description

After this state receives a Connection Closed confirmation, and 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. The Retry Open message shall include the following arguments:

a) initiator 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.

If this state receives a Connection Closed confirmation during an SMP connection, this state shall send a Connection Closed confirmation to the transport layer.

After this state receives a Connection Closed confirmation, and if this state was entered after the PL_PM2:Req_Wait state received an Open Failed (Port Layer Request) 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 (Port Layer Request) message to the PL_OC state machine.

The 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.

If this state receives a HARD_RESET Received confirmation, then this state shall terminate all operations.

8.2.3.5.2 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 a HARD_RESET Received confirmation is received.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9 Transport layer

9.1 Transport layer overview

The transport layer defines frame formats. Transport layer state machines interface to the application layer and port layer and construct and parses frame contents. For SSP, the transport layer only receives frames from the port layer that are going to be ACKed by the link layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9.2 SSP transport layer

9.2.1 SSP frame format

Table 91 defines the SSP frame format.

Table 91 — SSP frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	FRAME TYPE							
1	(MSB)							
3	HASHED DESTINATION SAS ADDRESS							(LSB)
4	Reserved							
5	(MSB)							
7	HASHED SOURCE SAS ADDRESS							(LSB)
8	Reserved							
9	Reserved							
10	Reserved						RETRANSMIT	Reserved
11	Reserved						NUMBER OF FILL BYTES	
12	Reserved							
13								
15	Reserved							
16	(MSB)							
17	TAG							(LSB)
18	(MSB)							
19	TARGET PORT TRANSFER TAG							(LSB)
20	(MSB)							
23	DATA OFFSET							(LSB)
24								
m	INFORMATION UNIT							
	Fill bytes, if needed							
n - 3	(MSB)							
n	CRC							(LSB)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 92 defines the FRAME TYPE field, which defines the format of the INFORMATION UNIT field.

Table 92 — FRAME TYPE **field**

Code	Name of frame	Information unit	Originator	Information unit size (bytes)	Reference
01h	DATA frame	Data	SSP initiator port or SSP target port	1 to 1 024	9.2.2.4
05h	XFER_RDY frame	Transfer ready	SSP target port	12	9.2.4.3
06h	COMMAND frame	Command	SSP initiator port	28 to 284	9.2.4.1
07h	RESPONSE frame	Response	SSP target port	24 to 1 024	9.2.4.5
16h	TASK frame	Task management function	SSP initiator port	28	9.2.4.2
F0h - FFh	Vendor specific				
All others	Reserved				

The HASHED DESTINATION SAS ADDRESS field contains the hashed value (see 4.2.3) of the destination SAS address. See 9.2.6.2.5 and 9.2.6.3.2 for transport layer requirements on checking this field.

The HASHED SOURCE SAS ADDRESS field contains the hashed value of the source SAS address. See 9.2.6.2.5 and 9.2.6.3.2 for transport layer requirements on checking this field.

The RETRANSMIT bit is set to one for RESPONSE frames under certain conditions (see 9.2.4.5) and shall be set to zero for all other frame types. This bit indicates the frame is a retransmission after the SSP target port failed in its previous attempt to transmit the frame.

The NUMBER OF FILL BYTES field indicates the number of fill bytes between the INFORMATION UNIT field and the CRC field. The NUMBER OF FILL BYTES field shall be set to zero for all frame types except DATA frames (i.e., all other frame types are already four-byte aligned).

The TAG field contains a value that allows the SSP initiator port to establish a context for commands and task management functions.

For COMMAND and TASK frames, the SSP initiator port shall set the TAG field to a value that is unique for the I_T nexus established by the connection (see 7.12). An SSP initiator port shall not reuse the same tag when transmitting COMMAND or TASK frames to different LUNs in the same SSP target port; it may reuse a tag when transmitting frames to different SSP target ports. The TAG field in a COMMAND frame contains the tag defined in SAM-3. The TAG field in a TASK frame does not correspond to a SAM-3 tag, but corresponds to an SAM-3 association (see 10.2.1). The tag space used in the TAG fields is shared across COMMAND and TASK frames (e.g., if a tag is used for a COMMAND frame, it is not simultaneously used for a TASK frame).

For DATA, XFER_RDY, and RESPONSE frames, the SSP target port shall set the TAG field to the tag of the command or task management function to which the frame pertains.

The TARGET PORT TRANSFER TAG field provides an additional optional method for an SSP target port to establish a write data context when receiving DATA frames while it has multiple XFER_RDY frames outstanding. SSP target ports may set the TARGET PORT TRANSFER TAG field to any value when transmitting a frame. SSP target ports that use this field should set the field in every XFER_RDY frame to a value that is unique for the L_Q portion of the I_T_L_Q nexus.

SSP initiator ports shall set the TARGET PORT TRANSFER TAG field as follows:

a) For each DATA frame that is sent in response to a 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 DATA frame that is sent containing first burst data (see 9.2.2.4), the SSP initiator port shall set the TARGET PORT TRANSFER TAG field to FFFFh; and

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

c) For frames other than 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 9.2.2.4. 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. The maximum size of the INFORMATION UNIT field is 1 024 bytes, making the maximum size of the frame 1 052 bytes (1 024 bytes of data + 24 bytes of header + 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 indicated by the NUMBER OF FILL BYTES field. The contents of the fill bytes are vendor specific.

The CRC field contains a CRC value (see 7.5) 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 EOF). The CRC field is checked by the link layer (see 7.16), not the transport layer.

9.2.2 Information units

9.2.2.1 COMMAND information unit

Table 93 defines the command IU. The COMMAND frame is sent by an SSP initiator port to request that a command be processed by a device server in a logical unit.

Table 93 — COMMAND information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7	LOGICAL UNIT NUMBER							(LSB)
8	Reserved							
9	Reserved					TASK ATTRIBUTE		
10	Reserved							
11	ADDITIONAL CDB LENGTH (n dwords)						Reserved	
12								
27	CDB							
28								
27+n×4	ADDITIONAL CDB BYTES							

The LOGICAL UNIT NUMBER field contains the address of the logical unit. The structure of the logical unit number field shall be as defined in SAM-3. If the addressed logical unit does not exist, the task manager shall follow the rules for selection of invalid logical units defined in SAM-3.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The TASK ATTRIBUTE field is defined in table 94.

Table 94 — TASK ATTRIBUTE **field**

Code	Task attribute	Description
000b	SIMPLE	Requests that the task be managed according to the rules for a simple task attribute (see SAM-3).
001b	HEAD OF QUEUE	Requests that the task be managed according to the rules for a head of queue task attribute (see SAM-3).
010b	ORDERED	Requests that the task be managed according to the rules for an ordered task attribute (see SAM-3).
011b	Reserved	
100b	ACA	Requests that the task be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-3).
101b-111b	Reserved	

The ADDITIONAL CDB LENGTH field contains the length in dwords (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 between the end of the CDB and the end of 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 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-3).

9.2.2.2 TASK information unit

Table 95 defines the task management function IU. The TASK frame is sent by an SSP initiator port to request that a task management function be processed by a task manager in a logical unit.

Table 95 — TASK information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7				LOGICAL UNIT NUMBER				(LSB)
8				Reserved				
9				Reserved				
10				TASK MANAGEMENT FUNCTION				
11				Reserved				
12	(MSB)							
13				TAG OF TASK TO BE MANAGED				(LSB)
14								
27				Reserved				

The LOGICAL UNIT NUMBER field contains the address of the logical unit. The structure of the logical unit number field shall be as defined in SAM-3. If the addressed logical unit does not exist, the task manager shall return a

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and its RESPONSE CODE field set to INVALID LOGICAL UNIT.

Table 96 defines the TASK MANAGEMENT FUNCTION field.

Table 96 — Task management functions

Code	Task management function	Uses LOGICAL UNIT NUMBER field	Uses TAG OF TASK TO BE MANAGED field	Description
01h	ABORT TASK	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 TAG OF TASK TO BE MANAGED field (see SAM-3).
02h	ABORT TASK SET	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-3).
04h	CLEAR TASK SET	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-3).
08h	LOGICAL UNIT RESET	yes	no	The task manager shall perform the LOGICAL UNIT RESET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-3).
20h	Reserved [a]			
40h	CLEAR ACA	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-3).
80h	QUERY TASK	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 TAG OF TASK TO BE MANAGED field (see SAM-3).
All others	Reserved			

[a] The TARGET RESET task management function defined in SAM-3 is not supported.

If TASK MANAGEMENT FUNCTION contains a reserved or unsupported value, 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 TASK MANAGEMENT FUNCTION is set to ABORT TASK or QUERY TASK, the TAG OF TASK TO BE MANAGED field specifies the TAG value from the COMMAND frame that contained the task to be aborted or checked. For all other task management functions, the TAG OF TASK TO BE MANAGED field shall be ignored.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.2.2.3 XFER_RDY information unit

Table 97 defines the transfer ready IU. The XFER_RDY frame is sent by an SSP target port to request write data from the SSP initiator port.

Table 97 — XFER_RDY information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)			REQUESTED OFFSET				
3								(LSB)
4	(MSB)			WRITE DATA LENGTH				
7								(LSB)
8				Reserved				
11								

The REQUESTED OFFSET field contains the application client buffer offset of the segment of write data the SSP initiator port may transmit to the logical unit (using DATA frames). The requested offset shall be a multiple of four (i.e., each DATA frame shall begin transferring data on a dword boundary). The REQUESTED OFFSET field shall be zero for the first XFER_RDY frame of a command unless the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 10.2.6.1.5) is not set to zero.

In the initial XFER_RDY frame for a given command, the SSP target port shall set the requested offset to the value indicated by the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 10.2.6.1.5). If any additional XFER_RDY frames are required, the REQUESTED OFFSET field shall be set to the value of the previous XFER_RDY frame's requested offset plus the previous XFER_RDY frame's write data length.

The WRITE DATA LENGTH field contains the number of bytes of write data the SSP initiator port may transmit to the logical unit (using DATA frames) from the application client 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 zero, the SSP target port shall set the WRITE DATA LENGTH field to a value less than or equal to the value in the MAXIMUM BURST SIZE field (see 10.2.6.1.4).

If an SSP target port transmits a XFER_RDY frame containing a WRITE DATA LENGTH field that is not divisible by four, 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).

9.2.2.4 DATA information unit

Table 98 defines the data IU. The DATA frame is sent by an SSP initiator port to deliver write data and is sent by an SSP target port to deliver read data. The maximum size of the data IU is the maximum size of any IU in an SSP frame (see 9.2.1). The minimum size of the data IU is one byte.

Table 98 — DATA information unit

Byte\Bit	7	6	5	4	3	2	1	0
0				DATA				
n-1								

The DATA field contains the read or write data.

An SSP initiator port shall only transmit a DATA frame:

a) in response to an XFER_RDY frame; or

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) after transmitting a COMMAND frame if the FIRST BURST SIZE field in the Disconnect-Reconnect mode page is not zero (see 10.2.6.1.5).

If the value in the MAXIMUM BURST SIZE field on the Disconnect-Reconnect mode page is not zero, 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 10.2.6.1.4).

The DATA frame shall only contain write data for a single XFER_RDY frame.

If an SSP target port transmits a DATA frame containing a non-zero value in the NUMBER OF FILL BYTES field in the frame header (see 9.2.1), the SSP target port shall not transmit any subsequent DATA frames for that command (i.e., only the last read DATA frame for a command may have data with a length that is not a multiple of four).

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 a XFER_RDY. An SSP initiator port shall set the NUMBER OF FILL BYTES field in the frame header (see 9.2.1) to zero in all other DATA frames that it transmits.

NOTE 24 - Combined with the restrictions on WRITE DATA LENGTH in the XFER_RDY frame (see 9.2.2.3), 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 initiator port shall not transmit a DATA frame for a given I_T_L_Q nexus after it has sent a TASK frame that terminates that task (e.g., an ABORT TASK).

The DATA OFFSET field in the frame header (see 9.2.1) contains the application client buffer offset as described by SAM-3. The data offset shall be a multiple of four (i.e., each DATA frame shall transfer data beginning on a dword boundary).

The initial read DATA frame for a given command shall set the DATA OFFSET field to zero. If any additional read DATA frames are required, the DATA OFFSET field shall be set to the value of the previous read DATA frame's data offset plus the previous read DATA frame's data length.

The initial write DATA frame for a given command shall set the DATA OFFSET field to zero. If any additional write DATA frames are required, the DATA OFFSET field shall be set to the value of the previous write DATA frame's data offset plus the previous write DATA frame's data length.

9.2.2.5 RESPONSE information unit

9.2.2.5.1 RESPONSE information unit overview

Table 99 defines the response IU. The RESPONSE frame is sent by an SSP target port to deliver SCSI status (e.g., GOOD or CHECK CONDITION) and sense data, or to deliver SSP-specific status (e.g., illegal frame

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

format). The maximum size of the RESPONSE frame is the maximum size of any IU in an SSP frame (see 9.2.1).

Table 99 — RESPONSE information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
9								
10	Reserved						DATAPRES	
11	STATUS							
12	Reserved							
15								
16	(MSB)	SENSE DATA LENGTH (n bytes)						
19								(LSB)
20	(MSB)	RESPONSE DATA LENGTH (m bytes)						
23								(LSB)
24	RESPONSE DATA							
23+m								
24+m	SENSE DATA							
23+m+n								

Table 100 defines the DATAPRES field, which indicates the format and content of the STATUS field, SENSE DATA LENGTH field, RESPONSE DATA LENGTH field, RESPONSE DATA field, and SENSE DATA field.

Table 100 — DATAPRES **field**

Code	Name	Description	Reference
00b	NO_DATA	No data present	9.2.2.5.2
01b	RESPONSE_DATA	Response data present	9.2.2.5.3
10b	SENSE_DATA	Sense data present	9.2.2.5.4
11b	Reserved		

The SSP target port shall return a RESPONSE frame with the DATAPRES field set to NO_DATA if a command completes without sense data to return.

The SSP target port shall return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA in response to every TASK frame and in response to errors that occur while the transport layer is processing a COMMAND frame.

The SSP target port shall return a RESPONSE frame with the DATAPRES field set to SENSE_DATA if a command 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.

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.2.2.5.2 RESPONSE information unit NO_DATA format

If the DATAPRES field is set to NO_DATA, then:

a) the STATUS field shall contain the status code for a command that has ended (see SAM-3 for a list of status codes);
b) the SENSE DATA LENGTH field and the RESPONSE DATA LENGTH field shall be set to zero and shall be ignored by the SSP initiator port; and
c) the SENSE DATA field and the RESPONSE DATA field shall not be present.

9.2.2.5.3 RESPONSE information unit RESPONSE_DATA format

If the DATAPRES field is set to RESPONSE_DATA, then:

a) the STATUS field and the SENSE DATA LENGTH field shall be set to zero and shall be ignored by the SSP initiator port;
b) the SENSE DATA field shall not be present;
c) the RESPONSE DATA LENGTH field shall be set to four. Other lengths are reserved for future standardization; and
d) the RESPONSE DATA field shall be present.

Table 101 defines the RESPONSE DATA field, which contains information describing protocol failures detected during processing of a request received by the SSP target port. The RESPONSE DATA field shall be present if the SSP target port detects any of the conditions described by a non-zero RESPONSE CODE value and shall be present for a RESPONSE frame sent in response to a TASK frame.

Table 101 — RESPONSE DATA **field**

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							
3	RESPONSE CODE							

Table 102 defines the RESPONSE CODE field, which indicates the error condition or the completion status of a task management function. See 10.2.1.5 and 10.2.1.13 for the mapping of these response codes to SCSI service responses.

Table 102 — 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	INVALID LOGICAL UNIT NUMBER [a]
All others	Reserved

[a] Only valid when responding to a TASK frame

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.2.2.5.4 RESPONSE information unit SENSE_DATA format

If the DATAPRES field is set to SENSE_DATA, then:

a) the STATUS field shall contain the status code for a command that has ended (see SAM-3 for a list of status codes);
b) the RESPONSE DATA LENGTH field shall be set to zero and shall be ignored by the initiator;
c) the RESPONSE DATA field shall not be present;
d) the SENSE DATA LENGTH field shall be set 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 larger than 1 000 (see table 92); and
e) the SENSE DATA field shall contain sense data (see SAM-3).

9.2.3 Sequences of SSP frames

Figure 109, figure 110, figure 111, and figure 112 show examples of the sequences of frames for single task management functions and commands. Frames may be interleaved in any order when multiple commands and/or task management functions are outstanding. Frames may be transmitted during one or more connections (e.g., the COMMAND frame could be transmitted in a connection originated by the SSP initiator port, and the DATA frames and RESPONSE frame transmitted in one or more connections originated by the SSP target port). 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).

Figure 109 shows the sequence of SSP frames for a task management function, including the transport protocol services (see 10.2.1) invoked by the SCSI application layer.

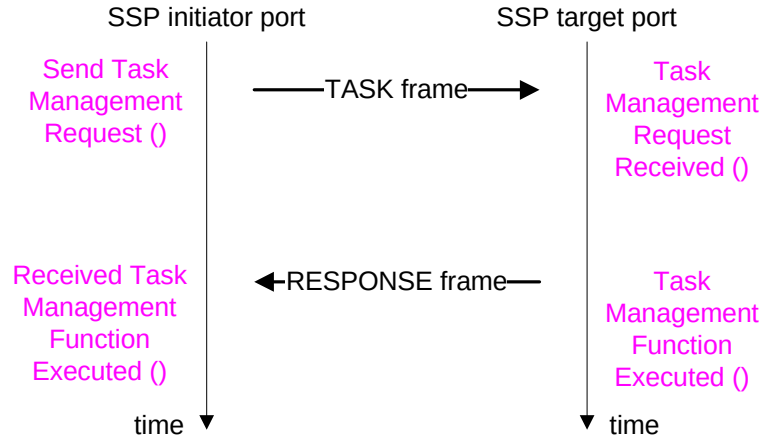


Figure 109 — Task management function sequence of SSP frames

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 110 shows the sequence of SSP frames for a write command, including the transport protocol services (see 10.2.1) invoked by the SCSI application layer.

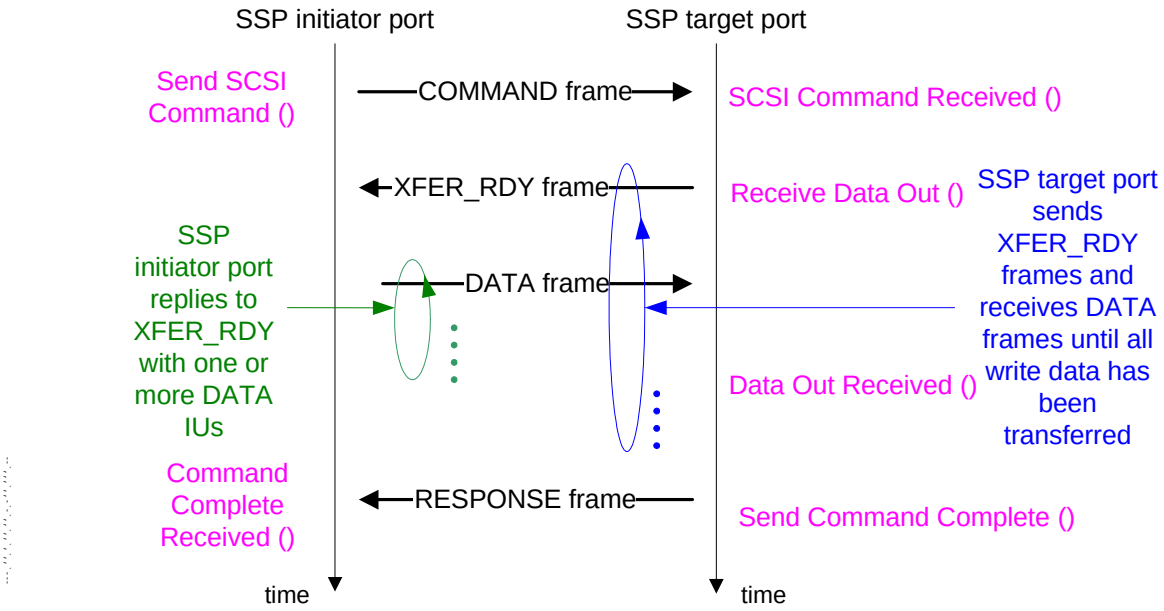


Figure 110 — Write command sequence of SSP frames

Figure 111 shows the sequence of SSP frames for a read command, including the transport protocol services (see 10.2.1) invoked by the SCSI application layer.

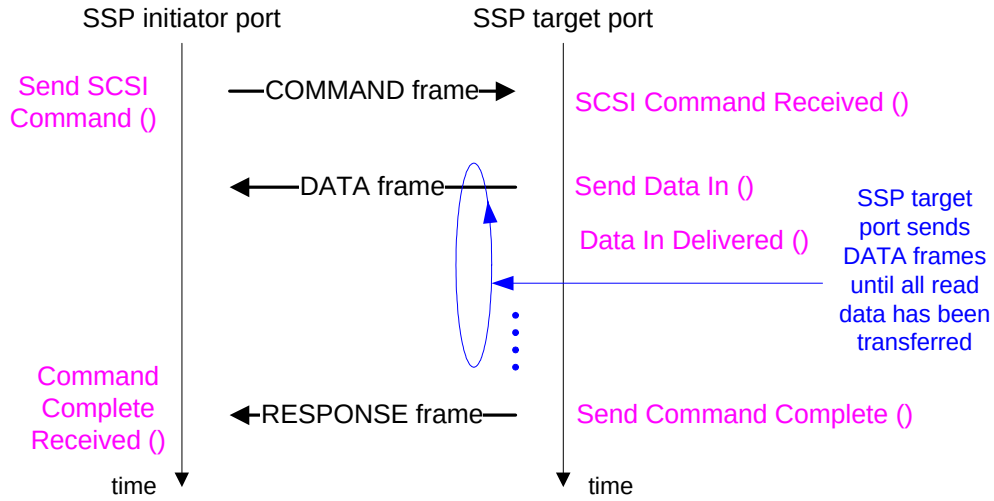


Figure 111 — Read command sequence of SSP frames

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 112 shows the sequence of SSP frames for a bidirectional command, including the transport protocol services (see 10.2.1) invoked by the SCSI application layer.

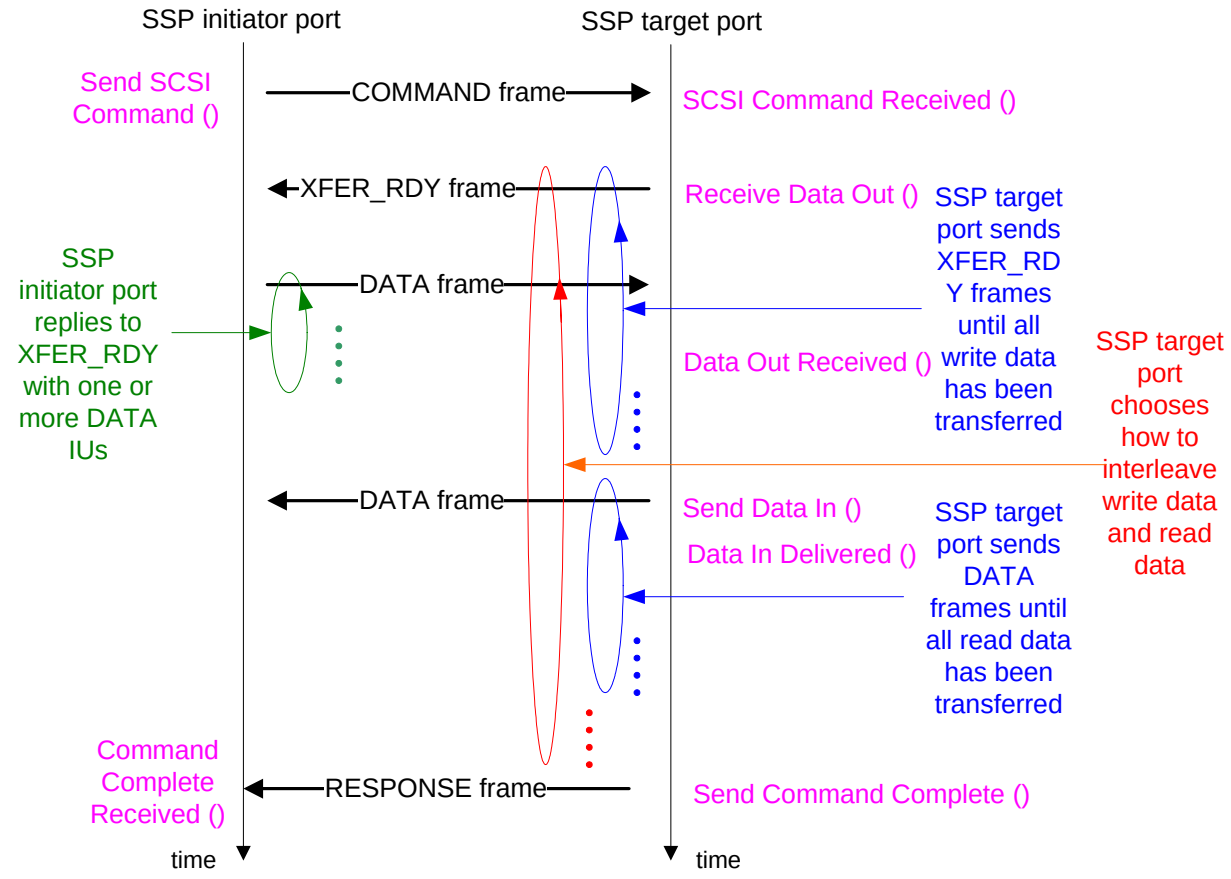


Figure 112 — Bidirectional command sequence of SSP frames

9.2.4 SSP transport layer handling of link layer errors

9.2.4.1 COMMAND frame

If an SSP initiator port transmits a COMMAND frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it shall transmit a QUERY TASK task management function in the next connection to determine whether the command was received. If QUERY TASK returns a TASK MANAGEMENT FUNCTION SUCCEEDED response, the SSP initiator port shall assume the COMMAND frame was ACKed. If QUERY TASK returns a TASK MANAGEMENT FUNCTION COMPLETE response, and a RESPONSE frame has not yet been received for that I_T_L_Q nexus, the SSP initiator port shall assume the command was NAKed or lost and may reuse the tag (see 10.2.2).

9.2.4.2 TASK frame

If an SSP initiator port transmits a TASK frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it shall retransmit the TASK frame in a new connection (see 10.2.2).

9.2.4.3 XFER_RDY frame

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT (see 10.2.3).

--`,,`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

If an SSP target port transmits an XFER_RDY frame and receives a NAK, it shall return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.2.3).

9.2.4.4 DATA frame

If an SSP target port transmits a DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT (see 10.2.3).

If an SSP target port transmits a DATA frame and receives a NAK, it shall return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.2.3).

If an SSP initiator port transmits a DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall abort the command (see 10.2.2).

If an SSP initiator port transmits a DATA frame and receives a NAK, it shall abort the command (see 10.2.2).

9.2.4.5 RESPONSE frame

If an SSP target port transmits a RESPONSE frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall try transmitting the RESPONSE frame again in a new connection. It shall do this at least one time. The RETRANSMIT bit shall be set to one on each of the retries (see 9.2.6.3.3.8).

If an SSP target port transmits a RESPONSE frame and receives a NAK, it shall retry transmitting the RESPONSE frame at least one time (see 9.2.6.3.3.8).

If an SSP initiator port receives a RESPONSE frame with a RETRANSMIT bit set to one, and it has previously received a RESPONSE frame for the same I_T_L_Q nexus, it shall discard the extra RESPONSE frame. If it has not previously received the RESPONSE frame, it shall treat it as the valid RESPONSE frame (see 10.2.2).

9.2.5 SSP transport layer error handling

9.2.5.1 SSP target port error handling

If an SSP target port receives an XFER_RDY frame or an unsupported frame type, it shall discard the frame (see 9.2.6.3.2).

If an SSP target port receives a COMMAND frame and:

a) the frame is too short to contain a LUN field;
b) the frame is too short to contain a CDB; or
c) the ADDITIONAL CDB LENGTH field indicates the frame should be a different length,

the SSP target port shall return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME (see 9.2.6.3.3.8).

If an SSP target port receives a TASK frame that is too short, it shall return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME (see 9.2.6.3.3.8).

If an SSP target port receives a COMMAND frame with a tag that is already in use, it may return a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of OVERLAPPED COMMANDS DETECTED (see 10.2.1.3).

If an SSP target port receives a TASK frame with a tag that is already in use, it may return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME (see 9.2.6.3.2).

If an SSP target port receives a DATA frame with an unknown tag, it shall discard the frame (see 9.2.6.3.2).

If an SSP target port receives a DATA frame that does not contain first burst data and for which there is no XFER_RDY frame outstanding, it shall discard the frame (see 9.2.6.3.2).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If an SSP target port receives a TASK frame with an unknown logical unit number, it shall return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID LOGICAL UNIT (see 9.2.6.3.2).

If an SSP target port receives a COMMAND frame or TASK frame with a target port transfer tag set to a value other than FFFFh, it may return a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME (see 9.2.6.3.2).

If an SSP target port is using target port transfer tags and it receives a DATA frame with an unknown target port transfer tag, it shall discard the frame (see 9.2.6.3.3).

If an SSP target port receives a DATA frame with a data offset that was not expected, it shall discard that frame and any subsequent DATA frames received for that command and, shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of DATA OFFSET ERROR (see 10.2.3).

If an SSP target port receives a DATA frame with more write data than expected (i.e., the length of the DATA frame extends past the end of the expected write data length), it shall discard the frame and terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of TOO MUCH WRITE DATA (see 10.2.3).

If an SSP target port receives a zero length DATA frame, it shall discard the frame and terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of INFORMATION UNIT TOO SHORT (see 10.2.3).

9.2.5.2 SSP initiator port error handling

If an SSP initiator port receives a COMMAND or TASK frame or an unsupported frame type, it shall discard the frame (see 9.2.6.2.5).

If an SSP initiator port receives a DATA, XFER_RDY, or RESPONSE frame with an unknown TAG field value (including a tag for which it has sent a COMMAND or TASK frame but not yet received an ACK), it shall discard the frame. It may then abort the command with that tag (see 9.2.6.2.5).

If an SSP initiator port receives an XFER_RDY frame that is not 12 bytes long, it shall discard the frame. It may then abort the command (see 10.2.2).

If an SSP initiator port receives an XFER_RDY frame in response to a command with no write data, it shall discard the frame. It shall then abort the command (see 10.2.2).

If an SSP initiator port receives an XFER_RDY frame requesting more write data than expected, it shall abort the command (see 10.2.2).

If an SSP initiator port receives an XFER_RDY frame requesting zero bytes, it shall abort the command (see 10.2.2).

If an SSP initiator port receives an XFER_RDY frame with a requested offset that was not expected, it shall abort the command (see 10.2.2).

If an SSP initiator port receives a DATA frame with more read data than expected, it shall discard the frame and abort the command (see 10.2.2). It may receive a RESPONSE for the command before being able to abort the command.

If an SSP initiator port receives a DATA frame with zero bytes, it shall discard the frame and abort the command (see 10.2.2). It may receive a RESPONSE for the command before being able to abort the command.

If an SSP initiator port receives a DATA frame with a data offset that was not expected, it shall discard that frame and any subsequent DATA frames received for that command and abort the command (see 10.2.2). It may receive a RESPONSE for the command before being able to abort the command.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.2.6 ST (transport layer for SSP ports) state machines

9.2.6.1 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 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) the tag;
b) the destination SAS address; and
c) the source SAS address;

and are used to route the confirmations to the correct ST state machines.

9.2.6.2 ST_I (transport layer for SSP initiator ports) state machines

9.2.6.2.1 ST_I state machines overview

The ST_I state machines are as follows:

a) ST_ISF (initiator send frame) state machine (see 9.2.6.2.2);
b) ST_IPD (initiator process data) state machine (see 9.2.6.2.3);
c) ST_IPR (initiator process response) state machine (see 9.2.6.2.4); and
d) ST_IFR (initiator frame router) state machine (see 9.2.6.2.5).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 113 shows the ST_I state machines.

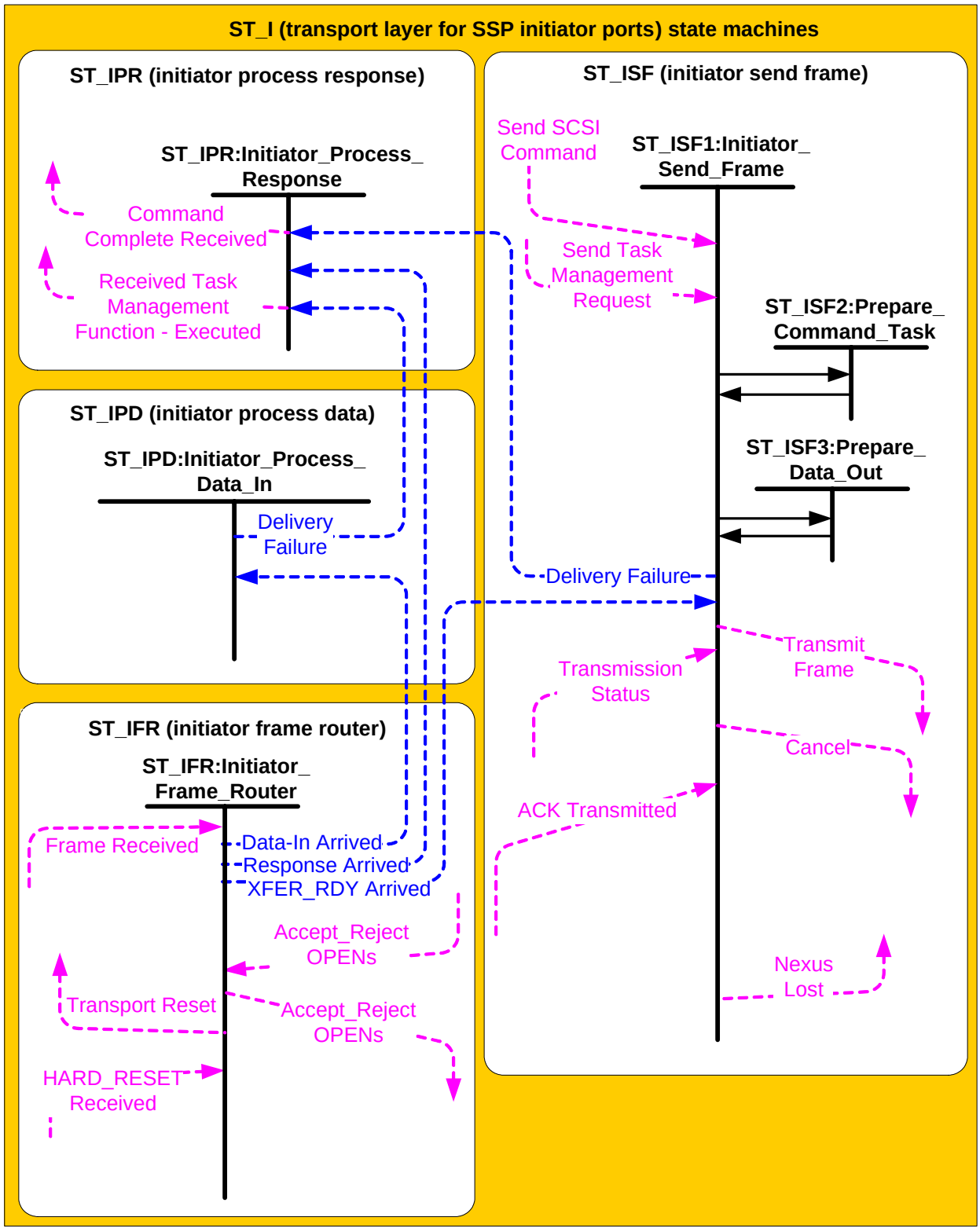


Figure 113 — ST_I (transport layer for SSP initiator ports) state machines

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9.2.6.2.2 ST_ISF (initiator send frame) state machine

9.2.6.2.2.1 ST_ISF state machine overview

The ST_ISF state machine receives transport protocol service requests from the SCSI application layer, receives XFER_RDY Arrived messages from the ST_IFR state machine, and constructs COMMAND, TASK, or data-out DATA frames. The service request may be to process either a command or task management function. This state machine also communicates with the port layer via requests and confirmations regarding frame transmission, and may communicate to the ST_IPR state machine regarding service delivery subsystem failures.

This state machine consists of the following states:

a) ST_ISF1:Send_Frame (see 9.2.6.2.2.2)(initial state);
b) ST_ISF2:Prepare_Command_Task (see 9.2.6.2.2.3); and
c) ST_ISF3:Prepare_Data_Out (see 9.2.6.2.2.4).

This state machine shall be started when a Send SCSI Command or a Send Task Management Request transport protocol service request is received from the SCSI application layer or when an XFER_RDY Arrived message is received.

9.2.6.2.2.2 ST_ISF1:Send_Frame state

9.2.6.2.2.2.1 State description

A Send SCSI Command transport protocol service request or a Send Task Management Request transport protocol service request includes the following to be used in any OPEN address frames required to service the request:

a) connection rate;
b) initiator connection tag; and
c) destination SAS address.

A Send SCSI Command transport protocol service request also includes the following to be used in any SSP frame for the request:

a) logical unit number;
b) tag;
c) task attribute;
d) additional CDB length;
e) CDB; and
f) additional CDB bytes.

If the request is for a data-out command, then the request also includes the number of bytes for the first burst size for the logical unit.

A Send Task Management Request transport protocol service request includes the following to be used in the TASK frame:

a) logical unit number;
b) tag;
c) task management function; and
d) tag of task to be managed.

If this state machine was started as the result of receiving an XFER_RDY Arrived message, then:

a) If an XFER_RDY frame is not expected for the command (e.g., for a read command), then this state shall discard the frame and shall send a Delivery Failure (XFER_RDY Not Expected) message to the ST_IPR state machine. This state machine shall terminate if it sends the message;
b) If the length of the information unit is not 12 bytes, then this state shall discard the frame and shall send a Delivery Failure (XFER_RDY Information Unit Too Short) message or Delivery Failure (XFER_RDY Information Unit Too Long) to the ST_IPR state machine. This state machine shall terminate after sending the message;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

c) If the length of the XFER_RDY information unit is 12 bytes and the write data length is zero or exceeds the amount of data remaining to be transferred for the data-out command, then this state shall send a Delivery Failure (XFER_RDY Incorrect Write Data Length) message to the ST_IPR state machine. This state machine shall terminate after sending the message; or
d) If the length of the XFER_RDY information unit is 12 bytes and the requested offset is not expected, then this state shall send a Delivery Failure (XFER_RDY Requested Offset Error) message to the ST_IPR state machine. This state machine shall terminate after sending the message.

If this state is entered from the ST_ISF2:Prepare_Command_Task state, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_ISF3:Prepare_Data_Out state, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer.

A Transmit Frame request shall include the SSP frame and the following to be used for any OPEN address frame:

a) the 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 set to the SAS address of the SSP initiator port.

After sending a Transmit Frame request to the port layer this state shall wait for a Transmission Status confirmation. If the confirmation is not Transmission Status (Frame Transmitted), then this state shall send a Delivery Failure (Service Delivery Subsystem Failure) message to the ST_IPR state machine. The Delivery Failure message shall include:

a) any argument received with the Transmission Status confirmation; and
b) I_T_L_x nexus information (i.e., destination SAS address and tag);

If the transmitted frame was a DATA frame, then this state shall transition to the ST_ISF3:Prepare_Data_Out state after receiving a Transmission Status (Frame Transmitted) confirmation if there is more data to transfer.

After receiving a Transmission Status (Frame Transmitted) confirmation, the state machine shall expect one of the following confirmations for the frame:

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 transmitted frame was a COMMAND frame or TASK frame requiring a data-out operation, then the state machine shall wait to receive a Transmission Status (ACK Received), Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK) confirmation before transitioning from this state. If the transmitted frame was a DATA frame, the state machine may transition to ST_ISF3:Prepare_Data_Out as described in 9.2.6.2.2.2.3.

If a Transmission Status (NAK Received) confirmation is received, then this state shall send a Delivery Failure (Service Delivery Subsystem Failure - NAK Received) message to the ST_IPR state machine.

If a Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK) confirmation is received, then this state shall send a Delivery Failure (Service Delivery Subsystem Failure - Connection Failed) message to the ST_IPR state machine.

After sending a Delivery Failure message to the ST_IPR state machine, this state machine shall terminate.

This state may also send a Cancel request to the port layer to cancel a previous Transmit Frame request. A Cancel request shall include the following arguments:

a) the destination SAS address; and
b) the tag.

This state machine shall terminate upon receipt of a Transmission Status (Cancel Acknowledge) confirmation.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.2.6.2.2.2 Transition ST_ISF1:Send_Frame to ST_ISF2:Prepare_Command_Task

This transition shall occur after receiving a Send SCSI Command or Send Task Management Request transport protocol service request.

9.2.6.2.2.3 Transition ST_ISF1:Send_Frame to ST_ISF3:Prepare_Data_Out

This transition shall occur after:

a) receiving a Transmission Status (ACK Received) confirmation for a COMMAND frame for a data-out operation if first burst is enabled;
b) receiving an XFER_RDY Arrived message followed by an ACK Transmitted confirmation, if the XFER_RDY frame is valid (see 9.2.6.2.2.1); or
c) receiving a Transmission Status (Frame Transmitted) confirmation for a Transmit Frame (Non-Interlocked) request if the number of data bytes that has been transmitted for the request is less than the first burst size or the write data length specified in the XFER_RDY.

9.2.6.2.3 ST_ISF2:Prepare_Command_Task state

9.2.6.2.3.1 State description

This state shall construct either a COMMAND or TASK frame.

If the frame to be constructed is a COMMAND frame, then this state shall include the following values received from the SCSI application layer in the frame:

a) logical unit number;
b) tag;
c) task attribute;
d) additional CDB length;
e) CDB; and
f) additional CDB bytes.

If the frame to be constructed is a TASK frame, then this state shall include the following values received from the SCSI application layer in the frame:

a) logical unit number;
b) tag;
c) task management function; and
d) tag of task to be managed.

This state shall generate and include the following values in either a COMMAND or TASK frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero; and
e) number of fill bytes.

9.2.6.2.3.2 Transition ST_ISF2:Prepare_Command_Task to ST_ISF1:Send_Frame

This transition shall occur after constructing a COMMAND or TASK frame.

9.2.6.2.4 ST_ISF3:Prepare_Data_Out state

9.2.6.2.4.1 State description

This state shall construct a DATA frame. This state shall include the following values in the frame, received either from the SCSI application layer or included in an XFER_RDY Arrived message:

a) logical unit number;
b) tag;
c) target port transfer tag;
d) data offset; and

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

e) data.

This state shall generate and include the following values in the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
e) number of fill bytes; and
f) fill bytes.

9.2.6.2.2.4.2 Transition ST_ISF3:Prepare_Data_Out to ST_ISF1:Send_Frame

This transition shall occur after constructing a DATA frame.

9.2.6.2.3 ST_IPD (initiator process data) state machine

The ST_IPD state machine receives and processes a message from the ST_IFR state machine containing a DATA frame.

This state machine consists of one state.

This state machine shall be started when a Data-In Arrived message is received.

This state shall check the length and data offset of the DATA information unit.

If the length of the information unit is zero, then this state shall send a Delivery Failure (DATA Incorrect Read Data Length) message to the ST_IPR state machine.

If the length of the information unit exceeds the amount of data remaining to be transferred for the data-in command, then this state shall send a Delivery Failure (DATA Too Much Read Data) message to the ST_IPR state machine.

If the data offset is not the expected offset, then this state shall send a Delivery Failure (DATA Offset Error) message to the ST_IPR state machine.

If the DATA information unit is valid, this state shall process the data-in data.

This state machine shall terminate after sending a message or processing the data-in data.

9.2.6.2.4 ST_IPR (initiator process response) state machine

The ST_IPR state machine receives a message from the ST_IFR state machine containing a RESPONSE frame or a message containing a service delivery subsystem failure from the ST_ISF state machine. This state machine processes the RESPONSE frame or the service delivery subsystem failure and sends a transport protocol service confirmation to the SCSI application layer.

This state machine consists of one state.

This state machine shall be started when a Response Arrived message is received or a Delivery Failure message is received.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If a Delivery Failure message is received, this state shall send a Command Complete Received or Received Task Management Function – Executed confirmation to the SCSI application layer with the Service Response argument set as indicated by table 103.

Table 103 — Delivery Failure to Command Complete Received mapping

Delivery Failure argument	Command Complete Received (Service Response)
XFER_RDY Information Unit Too Short	Service Delivery or Target Failure - XFER_RDY Information Unit Too Short
XFER_RDY Information Unit Too Long	Service Delivery or Target Failure - XFER_RDY Information Unit Too Long
XFER_RDY Incorrect Write Data Length	Service Delivery or Target Failure - XFER_RDY Incorrect Write Data Length
XFER_RDY Requested Offset Error	Service Delivery or Target Failure - XFER_RDY Requested Offset Error
XFER_RDY Not Expected	Service Delivery or Target Failure - XFER_RDY Not Expected
DATA Incorrect Data Length	Service Delivery or Target Failure - DATA Incorrect Data Length
DATA Too Much Read Data	Service Delivery or Target Failure - DATA Too Much Read Data
DATA Offset Error	Service Delivery or Target Failure - DATA Offset Error
Service Delivery Subsystem Failure - NAK Received	Service Delivery or Target Failure - NAK Received
Service Delivery Subsystem Failure - Connection Failed	Service Delivery or Target Failure - Connection Failed

If a Response Arrived message is received, this state shall check the length of the RESPONSE information unit.

If the length of the information unit is not correct, then this state shall send a Command Complete Received (Service Response = Service Delivery or Target Failure) or Received Task Management Function – Executed (Service Response = Service Delivery or Target Failure) confirmation to the SCSI application layer. The confirmation shall include the tag.

If the length is correct, this state shall send a Command Complete Received (Service Response = Task Complete), Command Complete Received (Service Response = Linked Command Complete), or Received Task Management Function – Executed (Service Response = Function Complete) transport protocol service confirmation. The confirmation shall also include a Retransmit argument indicating the state of the RETRANSMIT bit.

This state machine shall terminate after sending a confirmation.

9.2.6.2.5 ST_IFR (initiator frame router) state machine

The ST_IFR state machine receives confirmations from the port layer and, depending on the confirmation, may send a message to the ST_ISF, ST_IPD, or ST_IPR state machines. This state machine receives connection information from the port layer. This state machine also receives Accept_Reject OPENs requests from the SCSI application layer and sends these requests to the port layer.

This state machine consists of one state.

This state machine shall be started after:

a) an Accept_Reject OPENs request is received;

--`,,,,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) a Frame Received confirmation is received; or
c) a HARD_RESET Received confirmation is received.

If this state machine was started as the result of receiving an Accept_Reject OPENs (Accept SSP) or Accept_Reject OPENs (Reject SSP) request, then this state shall send an Accept_Reject OPENs request along with the received argument to the port layer. This state machine shall terminate after sending an Accept_Reject OPENs request to the port layer.

If this state machine was started as the result of a Frame Received (ACK/NAK Balanced) or Frame Received (ACK/NAK Not Balanced) confirmation, then this state shall check the frame type in the received frame. 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 and terminate. 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 and terminate.

If the frame type is correct relative to the confirmation, then this state may check that the hashed source SAS address matches the SAS address of the SAS port transmitting the frame and the hashed destination SAS address in the frame matches the SAS address of the SAS port receiving the frame based on the connection. If this state checks these SAS addresses and they do not match, then this state machine shall terminate.

If the frame type is:

a) XFER_RDY, then this state shall send a XFER_RDY Arrived message to the ST_ISF1:Send_Frame state specified by the tag;
b) RESPONSE, then this state shall send a Response Arrived message to the ST_IPR state machine specified by the tag; or
c) DATA, then this state shall send a Data-In Arrived message to the ST_IPD state machine specified by the tag.

Each of these messages shall contain the content of the SAS frame. If the tag does not specify a valid state machine, then this state shall discard the frame and may send a vendor-specific confirmation to the SCSI application layer to abort the command using that tag.

If this state machine was started as a result of a HARD_RESET Received confirmation, then this state shall send a Transport Reset event notification to the SCSI application layer.

This state machine shall terminate after sending a message or confirmation.

9.2.6.3 ST_T (transport layer for SSP target ports) state machines

9.2.6.3.1 ST_T state machines overview

The ST_T state machines are as follows:

a) ST_TFR (target frame router) state machine (see 9.2.6.3.2); and
b) ST_TTS (target transport server) state machine (see 9.2.6.3.3).

If implemented, this state machine shall maintain the timers listed in table 104.

Table 104 — ST_T state machine timers

Timer	Initial value
Initiator Response Timeout	The value in the INITIATOR RESPONSE TIMEOUT field in the Protocol-Specific Port Control mode page (see 10.2.6.2).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 114 shows the ST_T state machines.

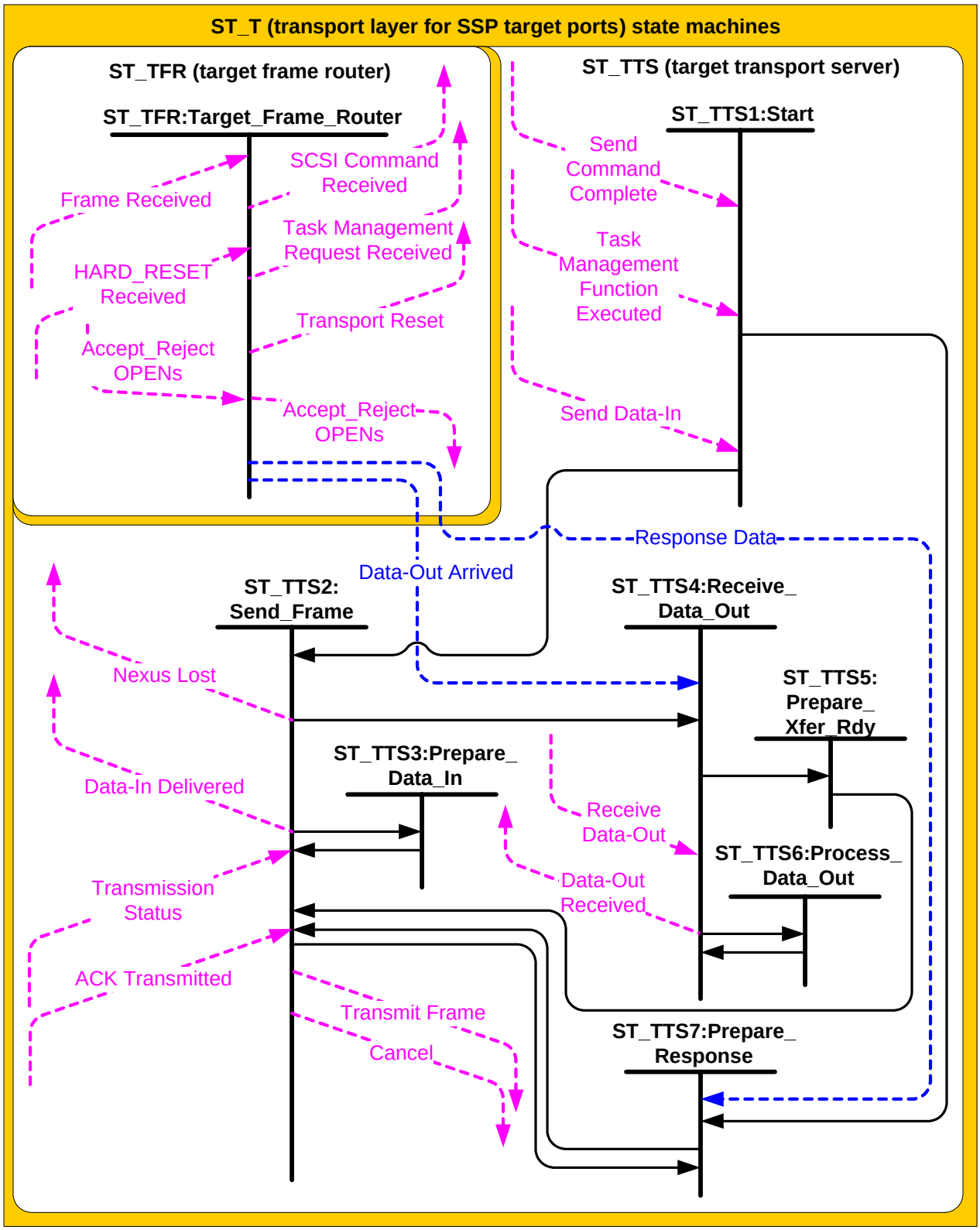


Figure 114 — ST_T (transport layer for SSP target ports) state machines

9.2.6.3.2 ST_TFR (target frame router) state machine

The ST_TFR state machine receives confirmations from the port layer and sends a transport protocol service indication to the SCSI application layer or a message to the ST_TTS state machine. This state machine also

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

receives Accept_Reject OPENs requests from the application layer and sends corresponding requests to the port layer.

This state machine consists of one state.

This state machine shall be started after:

a) an Accept_Reject OPENs request is received;
b) a Frame Received confirmation is received; or
c) a HARD_RESET Received confirmation is received.

If this state machine was started as the result of receiving an Accept_Reject OPENs (Accept SSP) or Accept_Reject OPENs (Reject SSP) request, then this state machine shall send a corresponding Accept_Reject OPENs request to the port layer. This state machine shall terminate after sending an Accept_Reject OPENs request to the port layer.

If this state machine was started as the result of receiving 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 92). If the frame type is not COMMAND, TASK, or DATA, then this state machine shall discard the frame and terminate.

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 and terminate.

This state machine may check that reserved fields in the frame are zero. If any reserved fields are not zero, then this state machine may send a Response Data (Invalid Frame) message to the ST_TTS7:Prepare_Response state including the logical unit number and tag.

NOTE 25 - This check only applies to reserved fields defined in the SSP frame formats (e.g. formats defined in this clause), not reserved fields within the CDB in a COMMAND frame. Handling checking of reserved fields in a CDB is described in SAM-3.

If the frame type is correct relative to the confirmation, then this state may check that the hashed source SAS address matches the SAS address of the SAS port transmitting the frame and the hashed destination SAS address in the frame matches the SAS address of the SAS port receiving the frame based on the connection. If this state checks these SAS addresses and they do not match, then this state machine shall discard the frame and terminate.

If the frame type is DATA, and the tag does not match a tag for an outstanding data-out command, then this state machine shall discard the frame and terminate.

If the frame type is DATA, and the tag matches a tag for an outstanding data-out command without first burst data for which no XFER_RDY frame is outstanding, then this state machine shall discard the frame and terminate.

If the frame type is COMMAND, then this state machine shall check the length of the information unit. If the length of the information unit is not correct (see 9.2.2.2), then this state machine shall send a Response Data (Invalid Frame) message to the ST_TTS7:Prepare_Response state including the logical unit number and tag.

If the frame type is TASK, then this state machine shall check the length of the information unit. If the length of the information unit is not correct (see 9.2.2.2), then this state machine shall send a Response Data (Invalid Frame) message to the ST_TTS7:Prepare_Response state including the logical unit number and tag.

If the frame type is COMMAND and the length of the information unit is correct, then this state machine shall send a SCSI Command Received transport protocol service indication to the SCSI application layer.

If the frame type is TASK, then this state machine shall check the logical unit number. If there is no logical unit at the specified logical unit number, then this state machine shall send a Response Data (Invalid Logical Unit Number) message to the ST_TTS7:Prepare_Response state including the logical unit number and tag.

If the frame type is COMMAND or TASK, then this state machine may check the target port transfer tag. If target port transfer tag is invalid, then this state machine may send a Response Data (Invalid Frame) message to the ST_TTS7:Prepare_Response state including the logical unit number and tag.

--`,,,`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If the frame type is TASK and the length of the information unit is correct, then this state may check if the tag conflicts with an existing tag (i.e., an existing command or task management function). If the tag is checked and it conflicts, this state shall send a Response Data (Invalid Frame) message to the ST_TTS7:Prepare_Response state including the logical unit number and tag. If it does not check the tag or the tag does not conflict, this state machine shall send a Task Management Request Received transport protocol service indication to the SCSI application layer. If the frame type is DATA, then this state machine shall send a Data-Out Arrived message to the ST_TTS4:Receive_Data_Out state. Each indication or message shall contain the content of the SAS frame.

If this state machine was started as the result of receiving a HARD_RESET Received confirmation, then this state machine shall send a Transport Reset event notification to the SCSI application layer and terminate.

This state machine shall terminate after sending a message, transport protocol service indication, or event notification.

9.2.6.3.3 ST_TTS (target transport server) state machine

9.2.6.3.3.1 ST_TTS state machine overview

The ST_TTS state machine performs the following functions:

a) processes and sends transport protocol service confirmations to the SCSI application layer;
b) receives and processes transport protocol service requests and responses from the SCSI application layer; and
c) communicates with the port layer via requests and confirmations regarding frame transmission.

This state machine consists of the following states:

a) ST_TTS1:Start (see 9.2.6.3.3.2);
b) ST_TTS2:Send_Frame (see 9.2.6.3.3.3);
c) ST_TTS3:Prepare_Data_In (see 9.2.6.3.3.4);
d) ST_TTS4:Receive_Data_Out (see 9.2.6.3.3.5);
e) ST_TTS5:Prepare_Xfer_Rdy (see 9.2.6.3.3.6);
f) ST_TTS6:Process_Data_Out (see 9.2.6.3.3.7); and
g) ST_TTS7:Prepare_Response (see 9.2.6.3.3.8).

This state machine shall be started in the ST_TTS1:Start state if one of the following is received:

a) a Send Data-In transport protocol service request;
b) a Task Management Function Executed transport protocol service response; or
c) a Send Command Complete transport protocol service response.

This state machine shall be started in the ST_TTS4:Receive_Data_Out state if this state machine is not already running and:

a) a Receive Data-Out transport protocol service request is received; or
b) a Data-Out Arrived message is received and first burst is enabled.

This state machine shall be started in the ST_TTS7:Prepare_Response state when a Response Data message is received and this state machine is not already running.

9.2.6.3.3.2 ST_TTS1:Start state

9.2.6.3.3.2.1 State description

The request or response that caused this state machine to be started includes the following to be used in any OPEN address frames required to service the request or response:

a) connection rate;
b) initiator connection tag; and
c) destination SAS address.

A Send Data-In transport protocol service request also includes the following:

a) logical unit number;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) tag;
c) device server buffer (e.g., starting logical block address); and
d) request byte count (e.g., transfer length).

A Task Management Function Executed transport protocol service response or Send Command Complete transport protocol service response also includes the following:

a) logical unit number;
b) tag;
c) task management function; and
d) tag of task to be managed.

9.2.6.3.3.2.2 Transition ST_TTS1:Start to ST_TTS2:Send_Frame

This transition shall occur after receiving a Send Data-In transport protocol service request and shall include all the information received in the Send Data-In transport protocol service request as arguments.

9.2.6.3.3.2.3 Transition ST_TTS1:Start to ST_TTS7:Prepare_Response

This transition shall occur after receiving a Task Management Function Executed transport protocol service response or a Send Command Complete transport protocol service response and shall include all the information received in the Task Management Function Executed transport protocol service response or Send Command Complete transport protocol service response as arguments.

9.2.6.3.3.3 ST_TTS2:Send_Frame state

9.2.6.3.3.3.1 State description

If this state is entered from the ST_TTS3:Prepare_Data_In state for transmission of a 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_TTS5:Prepare_Xfer_Rdy state for transmission of an XFER_RDY frame and this state has received an ACK Transmitted confirmation for each DATA frame previously received (i.e., received by this state machine with a Data-Out Arrived message), then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_TTS7:Prepare_Response state for transmission of a RESPONSE frame and this state has received an ACK Transmitted confirmation for each DATA frame previously received (i.e., received by this state machine with a Data-Out Arrived message), then this state shall send a Transmit Frame (Interlocked) request to the port layer.

NOTE 26 - The XFER_RDY and RESPONSE frame rules ensure that wide ports do not send an XFER_RDY or RESPONSE frame on a phy until all the ACKs have been transmitted for write DATA frames on a different phy. In a narrow port, the link layer ensures that ACK/NAKs are balanced before transmitting an interlocked frame.

A Transmit Frame request from this state shall include the SSP frame and the following to be used for any OPEN address frame:

a) the 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 set to the SAS address of the SSP target port.

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), this state shall send a Nexus Lost confirmation to the SCSI application layer.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If the confirmation is Transmission Status (Frame Transmitted) confirmation, then this state machine shall expect to receive one of the following confirmations for the frame:

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 frame transmitted was an XFER_RDY frame or a RESPONSE frame, then the state machine shall wait to receive a Transmission Status (ACK Received), Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK) confirmation before transitioning from this state. If the frame transmitted was a DATA frame, then the state machine may transition to ST_TTS3:Prepare_Data_In as described in 9.2.6.3.3.3.2.

This state shall send a Data-In Delivered (Delivery Result = DELIVERY SUCCESSFUL) transport protocol service confirmation to the SCSI application layer if:

a) for a DATA frame, this state receives a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) confirmation for each of the DATA frames transmitted and the number of bytes moved for the Send Data-In transport protocol service request equals the Request Byte Count; or
b) for a RESPONSE frame, this state receives a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) confirmation.

This state shall send a Data-In Delivered (Delivery Result = DELIVERY FAILURE - NAK RECEIVED) transport protocol service confirmation to the SCSI application layer if the received transmission status confirmation message for a DATA or XFER_RDY frame was Transmission Status (Frame Transmitted) followed by a confirmation of Transmission Status (NAK Received).

This state shall send a Data-In Delivered (Delivery Result = DELIVERY FAILURE - ACK/NAK TIMEOUT) transport protocol service confirmation to the SCSI application layer if the received transmission status confirmation message for a DATA or XFER_RDY frame was Transmission Status (Frame Transmitted) followed by a confirmation of Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK).

A Data-In Delivered transport protocol service confirmation to the SCSI application layer shall include the following:

a) any argument received from the port layer (e.g., Transmission Status (Frame Transmitted) or Service Delivery Subsystem Failure); and
b) I_T_L_x nexus information (i.e., destination SAS address and tag).

This state machine shall terminate after sending the Data-In Delivered confirmation.

This state may also send a Cancel request to the port layer to cancel a previous Transmit Frame request. A Cancel request shall include the following arguments:

a) the destination SAS address; and
b) the tag.

This state machine terminates upon receipt of a Transmission Status (Cancel Acknowledge) confirmation.

9.2.6.3.3.3.2 Transition ST_TTS2:Send_Frame to ST_TTS3:Prepare_Data_In

If this state machine was started as the result of receiving a Send Data-In transport protocol service request, the specified values are included with the request, and this state has received an ACK Transmitted confirmation for the COMMAND frame, then this state shall transition to the ST_TTS3:Prepare_Data_In state.

NOTE 27 - The COMMAND frame rule ensures that ports do not send a read DATA frame until the ACK has been transmitted for the COMMAND frame.

If this state receives a Transmission Status (Frame Transmitted) confirmation for a DATA frame and the number of bytes moved for the Send Data-In transport protocol service request is less than the Request Byte Count, then this state shall transition to the ST_TTS3:Prepare_Data_In state.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.2.6.3.3.3.3 Transition ST_TTS2:Send_Frame to ST_TTS4:Receive_Data_Out

This transition shall occur after receiving a Transmission Status (Frame Transmitted) confirmation and a Transmission Status (ACK Received) confirmation for an XFER_RDY frame.

9.2.6.3.3.3.4 Transition ST_TTS2:Send_Frame to ST_TTS7:Prepare_Response

This transition shall occur after receiving a Transmission Status confirmation with an argument other than ACK Received for a RESPONSE frame.

9.2.6.3.3.4 ST_TTS3:Prepare_Data_In state

9.2.6.3.3.4.1 State description

This state fetches the data from the Device Server Buffer and constructs a DATA frame. This state shall use the logical unit number and tag received from the SCSI application layer to construct the frame.

This state shall generate the following values when constructing the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
e) number of fill bytes;
f) fill bytes;
g) data offset; and
h) data.

9.2.6.3.3.4.2 Transition ST_TTS3:Prepare_Data_In to ST_TTS2:Send_Frame

This transition shall occur after constructing a DATA frame.

9.2.6.3.3.5 ST_TTS4:Receive_Data_Out state

9.2.6.3.3.5.1 State description

If a Receive Data-Out transport protocol service request caused this state machine to be started, then the request includes the following to be used in any OPEN address frames required to service the request:

a) connection rate;
b) initiator connection tag;
c) destination SAS address;
d) logical unit number;
e) tag;
f) device server buffer (e.g., starting logical block address); and
g) request byte count (e.g., transfer length).

If a Data-Out Arrived message caused this state machine to be started (i.e., first burst is enabled) and a Receive Data-Out transport protocol service request has not been received, then this state shall wait to process the Data-Out Arrived message until this state receives a Receive Data-Out transport protocol service request. The data received in the Data-Out Arrived message shall be saved in a first burst buffer until this state receives a Receive Data-Out transport protocol service request.

After this state receives a Receive Data-Out transport protocol service request and a Data-Out Arrived message, then this state shall verify the received data frame as follows:

1) check the target port transfer tag value in the DATA frame if the target port transfer tag is being used. If the value is incorrect, then this state shall discard the frame and terminate;
2) check the data offset. If the data offset was not expected, then this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - DATA OFFSET ERROR) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3) check the length of the data. If an XFER_RDY frame was sent for the data (i.e., it is not first burst data) and the length of the data exceeds that specified by the XFER_RDY frame that requested the data, then this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - TOO MUCH WRITE DATA) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation; and
4) check the length of the data. If the length of the data is zero, then this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - INFORMATION UNIT TOO SHORT) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation.

If the Initiator Response Timeout timer is implemented, this state shall initialize and start the Initiator Response Timeout timer after any of the following occur:

a) a Data-Out Arrived message is received;
b) this state is entered from the ST_TTS2:Send_Frame state; or
c) this state is entered from the ST_TTS6:Process_Data_Out state.

If the Initiator Response Timeout timer expires this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation.

If this state is entered from the ST_TTS2:Send_Frame state, then this state shall wait for a Data-Out Arrived message.

If this state is entered from the ST_TTS6:Process_Data_Out state and the number of bytes moved for the Receive Data-Out transport protocol service request is less than the Request Byte Count, then this state shall wait for a Data-Out Arrived message.

If this state is entered from the ST_TTS6:Process_Data_Out state and number of bytes moved for the Receive Data-Out transport protocol service request equals the Request Byte Count, then this state shall send a Data-Out Received (Delivery Result = DELIVERY SUCCESSFUL) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. If this state has no more bytes in its first burst buffer, then this state machine shall terminate after sending the confirmation. If this state has more bytes to move in its first burst buffer, then this state machine shall wait for a Receive Data-Out transport protocol service request.

9.2.6.3.3.5.2 Transition ST_TTS4:Receive_Data_Out to ST_TTS5:Prepare_Xfer_Rdy

This transition shall occur after:

a) this state has received a Receive Data-Out transport protocol service request and first burst is not enabled; or
b) this state has received a Receive Data-Out transport protocol service request, first burst is enabled, all the first burst data has been processed, and the first burst data did not satisfy the requested byte count.

9.2.6.3.3.5.3 Transition ST_TTS4:Receive_Data_Out to ST_TTS6:Process_Data_Out

This transition shall occur after:

a) a Receive Data-Out transport protocol service request is received; and
b) a Data-Out Arrived message is received and verified (see 9.2.6.3.3.5.1).

9.2.6.3.3.6 ST_TTS5:Prepare_Xfer_Rdy state

9.2.6.3.3.6.1 State description

This state shall construct an XFER_RDY frame. This state shall use the following values received from the SCSI application layer to construct the frame:

a) logical unit number;
b) tag;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

c) target port transfer tag;
d) requested offset; and
e) write data length.

If first burst is enabled, this state shall adjust the write data length to reflect the amount of first burst data.

This state shall generate the following values when constructing the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero; and
e) number of fill bytes.

9.2.6.3.3.6.2 Transition ST_TTS5:Prepare_Xfer_Rdy to ST_TTS2:Send_Frame

This transition shall occur after constructing an XFER_RDY frame.

9.2.6.3.3.7 ST_TTS6:Process_Data_Out state

9.2.6.3.3.7.1 State description

This state shall process the data received in the Data-Out Arrived message using the Device Server Buffer (e.g., logical block address) to which the data is to be transferred.

9.2.6.3.3.7.2 Transition ST_TTS6:Process_Data_Out to ST_TTS4:Receive_Data_Out

This transition shall occur after data received in a Data-Out message has been processed.

9.2.6.3.3.8 ST_TTS7:Prepare_Response state

9.2.6.3.3.8.1 State description

This state shall construct a RESPONSE frame if this state was entered as the result of this state machine:

a) receiving a Response Data message;
b) receiving a Task Management Function Executed transport protocol service response; or
c) receiving a Send Command Complete transport protocol service response.

If this state was entered as the result of receiving a Response Data message, this state shall use the logical unit number and tag received in the message and shall construct the frame as described in table 105.

Table 105 — Response Data argument to RESPONSE frame content mapping

Response Data argument	RESPONSE frame
Information Unit Too Short	The DATAPRES field shall be set to SENSE_DATA, the STATUS field shall be set to CHECK CONDITION and the additional sense code shall be set to INFORMATION UNIT TOO SHORT.
Information Unit Too Long	The DATAPRES field shall be set to SENSE_DATA, the STATUS field shall be set to CHECK CONDITION and the additional sense code shall be set to INFORMATION UNIT TOO LONG.
Invalid Frame	The DATAPRES field shall be set to RESPONSE_DATA and the RESPONSE CODE field shall be set to INVALID FRAME.
Invalid Logical Unit Number	The DATAPRES field shall be set to RESPONSE_DATA and the RESPONSE CODE field shall be set to INVALID LOGICAL UNIT NUMBER.

If this state was entered as a result of receiving a Task Management Function Executed transport protocol service response or a Send Command Complete transport protocol service response, this state shall use the following values received from the SCSI application layer to construct the frame:

a) logical unit number;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

b) tag;
c) status;
d) response data; and
e) sense data.

This state shall generate the following values when constructing the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
e) number of fill bytes;
f) fill bytes;
g) data present;
h) sense data length; and
i) response data length.

If this state was entered as the result of the ST_TTS2:Send_Frame state receiving something other than a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) confirmation for a RESPONSE frame (i.e., the frame transmission was unsuccessful), then this state shall check to see if the vendor-specific number of retries for the RESPONSE frame has been exceeded.

If the vendor-specific number of retries has not been exceeded, then this state generates a RESPONSE frame using all of the values for the previous RESPONSE frame except that the retransmit bit shall be set to one.

9.2.6.3.3.8.2 Transition ST_TTS7:Prepare_Response to ST_TTS2:Send_Frame

This transition shall occur after constructing a RESPONSE frame or if the vendor-specific number of retries for transmission of a RESPONSE frame has been exceeded.

9.3 STP transport layer

9.3.1 Initial FIS

A SATA device phy transmits a Register - Device to Host FIS after completing the link reset sequence. The expander device shall update a set of shadow registers with these contents and shall not deliver them to any STP initiator port. The STP initiator ports may read the shadow register contents using the SMP REPORT PHY SATA function (see 10.4.3.7).

9.3.2 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).

9.3.3 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.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.4 SMP transport layer

9.4.1 SMP transport layer overview

Table 106 defines the SMP frame format.

Table 106 — SMP frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE							
1	Frame-type dependent bytes							
	Fill bytes, if needed							
n - 3	(MSB)	CRC						
n								(LSB)

Table 107 defines the SMP FRAME TYPE field, which defines the format of the frame-type dependent bytes.

Table 107 — SMP FRAME TYPE **field**

Code	Name	Frame type	Originator	Reference
40h	SMP_REQUEST	SMP function request	SMP initiator port	9.4.2
41h	SMP_RESPONSE	SMP function response	SMP target port	9.4.3
All others	Reserved.			

The SMP target port in an expander device shall support the SMP_REQUEST and SMP_RESPONSE frames. Other SMP target ports may support these frames.

Fill bytes shall be included after the frame-type dependent bytes so the CRC field is aligned on a four byte boundary. The contents of the fill bytes are vendor specific.

The CRC field contains a CRC value (see 7.5) 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 7.18).

--`,,,`,,,-`-`,,`,,`,`,,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.4.2 SMP_REQUEST frame

The SMP_REQUEST frame is sent by an SMP initiator port to request an SMP function be performed by a management device server. Table 108 defines the SMP_REQUEST frame format.

Table 108 — SMP_REQUEST frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	REQUEST BYTES							
	Fill bytes, if needed							
n - 3	(MSB) CRC							
n								(LSB)

The SMP FRAME TYPE field shall be set to 40h indicating this is an SMP_REQUEST frame.

The REQUEST BYTES field definition and length is based on the SMP function (see 10.4.3.1). The maximum size of the REQUEST BYTES field is 1 024 bytes, making the maximum size of the frame 1 032 bytes (1 024 bytes of data + 4 bytes of header + 4 bytes of CRC).

Fill bytes shall be included after the ADDITIONAL REQUEST BYTES field so the CRC field is aligned on a four byte boundary. The contents of the fill bytes are vendor specific.

The CRC field is defined in 9.4.1.

9.4.3 SMP_RESPONSE frame

The SMP_RESPONSE frame is sent by an SMP target port in response to an SMP_REQUEST frame. Table 109 defines the SMP_RESPONSE frame format.

Table 109 — SMP_RESPONSE frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	RESPONSE BYTES							
	Fill bytes, if needed							
n - 3	(MSB) CRC							
n								(LSB)

The SMP FRAME TYPE field shall be set to 41h indicating this is an SMP_RESPONSE frame.

The RESPONSE BYTES field definition and length is based on the SMP function (see 10.4.3.2). The maximum size of the RESPONSE BYTES field is 1 024 bytes, making the maximum size of the frame 1 032 bytes (1 024 bytes of data + 4 bytes of header + 4 bytes of CRC).

Fill bytes shall be included after the ADDITIONAL RESPONSE BYTES field so the CRC field is aligned on a four byte boundary. The contents of the fill bytes are vendor specific.

The CRC field is defined in 9.4.1.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.4.4 Sequence of SMP frames

Inside an SMP connection, the source phy transmits a single SMP_REQUEST frame and the destination phy replies with a single SMP_RESPONSE frame.

Figure 115 shows the sequence of SMP frames.

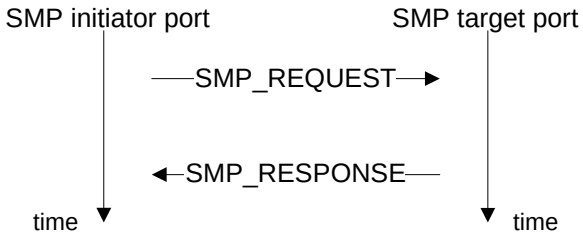


Figure 115 — Sequence of SMP frames

9.4.5 MT (transport layer for SMP ports) state machines

9.4.5.1 SMP transport layer state machines overview

The SMP transport layer contains state machines that process requests from the management application layer and returns 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 9.4.5.2); and
b) MT_TP (transport layer for SMP target ports) state machine (see 9.4.5.3).

9.4.5.2 MT_IP (transport layer for SMP initiator ports) state machine

9.4.5.2.1 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 9.4.5.2.2)(initial state);
b) MT_IP2:Send (see 9.4.5.2.3); and
c) MT_IP3:Receive (see 9.4.5.2.4).

This state machine shall maintain the timers listed in table 110.

Table 110 — MT_IP timers

Timer	Initial value
SMP Frame Receive Timeout timer	Vendor specific

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Figure 116 describes the MT_IP state machine.

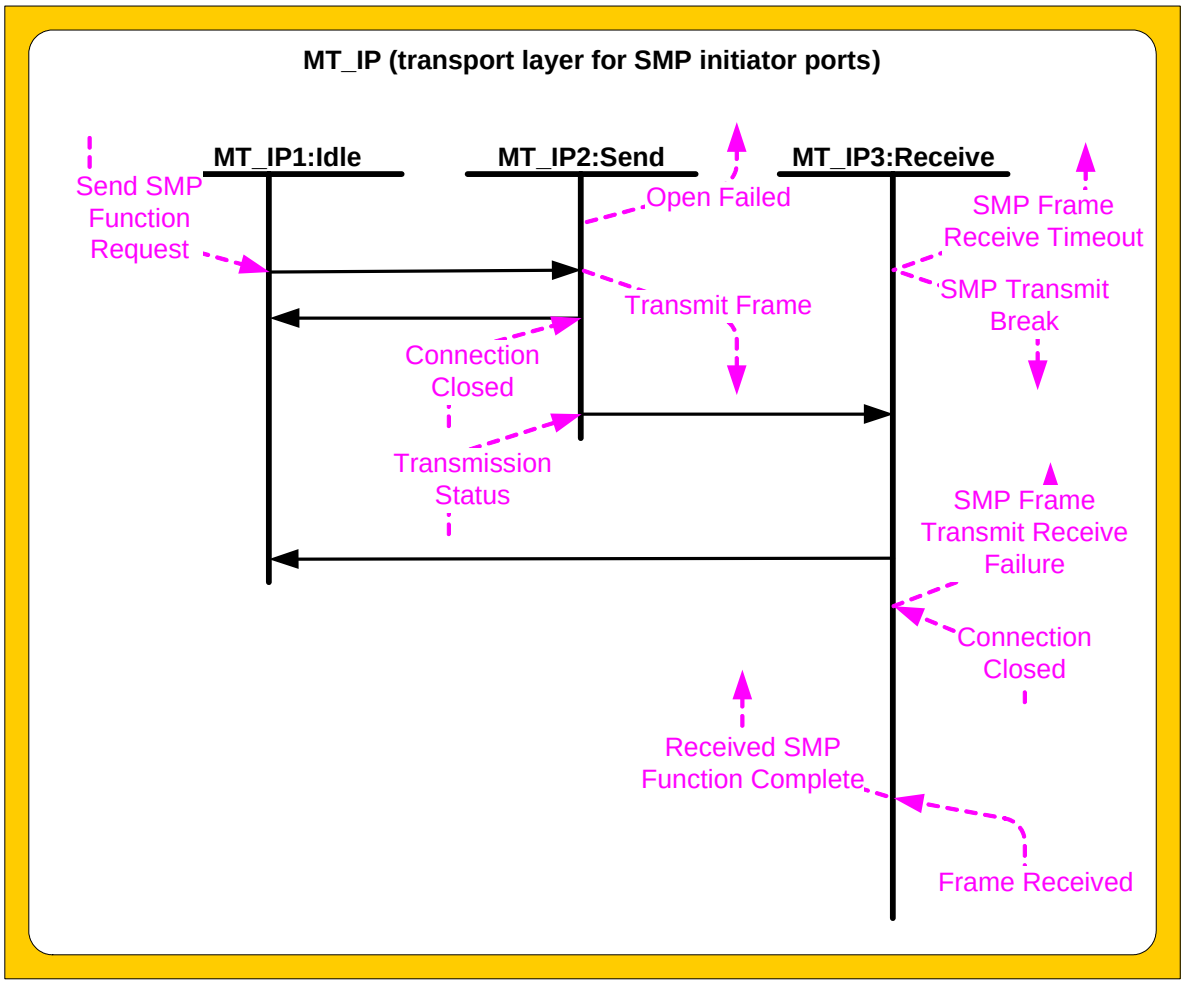


Figure 116 — MT_IP (transport layer for SMP initiator ports) state machine

9.4.5.2.2 MT_IP1:Idle state

9.4.5.2.2.1 State description

This state waits for a Send SMP Function Request request, which includes the following arguments:

a) connection rate;
b) destination SAS address; and
c) request bytes.

9.4.5.2.2.2 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:

a) connection rate;
b) destination SAS address; and
c) request bytes.

9.4.5.2.3 MT_IP2:Send state

9.4.5.2.3.1 State description

This state constructs an SMP_REQUEST frame using the following arguments received with the transition into this state: request bytes;

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

and sends a Transmit Frame request to the port layer. The Transmit Frame request shall contain the following arguments:

a) protocol set to SMP;
b) connection rate;
c) destination SAS address;
d) request bytes;
e) source SAS address set to the SAS address of the SMP initiator port;
f) initiator port bit set to one; and
g) initiator connection tag set to FFFFh.

9.4.5.2.3.2 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.

9.4.5.2.3.3 Transition MT_IP2:Send to MT_IP3:Receive

This transition shall occur after receiving a Transmission Status (Frame Transmitted) confirmation.

9.4.5.2.4 MT_IP3:Receive state

9.4.5.2.4.1 State description

This state waits for a confirmation from the port layer that either an SMP frame has been received or a failure occurred.

Upon entry into this state, this state shall initialize and start the SMP Frame Receive Timeout timer.

If a Frame Received confirmation is received and the SMP frame type is equal to 41h, this state shall send a Received SMP Function Complete confirmation to the management application layer. If the SMP frame type is not equal to 41h, this state shall send an SMP Transmit Break request to the port layer.

If a Connection Closed or Frame Received (SMP Failure) confirmation is received, this state shall send an SMP Frame Transmit Receive Failure confirmation to the management application layer.

If the SMP Frame Receive Timeout timer expires before a Received SMP Function Complete confirmation is received, this state shall send an SMP Frame Receive Timeout confirmation to the management application layer and send an SMP Transmit Break request to the port layer.

9.4.5.2.4.2 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;
b) sending an SMP Frame Transmit Receive Failure confirmation; or
c) sending an SMP Transmit Break request.

9.4.5.3 MT_TP (transport layer for SMP target ports) state machine

9.4.5.3.1 MT_TP state machine overview

The MT_TP state machine informs the management application layer of the receipt of an SMP frame. Confirmation of the receipt of an SMP frame is sent to the management application layer. The management application layer creates the corresponding SMP_RESPONSE frame and this state sends it to the port layer.

This state machine consists of the following states:

a) MT_TP1:Idle (see 9.4.5.3.2)(initial state); and
b) MT_TP2:Respond (see 9.4.5.3.3).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 117 describes the MT_TP state machine.

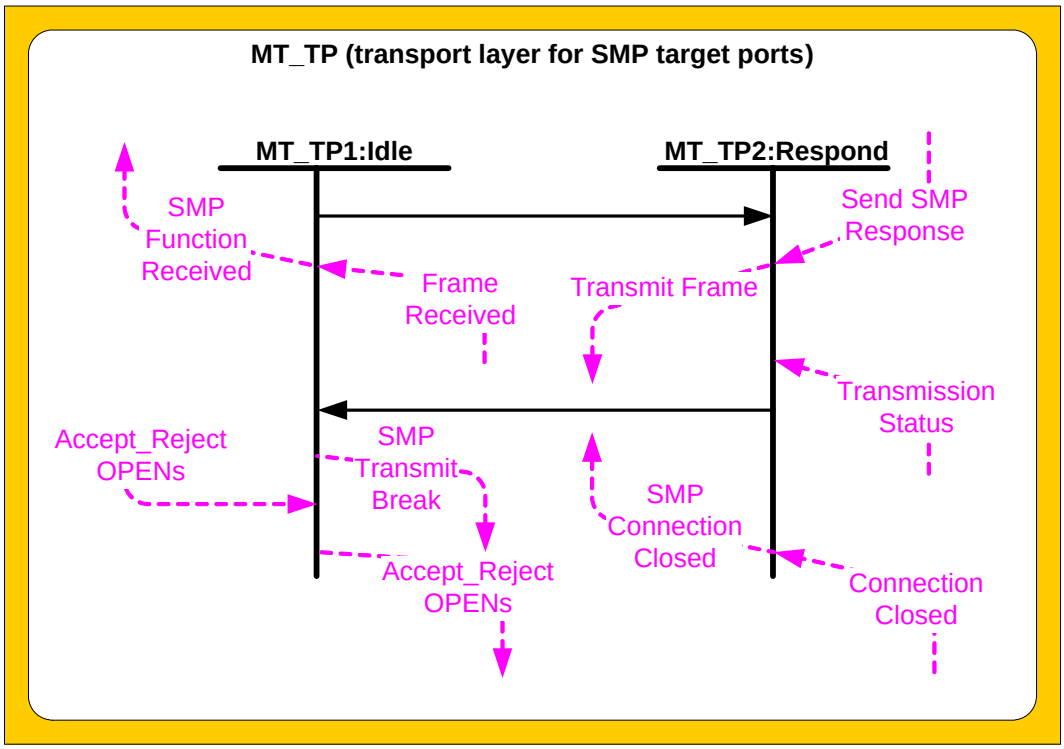


Figure 117 — MT_TP (transport layer for SMP target ports) state machine

9.4.5.3.2 MT_TP1:Idle state

9.4.5.3.2.1 State description

This state waits for a Frame Received confirmation. If the SMP frame type is not equal to 40h, this state shall discard the frame and send a 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) or Accept_Reject OPENs (Reject SMP) request is received, this state shall send an Accept_Reject OPENs request with the same arguments to the port layer.

9.4.5.3.2.2 Transition MT_TP1:Idle to MT_TP2:Respond

This transition shall occur after sending an SMP Function Received confirmation.

9.4.5.3.3 MT_TP2:Respond state

9.4.5.3.3.1 State description

This state waits for a Send SMP Response request, which includes the following arguments:

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.

If this state receives a Connection Closed confirmation, this state shall send an SMP Connection Closed confirmation to the management application layer.

9.4.5.3.3.2 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

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

b) sending an SMP Connection Closed confirmation.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

10 Application layer

10.1 Application layer overview

The application layer defines SCSI, ATA, and management specific features.

10.2 SCSI application layer

10.2.1 SCSI transport protocol services

10.2.1.1 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 remote procedure call (see SAM-3):

Service response = Execute Command (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Autosense Request]), OUT ([Data-In Buffer], [Sense Data], Status))

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 remote procedure calls (see SAM-3):

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 = LOGICAL UNIT RESET (IN (Nexus)); and
f) Service Response = QUERY TASK (IN (Nexus)).

SSP defines the transport protocol services required by SAM-3 in support of the these remote procedure calls.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 111 describes the mapping of the remote procedure calls to transport protocol services and the SSP implementation of each transport protocol service.

Table 111 — SCSI architecture mapping

Remote procedure call	Type of transport protocol service	Transport protocol service interaction	Transport protocol service	I/T [a]	SSP implementation
Execute Command	Request/ Confirmation	Request	Send SCSI Command	I	COMMAND frame
		Indication	SCSI Command Received	T	Receipt of the COMMAND frame
		Response	Send Command Complete	T	RESPONSE frame
		Confirmation	Command Complete Received	I	Receipt of the RESPONSE frame or problem transmitting the COMMAND frame
	Data Transfer [b]	Request	Send Data-In	T	DATA frames
		Confirmation	Data-In Delivered	T	Receipt of ACKs for the DATA frames
		Request	Receive Data-Out	T	XFER_RDY frame
		Confirmation	Data-Out Received	T	Receipt of DATA frames
ABORT TASK, ABORT TASK SET, CLEAR ACA, CLEAR TASK SET, LOGICAL UNIT RESET, and QUERY TASK	Request/ Confirmation	Request	Send Task Management Request	I	TASK frame
		Indication	Task Management Request Received	T	Receipt of the TASK frame
		Response	Task Management Function Executed	T	RESPONSE frame
		Confirmation	Received Task Management Function-Executed	I	Receipt of the RESPONSE frame or problem transmitting the TASK frame

[a] I/T indicates whether the SSP initiator port (I) or the SSP target port (T) implements the transport protocol service.
[b] SCSI initiator port Data Transfer transport protocol services are not specified by SAM-3.

These protocol services are used as the requests and confirmations to the SSP transport layer state machines (see 9.2.6) from the SCSI application layer.

10.2.1.2 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_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Autosense Request], [Command Reference Number], [First Burst Enabled]))

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 112 shows how the arguments to the Send SCSI Command transport protocol service are used.

Table 112 — Send SCSI Command transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T used to select a connection; b) L used to set the LOGICAL UNIT NUMBER field in the COMMAND frame header; and c) Q used to set the TAG field in the COMMAND frame header.
CDB	Used to set the CDB field in the COMMAND frame.
[Task Attribute]	Used to set the TASK ATTRIBUTE field in the COMMAND frame.
[Data-In Buffer Size]	Maximum of 2^{32}
[Data-Out Buffer]	Internal to the SSP initiator port.
[Data-Out Buffer Size]	Maximum of 2^{32}
[Autosense Request]	True
[First Burst Enabled]	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 10.2.6.1.5) for the SCSI target port without waiting for an XFER_RDY frame.

10.2.1.3 SCSI Command Received transport protocol service

An SSP target port uses the SCSI Command Received transport protocol service indication to notify a device server that it has received a COMMAND frame.

SCSI Command Received (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Autosense Request], [Command Reference Number]))

Table 113 shows how the arguments to the SCSI Command Received transport protocol service are determined.

Table 113 — SCSI Command Received transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T indicated by the connection; b) L indicated by the LOGICAL UNIT NUMBER field in the COMMAND frame header; and c) Q indicated by the TAG field in the COMMAND frame header.
CDB	From the CDB field in the COMMAND frame.
[Task Attribute]	From the TASK ATTRIBUTE field in the COMMAND frame.
[Autosense Request]	True
[Command Reference Number]	Ignored
[First Burst Enabled]	Indicates that first burst data is being delivered based on the FIRST BURST SIZE field in the Disconnect-Reconnect mode page (see 10.2.6.1.5).

If an SSP target port calls SCSI Command Received () with a TAG already in use by another SCSI command (i.e., an overlapped command), the device server responses are defined in SAM-3.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.1.4 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_x Nexus, [Sense Data], Status, Service Response))

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 have been responded to with Data-Out Received () and all its outstanding Send Data-In () calls have been responded to with Data-In Delivered ().

Table 114 shows how the arguments to the Send Command Complete transport protocol service are used.

Table 114 — Send Command Complete transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T used to select a connection; b) L used to set the LOGICAL UNIT NUMBER field in the RESPONSE frame header; and c) Q used to set the TAG field in the RESPONSE frame header.
[Sense Data]	Used to set the RESPONSE frame SENSE DATA field.
Status	Used to set the RESPONSE frame STATUS field.
Service Response	Used to set the DATAPRES field and STATUS field in the RESPONSE frame: a) TASK COMPLETE: The DATAPRES field is set to NO_DATA or SENSE_DATA and the STATUS field is set to a value other than INTERMEDIATE or INTERMEDIATE-CONDITION MET; b) LINKED COMMAND COMPLETE: The DATAPRES field is set to NO_DATA or SENSE_DATA and the STATUS field is set to INTERMEDIATE or INTERMEDIATE-CONDITION MET; or c) SERVICE DELIVERY OR TARGET FAILURE: The DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to INVALID FRAME.

10.2.1.5 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).

Command Complete Received (IN (I_T_L_x Nexus, [Data-In Buffer], [Sense Data], Status, Service Response))

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 115 shows how the arguments to the Command Complete Received transport protocol service are determined.

Table 115 — Command Complete Received transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T indicated by the connection; b) L indicated by the LOGICAL UNIT NUMBER field in the RESPONSE frame header or COMMAND frame header; and c) Q indicated by the TAG field in the RESPONSE frame header or COMMAND frame header.
[Data-In Buffer]	Internal to the SSP initiator port.
[Sense Data]	From the RESPONSE frame SENSE DATA field.
Status	From the RESPONSE frame STATUS field.
Service Response	From the RESPONSE frame DATAPRES field and STATUS field, or from a NAK on the COMMAND frame: a) TASK COMPLETE: The RESPONSE frame contains a DATAPRES field set to NO_DATA or SENSE_DATA and a STATUS field set to a value other than INTERMEDIATE or INTERMEDIATE-CONDITION MET; b) LINKED COMMAND COMPLETE: The RESPONSE frame contains a DATAPRES field set to NO_DATA or SENSE_DATA and a STATUS field set to INTERMEDIATE or INTERMEDIATE-CONDITION MET; or c) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to INVALID FRAME, or the COMMAND frame was NAKed.

10.2.1.6 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 DATA frame.

Send Data-In (IN (I_T_L_x Nexus, Device Server Buffer, Application Client Buffer Offset, Request Byte Count))

A device server shall only call Send Data-In () during a read or bidirectional command.

A device server shall not call Send Data-In () for a given I_T_L_Q nexus after it has called Send Command Complete () for that I_T_L_Q nexus (e.g., a RESPONSE frame with for that I_T_L_Q nexus) or called Task Management Function Executed for a task management function that terminates that task (e.g., an ABORT TASK).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 116 shows how the arguments to the Send Data-In transport protocol service are used.

Table 116 — Send Data-In transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T used to select a connection; b) L used to set the LOGICAL UNIT NUMBER field in the DATA frame header; and c) Q used to set the TAG field in the DATA frame header.
Device Server Buffer	Internal to the device server.
Application Client Buffer Offset	Used to set the DATA frame DATA OFFSET field.
Request Byte Count	Used to select the size of the DATA frame.

10.2.1.7 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 DATA frame.

Data-In Delivered (IN (I_T_L_x Nexus))

Table 117 shows how the arguments to the Data-In Delivered transport protocol service are determined.

Table 117 — Data-In Delivered transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T indicated by the connection; b) L indicated by the LOGICAL UNIT NUMBER field in the DATA frame header; and c) Q indicated by the TAG field in the DATA frame header.
Delivery Result	From the response to the outgoing DATA frame: a) DELIVERY SUCCESSFUL: The DATA frame received an ACK; or b) DELIVERY FAILURE: The DATA frame received a NAK or no response.

10.2.1.8 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_x Nexus, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))

A device server shall only call Receive Data-Out () during a write or bidirectional command.

A device server shall not call Receive Data-Out () for a given I_T_L_Q nexus until Data-Out Received () has completed successfully for the previous Receive Data-Out () call (i.e., no XFER_RDY frame until all write DATA frames for the previous XFER_RDY frame, if any, and has provided link layer 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 task (e.g., an ABORT TASK).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 118 shows how the arguments to the Receive Data-Out transport protocol service are used.

Table 118 — Receive Data-Out transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T used to select a connection; b) L used to set the LOGICAL UNIT NUMBER field in the XFER_RDY frame header; and c) Q used to set the TAG field in the XFER_RDY frame header.
Application Client Buffer Offset	Used to set the XFER_RDY frame REQUESTED OFFSET field.
Request Byte Count	Used to set the XFER_RDY frame WRITE DATA LENGTH field.
Device Server Buffer	Internal to the device server.

10.2.1.9 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 DATA frames in response).

Data-Out Received (IN (I_T_L_x Nexus))

Table 119 shows how the arguments to the Data-Out Received transport protocol service are determined.

Table 119 — Data-Out Received transport protocol service arguments

Argument	SAS SSP implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T indicated by the connection; b) L indicated by the LOGICAL UNIT NUMBER field in the XFER_RDY frame header; and c) Q indicated by the 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 successfully transmitted and all the DATA frames for the requested write data were received; or b) DELIVERY FAILURE: The XFER_RDY frame received a NAK or no response.

10.2.1.10 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, [Association]))

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 120 shows how the arguments to the Send Task Management Request transport protocol service are used.

Table 120 — Send Task Management Request transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T_L nexus or I_T_L_Q nexus (depending on the Function Identifier), where: a) I_T used to select a connection; b) L used to set the LOGICAL UNIT NUMBER field in the TASK frame header; and c) Q (for an I_T_L_Q nexus) used to set the TAG OF TASK TO BE MANAGED field in the TASK frame header.
Function Identifier	Used to set 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) LOGICAL UNIT RESET (Nexus argument specifies an I_T_L Nexus); and f) QUERY TASK (Nexus argument specifies an I_T_L_Q Nexus).
[Association]	Used to set the TAG field in the TASK frame header.

10.2.1.11 Task Management Request Received transport protocol service

An SSP target port uses the Task Management Request Received transport protocol service indication to notify a device server that it has received a TASK frame.

Task Management Request Received (IN (Nexus, Function Identifier, [Association]))

Table 121 shows how the arguments to the Task Management Request Received transport protocol service are determined.

Table 121 — Task Management Request Received transport protocol service arguments

Argument	SAS SSP implementation
Nexus	I_T_L nexus or I_T_L_Q nexus (depending on the Function Identifier), where: a) I_T indicated by the connection; b) L indicated by the LOGICAL UNIT NUMBER field in the TASK frame header; and c) Q (for an I_T_L_Q nexus) indicated by the TAG OF TASK TO BE MANAGED field in the TASK frame header.
Function Identifier	From 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) LOGICAL UNIT RESET (Nexus argument specifies an I_T_L Nexus); and f) QUERY TASK (Nexus argument specifies an I_T_L_Q Nexus).
[Association]	Indicated by the TAG field in the TASK frame header.

10.2.1.12 Task Management Function Executed transport protocol service

A device server 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, [Association]))

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

A device server shall only call Task Management Function Executed () after receiving Task Management Request Received ().

Table 122 shows how the arguments to the Task Management Function Executed transport protocol service are used.

Table 122 — 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_T used to select a connection; b) L used to set the LOGICAL UNIT NUMBER field in the RESPONSE frame header; and c) Q (for an I_T_L_Q nexus) indirectly used to set the TAG field in the RESPONSE frame header.
Service Response	Used to set the DATAPRES field and RESPONSE CODE field in the RESPONSE frame: a) FUNCTION REJECTED: The DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED or INVALID LOGICAL UNIT. b) FUNCTION COMPLETE: The RESPONSE frame DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION COMPLETE; c) FUNCTION SUCCEEDED: The RESPONSE frame DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION SUCCEEDED; or d) SERVICE DELIVERY OR SUBSYSTEM FAILURE: The RESPONSE frame DATAPRES field is set to RESPONSE_DATA and the RESPONSE CODE field is set to INVALID FRAME or TASK MANAGEMENT FUNCTION FAILED.
[Association]	Used to set the TAG field in the RESPONSE frame header.

10.2.1.13 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, [Association]))

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 123 shows how the arguments to the Received Task Management Function-Executed transport protocol service are determined.

Table 123 — Received 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_T indicated by the connection; b) L indicated by the LOGICAL UNIT NUMBER field in the RESPONSE frame header or TASK frame header; and c) Q (for an I_T_L_Q nexus) indirectly indicated by the TAG field in the RESPONSE frame header, or indicated by the TAG OF TASK TO BE MANAGED field TASK frame header.
Service Response	From the response to the TASK frame: a) 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 or INVALID LOGICAL UNIT. b) FUNCTION COMPLETE: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION COMPLETE; c) FUNCTION SUCCEEDED: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to TASK MANAGEMENT FUNCTION SUCCEEDED; or d) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE frame contains a DATAPRES field set to RESPONSE_DATA and a RESPONSE CODE field set to INVALID FRAME or TASK MANAGEMENT FUNCTION FAILED, or the TASK frame was NAKed.
[Association]	Indicated by the TAG field in the RESPONSE frame header or the TASK frame header.

10.2.2 Application client error handling

If an SSP initiator port calls Command Complete Received () and delivers a Service Response of:

a) Service Delivery of Target Failure - XFER_RDY Information Unit Too Short;
b) Service Delivery of Target Failure - XFER_RDY Information Unit Too Long;
c) Service Delivery of Target Failure - XFER_RDY Incorrect Write Data Length;
d) Service Delivery of Target Failure - XFER_RDY Requested Offset Error;
e) Service Delivery of Target Failure - XFER_RDY Not Expected;
f) Service Delivery of Target Failure - DATA Incorrect Data Length;
g) Service Delivery of Target Failure - DATA Too Much Read Data;
h) Service Delivery of Target Failure - DATA Data Offset Error;
i) Service Delivery of Target Failure - NAK Received; or
j) Service Delivery of Target Failure - Connection Failed,

it 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 of Target Failure - ACK/NAK Timeout, the application client shall send a QUERY TASK task management function with Send Task Management Request () to determine whether the command was received successfully. If Received Task Management Function Executed () returns a Service Response of FUNCTION SUCCEEDED, the application client shall assume the command was delivered successfully. 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), the application client shall assume the command was not delivered successfully and may reuse the tag.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

If an application client calls Send Task Management Request () and an SSP initiator port calls Received Task Management Function Executed () and delivers a Service Response of Service Delivery of Target Failure - ACK/NAK Timeout, the application client shall call Send Task Management Request () again with identical arguments.

If an application client calls Send SCSI Command () and an SSP initiator port calls Command Complete Received () a second time for the command, and the second call includes a Retransmit argument set to one, the application client shall ignore the second call.

If an application client calls Send Task Management Request () and an SSP initiator port calls Received Task Management Function Executed () a second time for the task management function, and the second call includes a Retransmit argument set to one, the application client shall ignore the second call.

After a Command Complete Received () or Received Task Management Function Executed () call, an application client shall not reuse the tag until it determines the 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 a 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 tag is no longer active in the logical unit.

10.2.3 Device server error handling

If a device server calls Receive Data-Out () and receives a Delivery Result set to a value in table 124, it shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code as indicated by table 124.

Table 124 — 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 - NAK RECEIVED	NAK RECEIVED
DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT	INITIATOR RESPONSE TIMEOUT

10.2.4 SCSI transport protocol event notifications

Table 125 lists the SCSI transport protocol event notifications supported by this standard.

Table 125 — SCSI transport protocol events

Event notification	SAS SSP implementation
Transport Reset	Receipt of a hard reset sequence (see 4.4.2)
Nexus Lost	Receipt of specific OPEN_REJECTs for a specific time period (see 4.5)

10.2.5 SCSI commands

10.2.5.1 INQUIRY command

The vital product data returned by the INQUIRY command (see SPC-3) that shall be returned for a SAS device is described in 10.2.9.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.5.2 LOG SELECT and LOG SENSE commands

SAS-specific log pages accessed with the LOG SELECT and LOG SENSE commands (see SPC-3) are described in 10.2.7.

10.2.5.3 MODE SELECT and MODE SENSE commands

SAS-specific mode pages accessed with the MODE SELECT and MODE SENSE commands (see SPC-3) are described in 10.2.6.

10.2.5.4 START STOP UNIT command

The power condition states controlled by the START STOP UNIT command (see SBC-2) for a SAS device are described in 10.2.8.

10.2.6 SCSI mode parameters

10.2.6.1 Disconnect-Reconnect mode page

10.2.6.1.1 Disconnect-Reconnect mode page overview

The Disconnect-Reconnect mode page (see SPC-3) provides the application client the means to tune the performance of the service delivery subsystem. Table 126 defines the parameters which are applicable to SSP. If any field in the Disconnect-Reconnect mode page is not implemented, the value assumed for the functionality of the field shall be zero (i.e., as if the field in the mode page is implemented and the field is set to zero).

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 below in this subclause. If any other fields within the Disconnect-Reconnect mode page of the MODE SELECT command contain a non-zero value, the device

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

server shall return CHECK CONDITION status for that MODE SELECT command. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ILLEGAL FIELD IN PARAMETER LIST.

Table 126 — Disconnect-Reconnect mode page for SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	Reserved	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	Reserved							
14	(MSB)	FIRST BURST SIZE						
15								(LSB)

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-3.

The PAGE CODE (PS) field shall be set to 02h.

The PAGE LENGTH field shall be set to 0Eh.

The BUS INACTIVITY TIME LIMIT field is defined in 10.2.6.1.2.

The MAXIMUM CONNECT TIME LIMIT field is defined in 10.2.6.1.3.

The MAXIMUM BURST SIZE field is defined in 10.2.6.1.4.

The FIRST BURST SIZE field is defined in 10.2.6.1.5.

10.2.6.1.2 BUS INACTIVITY TIME LIMIT field

The value in the BUS INACTIVITY TIME LIMIT field contains the maximum period that an SSP target port is permitted to maintain a connection (see 4.1.10) without transferring a frame to the SSP initiator port. This value shall be the number of 100 μs increments between frames that the SSP target port transmits during a connection. When this number 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-3. A value of zero in this field shall specify that there is no bus inactivity time limit. The bus inactivity time limit is enforced by the port layer (see 8.2.3).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.6.1.3 MAXIMUM CONNECT TIME LIMIT field

The value in the MAXIMUM CONNECT TIME LIMIT field contains the maximum duration of a connection (see 4.1.10). This value shall be the number of 100 μs increments that an SSP target port transmits during a connection after which the SSP target port shall prepare to close the connection (e.g., a value of one in this field means that the time is less than or equal to 100 μs and a value of two in this field means that the time is less than or equal to 200 μs). If an SSP target port is transferring a frame when the maximum connection time limit is exceeded, the SSP target port shall complete transfer of the frame before preparing to close the connection. A value of zero in this field shall specify that there is no maximum connection time limit. The maximum connection time limit is enforced by the port layer (see 8.2.3).

10.2.6.1.4 MAXIMUM BURST SIZE field

For read data, the value in the MAXIMUM BURST SIZE field contains the maximum amount of data 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;

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 9.2.2.3).

This value shall be specified in 512-byte increments (e.g., a value of one 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 two 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). The device server may round this value down as defined in SPC-3. A value of zero in this field shall specify that there is no maximum burst size.

In terms of the SCSI transport protocol services (see 10.2.1), 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.

10.2.6.1.5 FIRST BURST SIZE field

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 9.2.2.3) from the SSP target port (e.g., a value of one 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 two 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 the 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, 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, 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 initiator. All data for the command is not required to be transferred during the same connection in which the command is transferred.

A value of zero in this field shall specify that there is no first burst size (i.e., an SSP initiator port shall transmit no 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 10.2.1) and the SSP transport layer (see 9.2.6).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.6.2 Protocol-Specific Port mode page

10.2.6.2.1 Protocol-Specific Port mode page overview

The Protocol-Specific Port mode page (see SPC-3) contains parameters that affect SSP target port operation. If the mode page is implemented, all logical units in SCSI target devices in SAS domains supporting the MODE SELECT or MODE SENSE commands shall implement the page, and there shall be one copy of the mode page shared by all SSP initiator ports.

If a SAS target device has multiple SSP target ports, changes in the short page parameters for one SSP target port should not affect other SSP target ports.

Table 128 defines the subpages of this mode page.

Table 127 — Protocol-Specific Port Control mode page subpages

Subpage	Description	Reference
Short page	Short format	10.2.6.2.2
Long page 00h	Not allowed	
Long page 01h	Phy Control And Discover subpage	10.2.6.2.3
Long page E0h - FEh	Vendor specific	
Long page FFh	Return all subpages for the Port Control mode page	SPC-3
All others	Reserved	

10.2.6.2.2 Protocol-Specific Port mode page - short format

Parameters in this page shall affect all phys in the SSP target port, and may affect all SSP target ports in the SAS target device.

Table 128 defines the format of the page for SAS SSP.

Table 128 — Protocol-Specific Port Control mode page for SAS SSP - short format

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (06h)							
2	Reserved				PROTOCOL IDENTIFIER (6h)			
3	Reserved							
4	(MSB)	I_T NEXUS LOSS TIME						
5								(LSB)
6	(MSB)	INITIATOR RESPONSE TIMEOUT						
7								(LSB)

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-3.

The SPF field shall be set to zero for access to the short format mode page.

The PAGE CODE field shall be set to 19h.

The PAGE LENGTH field shall be set to 06h.

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The I_T NEXUS LOSS TIME field contains the time that the SSP target port shall retry connection requests to an SSP initiator port that are rejected with responses indicating the SSP initiator port may no longer be present (see 8.2.2) before recognizing an I_T nexus loss (see 4.5). Table 129 defines the values of the I_T NEXUS LOSS TIME field. If this mode page is not implemented, the I_T nexus loss time is vendor specific. This value is enforced by the port layer (see 8.2.2).

Table 129 — I_T nexus loss time

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 28 - If this mode page is implemented, 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.

The INITIATOR RESPONSE TIMEOUT field contains the 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 value of zero indicates that the SSP target port shall disable the initiator response timeout timer. If this mode page is not implemented, the logical unit shall not implement an initiator response timeout timer. This value is enforced by the transport layer (see 9.2.6.3).

10.2.6.2.3 Protocol-Specific Port mode page - Phy Control And Discover subpage

The Phy Control And Discover subpage contains phy-specific parameters. Parameters in this page shall affect only the referenced phy.

Table 130 defines the format of the subpage for SAS SSP.

Table 130 — Protocol-Specific Port Control mode page for SAS SSP - Phy Control And Discover subpage

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)						
3								(LSB)
4		Reserved						
6								
7	NUMBER OF PHYS							
SAS phy mode descriptors								
8	First SAS phy mode descriptor							
...	...							
n	Last SAS phy mode descriptor							

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-3.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The SPF field shall be set to one to access the long format mode pages.

The PAGE CODE field shall be set to 19h.

The SUBPAGE CODE field shall be set to 01h.

The PAGE LENGTH field shall be set to (4 + (the value of the NUMBER OF PHYS field) × (the length in bytes of the SAS phy mode descriptor)).

The NUMBER OF PHYS field contains the number of phys in the SAS target device and indicates the number of SAS phy mode descriptors that follow. This field shall not be changeable.

A SAS phy mode descriptor shall be included for each phy in the SAS target device, starting with the lowest numbered phy and ending with the highest numbered phy.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 131 defines the SAS phy mode descriptor.

Table 131 — SAS phy mode descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	Restricted (for SMP PHY CONTROL function's PHY OPERATION field)							
3	Reserved							
4	Reserved	ATTACHED DEVICE TYPE			Reserved			
5	Reserved				NEGOTIATED PHYSICAL 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 PHY IDENTIFIER field, ATTACHED DEVICE TYPE field, NEGOTIATED PHYSICAL LINK RATE field, ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, ATTACHED SMP INITIATOR PORT bit, ATTACHED SSP TARGET PORT bit, ATTACHED STP TARGET PORT bit, ATTACHED SMP TARGET PORT bit, SAS ADDRESS field, ATTACHED SAS ADDRESS field, ATTACHED PHY IDENTIFIER, HARDWARE MINIMUM PHYSICAL LINK RATE field, and HARDWARE

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

MAXIMUM PHYSICAL LINK RATE field are defined in the SMP DISCOVER function (see 10.4.3.5). These fields shall not be changeable.

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field are defined in the SMP PHY CONTROL function (see 10.4.3.10).

10.2.6.3 Protocol-Specific Logical Unit mode page

SSP logical units shall not include the Protocol-Specific Logical Unit mode page (see SPC-3).

10.2.7 SCSI log parameters

10.2.7.1 Protocol-Specific log page

The Protocol Specific log page for SAS defined in table 132 is used to report errors that have occurred on the SAS target device's phy(s).

Table 132 — Protocol-Specific log page for SAS

Byte\Bit	7	6	5	4	3	2	1	0
0	PAGE CODE (18h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (m - 3)						
3								(LSB)
Protocol-specific log parameters								
4	First protocol-specific log parameter							
...	...							
	n^{th} protocol-specific log parameter							
m								

The PAGE CODE field shall be set to 18h.

The PAGE LENGTH field shall be set to the total length in bytes of the log parameters.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 133 defines the format for a SAS log parameter.

Table 133 — Protocol-Specific log parameter format for SAS

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1	PARAMETER CODE (relative target port identifier)							(LSB)
2	DU	DS	TSD	ETC	TMC		LBIN	LP
3	PARAMETER LENGTH (y - 3)							
4	Reserved				PROTOCOL IDENTIFIER (6h)			
5	Reserved							
6								
7	NUMBER OF PHYS							
SAS phy log descriptors								
8	First SAS phy log descriptor							
	...							
	Last SAS phy log descriptor							
y								

The PARAMETER CODE field contains the relative target port identifier (see SPC-3) of the SSP target port that this log parameter describes.

Table 134 defines the values for the log parameter control bits for this log parameter.

Table 134 — Parameter control bits for SAS log parameters

Bit	Value	Description
DU	0	The value is provided by the device server.
DS	0	The device server supports saving of the parameter.
TSD	0	The device server manages saving of the parameter.
ETC	0	No threshold comparison is made on this value.
TMC	any	This field is ignored when the ETC bit is 0.
LBIN	1	The parameter is in binary format.
LP	1	The parameter is a list parameter.

The PARAMETER LENGTH field is set to the length of the log parameter minus three.

The PROTOCOL IDENTIFIER field is set to 6h.

The NUMBER OF PHYS field contains the number of SAS phy log descriptors that follow.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 135 defines the SAS phy log descriptor. Each SAS phy log descriptor is the same length.

Table 135 — SAS phy log descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	Reserved							
3								
4	Reserved	ATTACHED DEVICE TYPE			Reserved			
5	Reserved				NEGOTIATED PHYSICAL 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	(MSB)	INVALID DWORD COUNT						
35								(LSB)
36	(MSB)	RUNNING DISPARITY ERROR COUNT						
39								(LSB)
40	(MSB)	LOSS OF DWORD SYNCHRONIZATION						
43								(LSB)
44	(MSB)	PHY RESET PROBLEM						
47								(LSB)

The PHY IDENTIFIER field, ATTACHED DEVICE TYPE field, NEGOTIATED PHYSICAL LINK RATE field, ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, ATTACHED SMP INITIATOR PORT bit, ATTACHED SSP TARGET PORT bit, ATTACHED STP TARGET PORT bit, ATTACHED SMP TARGET PORT bit, SAS ADDRESS field, ATTACHED SAS ADDRESS field, and ATTACHED PHY IDENTIFIER field are defined in the SMP DISCOVER function (see 10.4.3.5).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The INVALID DWORD COUNT field, RUNNING DISPARITY ERROR COUNT field, LOSS OF DWORD SYNCHRONIZATION field, and PHY RESET PROBLEM COUNT field are each defined in the SMP REPORT PHY ERROR LOG response data (see 10.4.3.6).

10.2.8 SCSI power conditions

10.2.8.1 SCSI power conditions overview

The logical unit power condition states from the Power Condition mode page (see SPC-3) and START STOP UNIT command (see SBC-2), if implemented, shall interact with the NOTIFY (ENABLE SPINUP) primitive (see 7.2.5.9) to control temporary consumption of additional power (e.g., spin-up of rotating media) as described in this subclause.

The logical unit uses NOTIFY (ENABLE SPINUP) to:

a) automatically spin-up after power on; and
b) delay spin-ups requested by START STOP UNIT commands.

10.2.8.2 SA_PC (SCSI application layer power condition) state machine

10.2.8.2.1 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.

NOTE 29 - This state machine is an enhanced version of the logical unit power condition state machines described in SPC-3 and SBC-2.

This state machine consists of the following states:

a) SA_PC_0:Powered_On (see 10.2.8.2.2)(initial state);
b) SA_PC_1:Active (see 10.2.8.2.3);
c) SA_PC_2:Idle (see 10.2.8.2.4);
d) SA_PC_3:Standby (see 10.2.8.2.5);
e) SA_PC_4:Stopped (see 10.2.8.2.6)(specific to SBC-2 logical units);
f) SA_PC_5:Active_Wait (see 10.2.8.2.7)(specific to SAS devices); and
g) SA_PC_6:Idle_Wait (see 10.2.8.2.8)(specific to SAS devices).

This state machine stall start in the SA_PC_0:Powered_On state after power on.

For transitions based on receipt of a START STOP UNIT command with the IMMED bit is set to one, the command may complete with GOOD status before any operation that occurs as a result of the value in the POWER CONDITIONS field completes.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Figure 118 describes the SA_PC state machine.

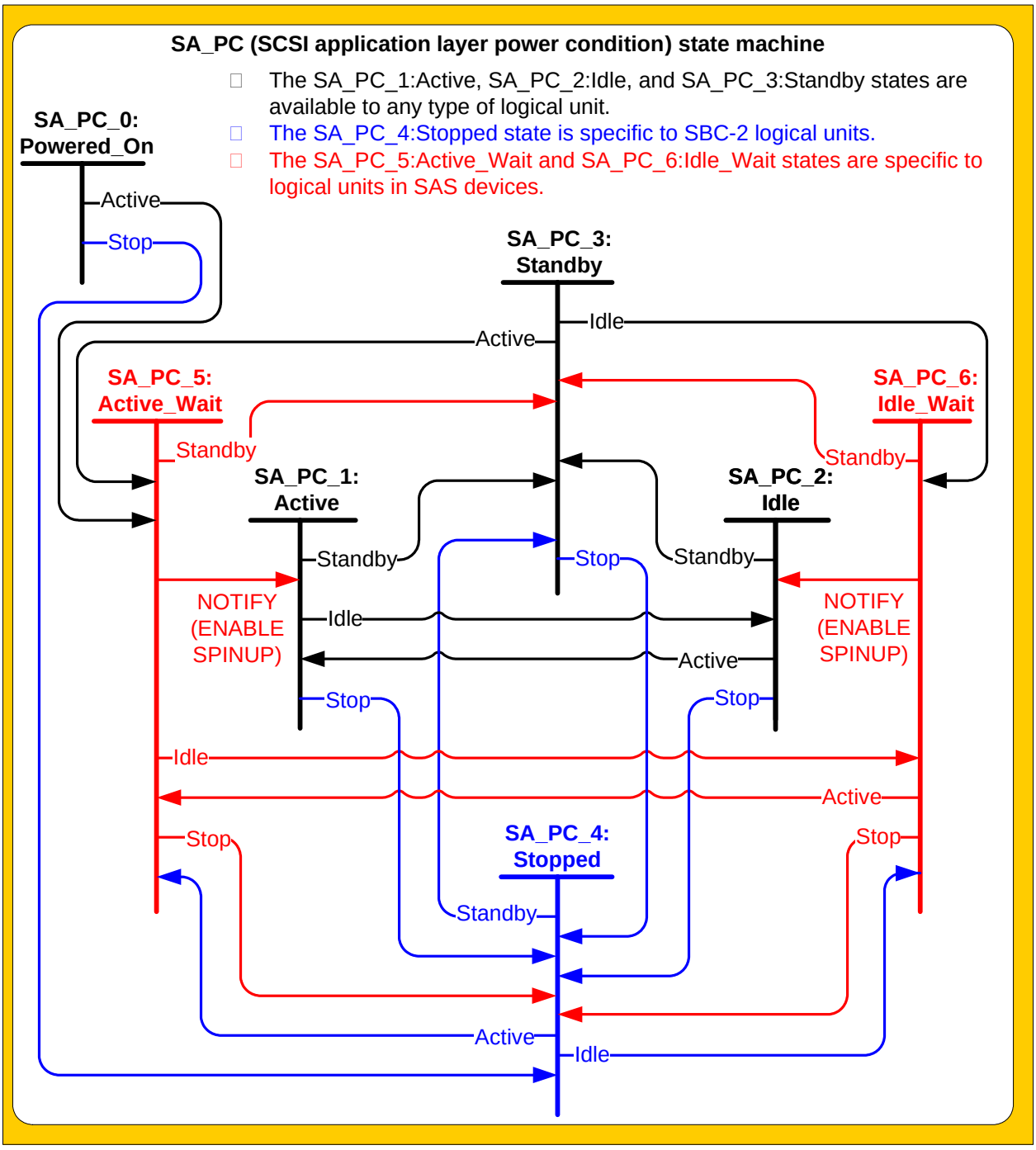


Figure 118 — SA_PC (SCSI application layer power condition) state machine for SAS

10.2.8.2.2 SA_PC_0:Powered_On state

10.2.8.2.2.1 State description

This state shall be entered upon power on. This state consumes zero time.

10.2.8.2.2.2 Transition SA_PC_0:Powered_On to SA_PC_4:Stopped

This transition shall occur if the SAS device has been configured to start in the SA_PC_4:Stopped state.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

10.2.8.2.2.3 Transition SA_PC_0:Powered_On to SA_PC_5:Active_Wait

This transition shall occur if the SAS device has been configured to start in the SA_PC_5:Active_Wait state.

10.2.8.2.3 SA_PC_1:Active state

10.2.8.2.3.1 State description

While in this state, rotating media in block devices shall be active (i.e., rotating or spinning).

See SPC-3 for more details about this state.

10.2.8.2.3.2 Transition SA_PC_1:Active to SA_PC_2:Idle

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to IDLE is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_IDLE_0 is received; or
c) the Power Condition mode page idle condition timer expires.

10.2.8.2.3.3 Transition SA_PC_1:Active to SA_PC_3:Standby

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to STANDBY is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_STANDBY_0 is received; or
c) the Power Condition mode page standby condition timer expires.

10.2.8.2.3.4 Transition SA_PC_1:Active to SA_PC_4:Stopped

This transition shall occur if:

a) a START STOP UNIT command with the START bit set to zero is received.

10.2.8.2.4 SA_PC_2:Idle state

10.2.8.2.4.1 State description

While in this state, rotating media in block devices shall be active (i.e., rotating or spinning).

See SPC-3 for more details about this state.

10.2.8.2.4.2 Transition SA_PC_2:Idle to SA_PC_1:Active

This transition shall occur if:

a) a START STOP UNIT command with the START bit set to one is received;
b) a START STOP UNIT command with the POWER CONDITION field set to ACTIVE is received; or
c) a command is received which requires the active power condition.

10.2.8.2.4.3 Transition SA_PC_2:Idle to SA_PC_3:Standby

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to STANDBY is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_STANDBY_0 is received; or
c) the Power Condition mode page standby condition timer expires.

10.2.8.2.4.4 Transition SA_PC_2:Idle to SA_PC_4:Stopped

This transition shall occur if:

a) a START STOP UNIT command with the START bit set to zero is received.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.8.2.5 SA_PC_3:Standby state

10.2.8.2.5.1 State description

While in this state, rotating media in block devices shall be stopped.

See SPC-3 for more details about this state.

10.2.8.2.5.2 Transition SA_PC_3:Standby to SA_PC_4:Stopped

This transition shall occur if:

a) a START STOP UNIT command with the START bit set to zero is received.

10.2.8.2.5.3 Transition SA_PC_3:Standby to SA_PC_5:Active_Wait

This transition shall occur if:

a) a START STOP UNIT command with the START bit set to one is received;
b) a START STOP UNIT command with the POWER CONDITION field set to ACTIVE is received; or
c) a command is received which requires the active power condition.

For transitions based on a START STOP UNIT command with the IMMED bit set to zero, the command shall not complete with GOOD status until this state machine reaches the SA_PC_1:Active state.

10.2.8.2.5.4 Transition SA_PC_3:Standby to SA_PC_6:Idle_Wait

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to IDLE is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_IDLE_0 is received; or
c) a command is received which requires the idle power condition.

For transitions based on a START STOP UNIT command with the IMMED bit set to zero, the command shall not complete with GOOD status until this state machine reaches the SA_PC_2:Idle state.

10.2.8.2.6 SA_PC_4:Stopped state

10.2.8.2.6.1 State description

This state is only implemented in block devices.

While in this state, rotating media shall be stopped.

See SBC-2 for more details about this state.

10.2.8.2.6.2 Transition SA_PC_4:Stopped to SA_PC_3:Standby

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to STANDBY is received; or
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_STANDBY_0 is received.

10.2.8.2.6.3 Transition SA_PC_4:Stopped to SA_PC_5:Active_Wait

This transition shall occur if:

a) a START STOP UNIT command with the START bit set to one is received; or
b) a START STOP UNIT command with the POWER CONDITION field set to ACTIVE is received.

If the IMMED bit is set to zero, the START STOP UNIT command shall not complete with GOOD status until this state machine reaches the SA_PC_1:Active state.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.8.2.6.4 Transition SA_PC_4:Stopped to SA_PC_6:Idle_Wait

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to IDLE is received; or
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_IDLE_0 is received.

If the IMMED bit is set to zero, the START STOP UNIT command shall not complete with GOOD status until this state machine reaches the SA_PC_2:Idle state.

10.2.8.2.7 SA_PC_5:Active_Wait state

10.2.8.2.7.1 State description

This state shall only be implemented in SAS devices.

While in this state, rotating media in block devices shall be stopped and the device server is not capable of processing media access commands. Any media access commands received while in this state shall cause the device server to terminate the command with CHECK CONDITION status with a sense key of NOT READY and an additional sense code of LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED.

10.2.8.2.7.2 Transition SA_PC_5:Active_Wait to SA_PC_1:Active

This transition shall occur if:

a) a NOTIFY (ENABLE SPINUP) is detected; or
b) the SAS device does not consume additional power as a result of the transition to SA_PC_1:Active.

10.2.8.2.7.3 Transition SA_PC_5:Active_Wait to SA_PC_3:Standby

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to STANDBY is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_STANDBY_0 is received; or
c) the Power Condition mode page standby condition timer expires.

10.2.8.2.7.4 Transition SA_PC_5:Active_Wait to SA_PC_4:Stopped

This transition shall occur if a START STOP UNIT command with the START bit set to zero is received.

10.2.8.2.7.5 Transition SA_PC_5:Active_Wait to SA_PC_6:Idle_Wait

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to IDLE is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_IDLE_0 is received; or
c) the Power Condition mode page idle condition timer expires.

For transitions based on a START STOP UNIT command with the IMMED bit set to zero, the command shall not complete with GOOD status until this state machine reaches the SA_PC_2:Idle state.

10.2.8.2.8 SA_PC_6:Idle_Wait state

10.2.8.2.8.1 State description

This state shall only be implemented in SAS devices.

While in this state, rotating media in block devices shall be stopped and the device server is not capable of processing media access commands. Any media access commands received while in this state shall cause the device server to terminate the command with CHECK CONDITION status with a sense key of NOT READY and an additional sense code of LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.8.2.8.2 Transition SA_PC_6:Idle_Wait to SA_PC_2:Idle

This transition shall occur if:

a) a NOTIFY (ENABLE SPINUP) is detected; or
b) the SAS device does not consume additional power as a result of the transition to SA_PC_2:Idle.

10.2.8.2.8.3 Transition SA_PC_6:Idle_Wait to SA_PC_3:Standby

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to STANDBY is received;
b) a START STOP UNIT command with the POWER CONDITION field set to FORCE_STANDBY_0 is received; or
c) the Power Condition mode page standby condition timer expires.

10.2.8.2.8.4 Transition SA_PC_6:Idle_Wait to SA_PC_4:Stopped

This transition shall occur if a START STOP UNIT command with the START bit set to zero is received.

10.2.8.2.8.5 Transition SA_PC_6:Idle_Wait to SA_PC_5:Active_Wait

This transition shall occur if:

a) a START STOP UNIT command with the POWER CONDITION field set to ACTIVE is received; or
b) a command is received which requires the active power condition.

For transitions based on a START STOP UNIT command with the IMMED bit set to zero, the command shall not complete with GOOD status until this state machine reaches the SA_PC_1:Active state.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2.9 SCSI vital product data (VPD)

Each logical unit in a SAS target device that is a SCSI target device shall include the identification descriptors listed in table 136 in the Device Identification VPD page (83h) returned by the INQUIRY command (see SPC-3).

Table 136 — Device Identification VPD page required identification descriptors

Field in identification descriptor	Identification descriptor			
	Logical unit name	**Target port identifier**	**Relative target port identifier**	**Target device name**
IDENTIFIER TYPE	3h (i.e., NAA)	3h (i.e., NAA)	4h (i.e., relative target port identifier)	3h (i.e., NAA)
ASSOCIATION	0h (i.e., logical unit)	1h (i.e., SCSI target port)	1h (i.e., SCSI target port)	2h (i.e., SCSI target device)
CODE SET	1h (i.e., binary)	1h (i.e., binary)	1h (i.e., binary)	1h (i.e., binary)
IDENTIFIER LENGTH	8 [a] or 16 [b]	8	4	8
PIV (PROTOCOL IDENTIFIER VALID)	0	1	0	1
PROTOCOL IDENTIFIER	Any	6h (i.e., SAS)	Any	6h (i.e., SAS)
IDENTIFIER	NAA IEEE Registered format [a] or NAA IEEE Registered Extended format [b]	NAA IEEE Registered format [a] [c]	SCSI target ports shall be numbered sequentially starting with 00000001h [d]	NAA IEEE Registered format [a]

[a] The IDENTIFIER field contains an NAA field set to 5h (i.e., IEEE Registered); the IDENTIFIER LENGTH field is set to 8.
[b] The IDENTIFIER field contains an NAA field set to 6h (i.e., IEEE Registered Extended); the IDENTIFIER LENGTH field is set to 16.
[c] The IDENTIFIER field contains the SAS address of the SSP target port through which the INQUIRY command was received.
[d] The IDENTIFIER field contains the relative target port identifier of the SSP target port through which the INQUIRY command was received.

The SAS target device shall use different identifiers for each logical unit name, each target port identifier, and the target device name.

Logical units may include additional identification descriptors than those required by this standard (e.g., SCSI target devices with SCSI target ports using other SCSI transport protocols may return additional target device names for those other SCSI transport protocols.)

10.3 ATA application layer

No SAS-specific ATA features are defined.

--`,,,``,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

10.4 Management application layer

10.4.1 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 5.4. All SAS target devices using the SAS plug connector (see 5.2.3.2) shall support the READY LED signal.

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.

A SAS target device containing an SSP target port shall transmit the READY LED signal using the following patterns:

a) If the SAS target device is in the standby or stopped power condition state (see 10.2.8), it shall assert the READY LED signal while processing a command (i.e., the LED is usually off, but flashes on when commands are processed). The SAS target device may be removed with no danger of mechanical or electrical damage in this state;
b) If the SAS target device has rotating media and is in the process of performing a spin-up or spin-down, it shall assert and negate the READY LED signal with a 1 s ± 0,1 s cycle using a 50 % ± 10 % duty cycle (i.e., LED is on for 0,5 s and off for 0,5 s);
c) If the SAS target device is in the active or idle power condition state (see 10.2.8), it shall assert the READY LED signal continuously except when the SAS target device is processing a command. When processing a command, the SAS target device shall toggle the READY LED signal in a vendor-specified manner (e.g., the LED is usually on, but is momentarily off when commands are processed); or
d) If the SAS target device is formatting the media, it shall toggle the READY LED signal in a vendor-specific manner (e.g., with each cylinder change on a disk drive).

SAS target devices with rotating media transition from pattern c) to pattern b) during the spin-down process. When the SAS target device has reached a state stable enough for it to be removed without mechanical or electrical damage, it shall change from pattern b) to pattern a).

SAS target devices without SSP target ports may transmit the READY LED signal using vendor-specific patterns.

10.4.2 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.

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.4.3 SMP functions

10.4.3.1 SMP function request frame format

An SMP request frame is sent by an SMP initiator port to request an SMP function be performed by a management device server. Table 137 defines the SMP request frame format.

Table 137 — SMP request frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION							
2	Reserved							
3								
4	ADDITIONAL REQUEST BYTES							
	Fill bytes, if needed							
n - 3	(MSB)	CRC						
n								(LSB)

The SMP FRAME TYPE field is included in each frame format defined in this clause, although that field is parsed by the SMP transport layer (see 9.4). The SMP FRAME TYPE field is set to 40h.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

The FUNCTION field specifies which SMP function is being requested and is defined in table 138. If the value in the FUNCTION field is not supported by the SMP target port, it shall return a function result of UNKNOWN SMP FUNCTION as described in table 140.

Table 138 — SMP functions

Code	SMP function	Description	Request frame size (in bytes)	Response frame size (in bytes)	Reference
00h	REPORT GENERAL	Return general information about the device	8	32	10.4.3.3
01h	REPORT MANUFACTURER INFORMATION	Return vendor and product identification	8	64	10.4.3.4
02h - 0Fh	Reserved for general SMP input functions				
10h	DISCOVER	Return information about the specified phy	16	56	10.4.3.5
11h	REPORT PHY ERROR LOG	Return error logging information about the specified phy	16	32	10.4.3.6
12h	REPORT PHY SATA	Return information about a phy currently attached to a SATA device	16	60	10.4.3.7
13h	REPORT ROUTE INFORMATION	Return route table information	16	44	10.4.3.8
14h - 1Fh	Reserved for phy-based SMP input functions				
20h - 3Fh	Reserved for SMP input functions				
40h - 7Fh	Vendor specific				
80h - 8Fh	Reserved for general SMP output functions				
90h	CONFIGURE ROUTE INFORMATION	Change route table information	44	8	10.4.3.9
91h	PHY CONTROL	Request actions by the specified phy	44	8	10.4.3.10
92h - 9Fh	Reserved for phy-based SMP output functions				
A0h - BFh	Reserved for SMP output functions				
C0h - FFh	Vendor specific				

The ADDITIONAL REQUEST BYTES field definition and length is based on the SMP function. The maximum size of the ADDITIONAL REQUEST BYTES field is 1 024 bytes, making the maximum size of the frame 1 032 bytes (1 024 bytes of data + 4 bytes of header + 4 bytes of CRC).

Fill bytes shall be included after the ADDITIONAL REQUEST BYTES field so the CRC field is aligned on a four byte boundary. The contents of the fill bytes are vendor specific.

The CRC field is included in each request frame format defined in this clause, although that field is defined by the SMP transport layer (see 9.4.1) and parsed by the SMP link layer (see 7.18).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.4.3.2 SMP function response frame format

An SMP response frame is sent by an SMP target port in response to an SMP request frame. Table 139 defines the SMP response frame format.

Table 139 — SMP response frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION							
2	FUNCTION RESULT							
3	Reserved							
4	ADDITIONAL RESPONSE BYTES							
	Fill bytes, if needed							
n - 3	(MSB) CRC							
n								(LSB)

The SMP FRAME TYPE field is included in each frame format defined in this clause, although that field is parsed by the SMP transport layer (see 9.4). The SMP FRAME TYPE field is set to 41h.

The FUNCTION field indicates the SMP function to which this frame is a response, and is defined in table 138 in 10.4.3.1.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The FUNCTION RESULT field is defined in table 140.

Table 140 — Function results

Code	Name	SMP function(s)	Description
00h	SMP FUNCTION ACCEPTED	All	The SMP target port supports the SMP function; the ADDITIONAL RESPONSE BYTES field contains the requested information.
01h	UNKNOWN SMP FUNCTION	Unknown	The SMP target port does not support the requested SMP function; the ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
02h	SMP FUNCTION FAILED	All	The SMP target port supports the SMP function, but the requested SMP function failed. The ADDITIONAL RESPONSE BYTES may be present but shall be ignored.
03h	INVALID REQUEST FRAME LENGTH	All	The SMP target port supports the SMP function, but the request frame length was invalid (i.e., did not match the frame size defined for the function). The ADDITIONAL RESPONSE BYTES may be present but shall be ignored.
10h	PHY DOES NOT EXIST	DISCOVER, REPORT PHY ERROR LOG, REPORT PHY SATA, REPORT ROUTE INFORMATION, CONFIGURE ROUTE INFORMATION, PHY CONTROL	The phy specified by the PHY IDENTIFIER field does not exist (i.e., the value is not within the range of zero to the value of the NUMBER OF PHYS field reported in the REPORT GENERAL function). The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
11h	INDEX DOES NOT EXIST	REPORT ROUTE INFORMATION, CONFIGURE ROUTE INFORMATION	The phy specified by the PHY IDENTIFIER field 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 REPORT GENERAL function). The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
12h	PHY DOES NOT SUPPORT SATA	REPORT PHY SATA	The phy specified by the PHY IDENTIFIER field is not part of an STP target port. The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
13h	UNKNOWN PHY OPERATION	PHY CONTROL	The operation specified by the PHY OPERATION field is unknown. The SMP function had no affect. The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
All others	Reserved		

The ADDITIONAL RESPONSE BYTES field definition depends on the SMP function requested. The maximum size of the ADDITIONAL RESPONSE BYTES field is 1 024 bytes, making the maximum size of the frame 1 032 bytes (1 024 bytes of data + 4 bytes of header + 4 bytes of CRC).

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Fill bytes shall be included after the ADDITIONAL RESPONSE BYTES field so the CRC field is aligned on a four byte boundary. The contents of the fill bytes are vendor specific.

The CRC field is included in each response frame format defined in this clause, although that field is defined by the SMP transport layer (see 9.4.1) and parsed by the SMP link layer (see 7.18).

10.4.3.3 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 SMP target ports.

Table 141 defines the request format.

Table 141 — REPORT GENERAL request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (00h)							
2	Reserved							
3								
4	(MSB)							
7	CRC							(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 00h.

The CRC field is defined in 10.4.3.1.

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 142 defines the response format.

Table 142 — REPORT GENERAL response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (00h)							
2	FUNCTION RESULT							
3	Reserved							
4	(MSB)	EXPANDER CHANGE COUNT						
5								(LSB)
6	(MSB)	EXPANDER ROUTE INDEXES						
7								(LSB)
8	Reserved							
9	NUMBER OF PHYS							
10	Reserved						CONFIGURING	CONFIGURABLE ROUTE TABLE
11	Reserved							
27								
28	(MSB)	CRC						
31								(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 00h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The EXPANDER CHANGE COUNT field counts the number of BROADCAST (CHANGE)s originated by an expander device (see 7.11). Expander devices shall support this field. Other device types shall not support this field. This field shall be set to zero at power on. The expander device shall increment this field at least once when it transmits a BROADCAST (CHANGE) for one of the following reasons:

a) after an expander phy has lost dword synchronization and restarted the link reset sequence (see 6.7);
b) after the link reset sequence completes on an expander phy; or
c) after a self-configuring expander device has completed configuration and has changed its CONFIGURING bit from one to zero.

The expander device need not increment this field again unless a REPORT GENERAL response is transmitted. This field shall not be incremented when forwarding a BROADCAST (CHANGE) from another expander device. The EXPANDER CHANGE COUNT field shall wrap to zero after the maximum value (i.e., FFFFh) has been reached.

NOTE 30 - Application clients that use the EXPANDER CHANGE COUNT field should read it often enough to ensure that it does not increment a multiple of 65 536 times between reading the field.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The EXPANDER ROUTE INDEXES field contains the maximum number of route indexes per phy for the expander device (see 4.6.7.3). Expander devices shall support this field. Other device types shall not support this field. Not all phys in an edge expander device are required to support the maximum number indicated by this field.

If an edge expander device supports an expander route table, then the number of expander route indexes for each phy identifier shall be greater than or equal to the number of addressable phys downstream from the edge expander phy.

The NUMBER OF PHYS field contains the number of phys in the device, including any virtual phys.

A CONFIGURING bit set to one indicates that a self-configuring expander device has not completed configuring its expander route table. A CONFIGURING bit set to zero indicates that configuration is complete and the expander device is ready for connection requests. Changes in this bit from one to zero result in a BROADCAST (CHANGE) being originated.

The CONFIGURABLE ROUTE TABLE bit indicates whether the expander device has an expander route table that is required to be configured with the SMP CONFIGURE ROUTE INFORMATION function (see 4.6.7.3). An expander device with a configurable route table shall have the CONFIGURABLE ROUTE TABLE bit set to one. An expander device without a configurable route table shall have the CONFIGURABLE ROUTE TABLE bit set to zero.

The CRC field is defined in 10.4.3.2.

10.4.3.4 REPORT MANUFACTURER INFORMATION function

The REPORT MANUFACTURER INFORMATION function returns vendor and product identification. This SMP function may be implemented by any SMP target port.

Table 143 defines the request format.

Table 143 — REPORT MANUFACTURER INFORMATION request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (01h)							
2	Reserved							
3								
4	(MSB) CRC							
7								(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 01h.

The CRC field is defined in 10.4.3.1.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 144 defines the response format.

Table 144 — 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	Reserved							
4	Ignored							
7								
8	Reserved							
9	Ignored							
10								
11	Reserved							
12	(MSB)	VENDOR IDENTIFICATION						
19								(LSB)
20	(MSB)	PRODUCT IDENTIFICATION						
35								(LSB)
36	(MSB)	PRODUCT REVISION LEVEL						
39								(LSB)
40	Vendor specific							
59								
60	(MSB)	CRC						
63								(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 01h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The VENDOR IDENTIFICATION field contains eight bytes of ASCII data identifying the vendor of the product. 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-3 and on the T10 web site (http://www.t10.org).

The PRODUCT IDENTIFICATION field contains sixteen bytes of ASCII data as defined by the vendor. The data shall be left aligned within the field.

The PRODUCT REVISION LEVEL field contains four bytes of ASCII data as defined by the vendor. The data shall be left-aligned within the field.

--`,,,`,,,-`-`,,`,,`,`,,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ASCII data fields 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 (highest offset) and the unused bytes shall be filled with space characters (20h).

The CRC field is defined in 10.4.3.2.

10.4.3.5 DISCOVER function

The DISCOVER function returns the physical link configuration information for the specified phy. This SMP function provides information from the IDENTIFY address frame received by the phy and additional phy-specific information. This SMP function shall be implemented by all SMP target ports.

Table 145 defines the request format.

Table 145 — DISCOVER request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (10h)							
2	Reserved							
3								
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	(MSB)							
15	CRC							(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 10h.

The PHY IDENTIFIER field specifies the phy (see 4.2.7) for the link configuration information being requested.

The CRC field is defined in 10.4.3.1.

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 146 defines the response format.

Table 146 — DISCOVER response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (10h)							
2	FUNCTION RESULT							
3	Reserved							
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	Ignored	ATTACHED DEVICE TYPE			Ignored			
13	Reserved				NEGOTIATED PHYSICAL LINK RATE			
14	Reserved				ATTACHED SSP INITIATOR	ATTACHED STP INITIATOR	ATTACHED SMP INITIATOR	ATTACHED SATA HOST
15	Reserved				ATTACHED SSP TARGET	ATTACHED STP TARGET	ATTACHED SMP TARGET	ATTACHED SATA DEVICE
16	SAS ADDRESS							
23								
24	ATTACHED SAS ADDRESS							
31								
32	ATTACHED PHY IDENTIFIER							
33	Reserved							
39								
40	PROGRAMMED MINIMUM PHYSICAL LINK RATE				HARDWARE MINIMUM PHYSICAL LINK RATE			
41	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				HARDWARE MAXIMUM PHYSICAL LINK RATE			
42	PHY CHANGE COUNT							
43	VIRTUAL PHY	Reserved			PARTIAL PATHWAY TIMEOUT VALUE			
44	Reserved				ROUTING ATTRIBUTE			
45	Reserved							
49								
50	Vendor specific							
51								
52	(MSB) CRC							
55								(LSB)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 10h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The PHY IDENTIFIER field indicates the phy for which physical configuration link information is being returned.

The ATTACHED DEVICE TYPE field indicates the DEVICE TYPE value received during the link reset sequence and is defined in table 147. The ATTACHED DEVICE TYPE field shall only be set to a value other than 000b after the identification sequence is complete if a SAS device or expander device is attached or after the initial Register - Device to Host FIS has been received if a SATA device is attached.

Table 147 — Attached device types

Code	Description
000b	No device attached
001b	End device
010b	Edge expander device
011b	Fanout expander device
All others	Reserved

The NEGOTIATED PHYSICAL LINK RATE field is defined in table 148 and indicates the physical link rate negotiated during the link reset sequence. The negotiated physical link rate may be less than the programmed minimum physical link rate or greater than the programmed maximum physical link rate if the programmed physical link rates have been changed since the last link reset sequence.

Table 148 — Negotiated physical link rate

Code	Description
0h	Phy is enabled; unknown physical link rate. [a]
1h	Phy is disabled.
2h	Phy is enabled; the phy obtained dword synchronization for at least one physical link rate during speed negotiation (either SAS or SATA), but the speed negotiation sequence failed (i.e., the last speed negotiation window, using a physical link rate expected to succeed, failed). These failures may be logged in the SMP REPORT PHY ERROR LOG function (see 10.4.3.6) and/or the Protocol-Specific log page (see 10.2.7.1).
3h	Phy is enabled; detected a SATA device and entered the SATA spinup hold state. The LINK RESET and HARD RESET operations in the SMP PHY CONTROL function (see 10.4.3.10) may be used to release the phy. This field shall be updated to this value after the SATA OOB sequence completes if SATA spinup hold is supported.
8h	Phy is enabled; This field shall be updated to this value after the speed negotiation sequence completes indicating a negotiated physical link rate of 1,5 Gbps.
9h	Phy is enabled; This field shall be updated to this value after the speed negotiation sequence completes indicating a negotiated physical link rate of 3,0 Gbps.
All others	Reserved.

[a] This code may be used by an application client in its local data structures to indicate an unknown negotiated physical link rate (e.g., before the discover process has queried the phy).

The ATTACHED SSP INITIATOR PORT bit indicates the SSP INITIATOR value received in the IDENTIFY address frame (see 7.8.2) during the link reset sequence.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The ATTACHED STP INITIATOR PORT bit indicates the STP INITIATOR value received in the IDENTIFY address frame (see 7.8.2) during the link reset sequence.

The ATTACHED SMP INITIATOR PORT bit indicates the SMP INITIATOR value received in the IDENTIFY address frame (see 7.8.2) during the link reset sequence.

The ATTACHED SSP TARGET PORT bit indicates the SSP TARGET value received in the IDENTIFY address frame (see 7.8.2) during the link reset sequence.

The ATTACHED STP TARGET PORT bit indicates the STP TARGET value received in the IDENTIFY address frame (see 7.8.2) during the link reset sequence.

The ATTACHED SMP TARGET PORT bit indicates the SMP TARGET value received in the IDENTIFY address frame (see 7.8.2) during the link reset sequence.

The ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, ATTACHED SMP INITIATOR PORT bit, ATTACHED SSP TARGET PORT bit, ATTACHED STP TARGET PORT bit, ATTACHED SMP TARGET PORT bit, and ATTACHED SMP TARGET PORT bit shall be updated at the end of the identification sequence.

An ATTACHED SATA DEVICE bit set to one indicates a SATA device port is attached. An ATTACHED SATA DEVICE bit set to zero indicates a SATA device port is not attached. This bit shall be updated after the SATA OOB sequence completes and before the SATA speed negotiation sequence begins (i.e., at SATA spinup hold time (see 6.9)).

An ATTACHED SATA HOST bit set to one indicates a SATA host port is attached. An ATTACHED SATA HOST bit set to zero indicates a SATA host port is not attached.

NOTE 31 - Supports for SATA hosts is outside the scope of this standard.

The SAS ADDRESS field contains the SAS address transmitted by this phy during an identification sequence. If the phy is an expander phy, the SAS ADDRESS field contains the SAS address of the expander device. If the phy is a SAS phy, the SAS ADDRESS field contains the SAS address of the SAS port.

The ATTACHED SAS ADDRESS field contains the SAS address received by this phy during the identification sequence, which is either:

a) the SAS address of the attached SAS port if the attached port is a SAS port;
b) the SAS address of the attached expander device if the attached port is an expander port; or
c) the SAS address provided for the attached port if the attached port is a SATA device port.

If the ATTACHED DEVICE TYPE field is set to 000b (i.e., no device attached), the ATTACHED SAS ADDRESS field is invalid. The ATTACHED SAS ADDRESS field shall be updated:

a) after the identification sequence completes, if a SAS device or expander device is attached; or
b) after the SATA OOB sequence completes, if a SATA device is attached.

The ATTACHED PHY IDENTIFIER field contains the phy identifier received by this phy during the identification sequence, which is either:

a) the phy identifier of the attached SAS port if the attached port is a SAS port;
b) the phy identifier of the attached expander device if the attached port is an expander port; or
c) the phy identifier provided for the attached port if the attached port is a SATA device port.

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field indicates the minimum physical link rate set by the PHY CONTROL function (see 10.4.3.10). The values are defined in table 149. The default value shall be the value of the HARDWARE MINIMUM PHYSICAL LINK RATE field.

The HARDWARE MINIMUM PHYSICAL LINK RATE field indicates the minimum physical link rate supported by the phy. The values are defined in table 150.

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The PROGRAMMED MAXIMUM PHYSICAL LINK RATE field indicates the maximum physical link rate set by the PHY CONTROL function (see 10.4.3.10). The values are defined in table 149. The default value shall be the value of the HARDWARE MAXIMUM PHYSICAL LINK RATE field.

Table 149 — Programmed minimum and maximum physical link rates

Code	Description
0h	Not programmable
8h	1,5 Gbps
9h	3,0 Gbps
All others	Reserved

The HARDWARE MAXIMUM PHYSICAL LINK RATE field indicates the maximum physical link rate supported by the phy. The values are defined in table 150.

Table 150 — Hardware minimum and maximum physical link rates

Code	Description
8h	1,5 Gbps
9h	3,0 Gbps
All others	Reserved

The PHY CHANGE COUNT field counts the number of BROADCAST (CHANGE)s originated by an expander phy. Expander devices shall support this field. Other device types shall not support this field. This field shall be set to zero at power on. The expander device shall increment this field at least once when it transmits a BROADCAST (CHANGE) for one of the following reasons:

a) after the expander phy has lost dword synchronization and restarted the link reset sequence (see 6.7); or
b) after the link reset sequence completes on the expander phy.

The expander device need not increment the PHY CHANGE COUNT field again unless a DISCOVER response is transmitted. The PHY CHANGE COUNT field shall wrap to zero after the maximum value (i.e., FFh) has been reached.

NOTE 32 - Application clients that use the PHY CHANGE COUNT field should read it often enough to ensure that it does not increment a multiple of 256 times between reading the field.

An VIRTUAL PHY bit set to one indicates the phy is part of an virtual port and the attached device is contained within the expander device. An VIRTUAL PHY bit set to zero indicates the phy is a physical phy and the attached device is not contained within the expander device.

The PARTIAL PATHWAY TIMEOUT VALUE field indicates the partial pathway timeout value in microseconds (see 7.12.4.3).

NOTE 33 - The recommended default value for PARTIAL PATHWAY TIMEOUT VALUE is 7 μs. The partial pathway timeout value may be set by the PHY CONTROL function (see 10.4.3.10).

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The ROUTING ATTRIBUTE field indicates the routing attribute supported by the phy (see 4.6.7.1) and is defined in table 151.

Table 151 — Routing attributes

Code	Name	Description
0h	Direct routing attribute	Direct routing method for attached end devices. Attached expander devices are not supported on this phy.
1h	Subtractive routing attribute	Either: a) subtractive routing method for attached expander devices; or b) direct routing method for attached end devices.
2h	Table routing attribute	Either: a) table routing method for attached expander devices; or b) direct routing method for attached end devices.
All others	Reserved	

The ROUTING ATTRIBUTE field shall not change based on the attached device type.

The CRC field is defined in 10.4.3.2.

10.4.3.6 REPORT PHY ERROR LOG function

The REPORT PHY ERROR LOG returns error logging information about the specified phy. This SMP function may implemented by any SMP target port.

Table 152 defines the request format.

Table 152 — REPORT PHY ERROR LOG request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (11h)							
2	Reserved							
3								
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	(MSB)							
15	CRC							(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 11h.

The PHY IDENTIFIER field specifies the phy (see 4.2.7) for which information shall be reported.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The CRC field is defined in 10.4.3.1.

Table 153 defines the response format.

Table 153 — REPORT PHY ERROR LOG response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (11h)							
2	FUNCTION RESULT							
3	Reserved							
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	(MSB)	INVALID DWORD COUNT						
15								(LSB)
16	(MSB)	RUNNING DISPARITY ERROR COUNT						
19								(LSB)
20	(MSB)	LOSS OF DWORD SYNCHRONIZATION COUNT						
23								(LSB)
24	(MSB)	PHY RESET PROBLEM COUNT						
27								(LSB)
28	(MSB)	CRC						
31								(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 11h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The PHY IDENTIFIER field indicates the phy (see 4.2.7) for which information is being reported.

The INVALID DWORD COUNT field indicates the number of invalid dwords (see 3.1.66) that have been received outside of phy reset sequences (i.e., between when the SP_DWS state machine (see 6.8) sends a Phy Layer Ready (SAS) confirmation and when it sends a Phy Layer Not Ready confirmation to the link layer). The count shall stop at the maximum value.

The RUNNING DISPARITY ERROR COUNT field indicates the number of dwords containing running disparity errors (see 6.2) that have been received outside of phy reset sequences. The count shall stop at the maximum value.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The LOSS OF DWORD SYNCHRONIZATION COUNT field indicates the number of times the phy has lost dword synchronization and restarted the link reset sequence (see 6.7) of phy reset sequences. The count shall stop at the maximum value.

The PHY RESET PROBLEM COUNT field indicates the number of times the phy reset sequence has failed. The count shall stop at the maximum value.

The CRC field is defined in 10.4.3.2.

10.4.3.7 REPORT PHY SATA function

The REPORT PHY SATA function returns information about the SATA state for a specified phy. This SMP function shall be implemented by SMP target ports that share SAS addresses with STP target ports and by SMP target ports in expander devices with STP/SATA bridges. This SMP function shall not be implemented by any other type of SMP target port.

Table 154 defines the request format.

Table 154 — REPORT PHY SATA request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (12h)							
2	Reserved							
3								
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	(MSB)							
15	CRC							(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 12h.

The PHY IDENTIFIER field specifies the phy (see 4.2.7) for which information shall be reported.

The CRC field is defined in 10.4.3.1.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 155 defines the response format.

Table 155 — REPORT PHY SATA response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (12h)							
2	FUNCTION RESULT							
3	Reserved							
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved						AFFILIATIONS SUPPORTED	AFFILIATION VALID
12	Reserved							
15								
16	STP SAS ADDRESS							
23								
24	REGISTER DEVICE TO HOST FIS							
43								
44	Reserved							
47								
48	AFFILIATED STP INITIATOR SAS ADDRESS							
55								
56	(MSB) CRC							
59								(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 12h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The PHY IDENTIFIER field indicates the phy (see 4.2.7) for which information is being reported.

An AFFILIATIONS SUPPORTED bit set to one indicates that affiliations are supported by the STP target port containing the specified phy. An AFFILIATIONS SUPPORTED bit set to zero indicates that affiliations are not supported by the STP target port containing the specified phy.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

The AFFILIATION VALID bit shall be set to one when the AFFILIATED STP INITIATOR SAS ADDRESS field is valid and the STP target port containing the specified phy is maintaining an affiliation (see 7.17.3). The AFFILIATION VALID bit shall be set to zero when no affiliation is being maintained.

The STP SAS ADDRESS field contains the SAS address of the STP target port that contains the specified phy.

The REGISTER DEVICE TO HOST FIS field contains the contents of the initial Register - Device to Host FIS. For an STP/SATA bridge, this is delivered by the attached SATA device after a link reset sequence (see ATA/ATAPI-7 V3). For a native STP target port in an end device, this is directly provided.

The FIS contents shall be stored with little-endian byte ordering (i.e., the first byte, byte 24, contains the FIS Type).

For an STP/SATA bridge, the first byte of the field (i.e., the FIS Type) shall be initialized to zero on power on and whenever the phy has lost dword synchronization and restarted the link reset sequence (see 6.7) to indicate the field is invalid and the attached SATA device has not delivered a Register – Device to Host FIS. The first byte of the field shall be set to 34h when the attached SATA device has delivered the initial Register – Device to Host FIS. The remaining contents of the REGISTER DEVICE TO HOST FIS field shall remain constant until a link reset sequence causes the attached SATA device to deliver another initial Register – Device to Host FIS.

An STP/SATA bridge that receives a connection request for a SATA device that has not successfully delivered the initial Register – Device to Host FIS shall return an OPEN_REJECT (NO DESTINATION).

NOTE 34 - If there is a problem receiving the expected initial Register - Device to Host FIS, the STP/SATA bridge should use SATA_R_ERR to retry until it succeeds. 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.

The AFFILIATED STP INITIATOR SAS ADDRESS field contains the SAS address of the STP initiator port that currently has an affiliation with the STP target port that contains the specified phy.

The CRC field is defined in 10.4.3.2.

10.4.3.8 REPORT ROUTE INFORMATION function

The REPORT ROUTE INFORMATION function returns an expander route entry from the expander route table within an expander device. This SMP function shall be supported by SMP target ports in expander devices if the EXPANDER ROUTE INDEXES field is non-zero in the REPORT GENERAL function. This SMP function may be used as a diagnostic tool to resolve topology issues.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 156 defines the request format.

Table 156 — REPORT ROUTE INFORMATION request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (13h)							
2	Reserved							
3								
4	Reserved							
5								
6	(MSB) EXPANDER ROUTE INDEX							
7								(LSB)
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	(MSB) CRC							
15								(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 13h.

The EXPANDER ROUTE INDEX field specifies the expander route index for the expander route entry being requested.

The PHY IDENTIFIER field specifies the phy for which the expander route entry is being requested.

The CRC field is defined in 10.4.3.1.

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 157 defines the response format.

Table 157 — REPORT ROUTE INFORMATION response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (13h)							
2	FUNCTION RESULT							
3	Reserved							
5								
6	(MSB)							
7	EXPANDER ROUTE INDEX							(LSB)
8	Reserved							
9	PHY IDENTIFIER							
10	Ignored							
11	Reserved							
12	EXPANDER ROUTE ENTRY DISABLED	Ignored						
13	Ignored							
15								
16	ROUTED SAS ADDRESS							
23								
24	Ignored							
35								
36	Reserved							
39								
40	(MSB)							
43	CRC							(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 13h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The EXPANDER ROUTE INDEX field contains the expander route index for the expander route entry being returned.

The PHY IDENTIFIER field contains the phy identifier for the expander route entry being returned.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The EXPANDER ROUTE ENTRY DISABLED bit indicates whether the ECM shall use the expander route entry to route connection requests. If the EXPANDER ROUTE ENTRY DISABLED bit is set to zero, then the ECM shall use the expander route entry to route connection requests. If the EXPANDER ROUTE ENTRY DISABLED bit is set to one, the ECM shall not use the expander route entry to route connection requests.

The ROUTED SAS ADDRESS field contains the routed SAS address in the expander route entry.

The CRC field is defined in 10.4.3.2.

10.4.3.9 CONFIGURE ROUTE INFORMATION function

The CONFIGURE ROUTE INFORMATION function sets an expander route entry within the expander route table of a configurable expander device. This SMP function shall be supported by SMP target ports in expander devices if the CONFIGURABLE ROUTE TABLE field is set to one in the REPORT GENERAL response data. Other SMP target ports shall not support this SMP function.

--`,,,`,,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 158 defines the request format.

Table 158 — CONFIGURE ROUTE INFORMATION request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (90h)							
2	Reserved							
5								
6	(MSB)	EXPANDER ROUTE INDEX						
7								(LSB)
8	Reserved							
9	PHY IDENTIFIER							
10	Reserved							
11								
12	DISABLE EXPANDER ROUTE ENTRY	Ignored						
13	Ignored							
15								
16	ROUTED SAS ADDRESS							
23								
24	Ignored							
35								
36	Reserved							
39								
40	(MSB)	CRC						
43								(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 90h.

The EXPANDER ROUTE INDEX field specifies the expander route index for the expander route entry being configured.

The PHY IDENTIFIER field specifies the phy for which the expander route entry is being configured.

The DISABLE EXPANDER ROUTE ENTRY bit specifies whether the ECM shall use the expander route entry to route connection requests. If the DISABLE EXPANDER ROUTE ENTRY bit is set to zero, then the ECM shall use the

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

expander route entry to route connection requests. If the DISABLE EXPANDER ROUTE ENTRY bit is set to one, the ECM shall not use the expander route entry to route connection requests.

The ROUTED SAS ADDRESS field contains the routed SAS address for the expander route entry being configured.

The CRC field is defined in 10.4.3.1.

Table 159 defines the response format.

Table 159 — CONFIGURE ROUTE INFORMATION response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (90h)							
2	FUNCTION RESULT							
3	Reserved							
4	(MSB)							
				CRC				
7								(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 90h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The CRC field is defined in 10.4.3.2.

10.4.3.10 PHY CONTROL function

The PHY CONTROL function requests actions by the specified phy. This SMP function may implemented by any SMP target port.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 160 defines the request format.

Table 160 — PHY CONTROL request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (91h)							
2	Reserved							
3								
4	Ignored							
7								
8	Reserved							
9	PHY IDENTIFIER							
10	PHY OPERATION							
11	Reserved							UPDATE PARTIAL PATHWAY TIMEOUT VALUE
12	Ignored							
31								
32	PROGRAMMED MINIMUM PHYSICAL LINK RATE				Ignored			
33	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				Ignored			
34	Ignored							
35								
36	Reserved				PARTIAL PATHWAY TIMEOUT VALUE			
37	Reserved							
39								
40	(MSB) CRC							
43								(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 91h.

The PHY IDENTIFIER field specifies the phy (see 4.2.7) to which the PHY CONTROL request applies.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 161 defines the PHY OPERATION field.

Table 161 — Phy operation

Code	Operation	Description
00h	NOP	No operation.
01h	LINK RESET	If the specified phy is not a virtual phy, perform a link reset sequence (see 4.4) on the specified phy and enable the specified phy. If the specified phy is a virtual phy, perform an internal reset and enable the specified phy. Any affiliation (see 7.17.3) shall continue to be present. The phy shall bypass the SATA spinup hold state. The SMP response shall be returned without waiting for the link reset to complete.
02h	HARD RESET	If the specified phy is not a virtual phy, perform a link reset sequence (see 4.4) on the specified phy and enable the specified phy. If the attached phy is not a SATA phy, the link reset sequence shall include a hard reset sequence (see 4.4.2). If the attached phy is a SATA phy, the phy shall bypass the SATA spinup hold state. If the specified phy is a virtual phy, perform an internal reset and enable the specified phy. Any affiliation (see 7.17.3) shall be cleared. The SMP response shall be returned without waiting for the hard reset to complete.
03h	DISABLE	Disable the specified phy (i.e., stop transmitting valid dwords and receiving dwords on the specified phy). The LINK RESET and HARD RESET operations may be used to enable the phy.
04h	Reserved	
05h	CLEAR ERROR LOG	Clear the error log counters (see 10.4.3.6) for the specified phy.
06h	CLEAR AFFILIATION	Clear an affiliation (see 7.17.3) from the STP initiator port with the same SAS address as the SMP initiator port that opened this SMP connection. If there is no such affiliation, the SMP target port shall return a function result of SMP FUNCTION FAILED in the response frame.
All others	Reserved	

If the PHY IDENTIFIER field specifies the phy which is being used for the SMP connection and a phy operation of LINK RESET, HARD RESET, or DISABLE is requested, the SMP target port shall not perform the requested operation and shall return a function result of SMP FUNCTION FAILED in the response frame.

An UPDATE PARTIAL PATHWAY TIMEOUT VALUE bit set to one specifies that the PARTIAL PATHWAY TIMEOUT VALUE field shall be honored. An UPDATE PARTIAL PATHWAY TIMEOUT VALUE bit set to zero specifies that the PARTIAL PATHWAY TIMEOUT VALUE field shall be ignored.

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field specifies the minimum physical link rate the phy shall support during a link reset sequence (see 4.4.1). Table 162 defines the values for this field.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The PROGRAMMED MAXIMUM PHYSICAL LINK RATE field specifies the maximum physical link rates the phy shall support during a link reset sequence (see 4.4.1).Table 162 defines the values for this field.

Table 162 — Programmed minimum and maximum physical link rate

Code	Description
0h	Do not change current value
8h	1,5 Gbps
9h	3,0 Gbps
All others	Reserved

The PARTIAL PATHWAY TIMEOUT VALUE field specifies the amount of time in microseconds the expander phy shall wait after receiving an Arbitrating (Blocked On Partial) confirmation from the ECM before requesting that the ECM resolve pathway blockage (see 7.12.4.4). A PARTIAL PATHWAY TIMEOUT VALUE field value of zero (i.e., 0 μs) specifies that partial pathway resolution shall be requested by the expander phy immediately upon reception of an Arbitrating (Blocked On Partial) confirmation from the ECM. The PARTIAL PATHWAY TIMEOUT VALUE field is only honored when the UPDATE PARTIAL PATHWAY TIMEOUT VALUE bit is set to one.

The CRC field is defined in 10.4.3.1.

Table 163 defines the response format.

Table 163 — PHY CONTROL response

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (91h)							
2	FUNCTION RESULT							
3	Reserved							
4	(MSB)			CRC				
7								(LSB)

The SMP FRAME TYPE field shall be set to 41h.

The FUNCTION field shall be set to 91h.

The FUNCTION RESULT field is defined in 10.4.3.2.

The CRC field is defined in 10.4.3.2.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex A
(normative)

Compliant jitter test pattern (CJTPAT)

The CJTPAT consists of a long run of low transition density pattern, followed by a long run of high transition density pattern, followed by another short run of low transition density pattern. The transitions between the pattern segments stress the receiver. The test pattern is designed to contain the phase shift in both polarities, from 0 to 1 and from 1 to 0. The critical pattern sections with the phase shifts are underlined.

Table A.1 shows the CJTPAT when there is positive running disparity (RD+) at the beginning of the pattern. The 8b and 10b values of each character are shown. If the same 8b characters are used when there is negative running disparity (RD-), the resulting 10b pattern is different and does not provide the critical phase shifts. To achieve the same phase shift effects with RD-, a different 8b pattern is required.

Table A.1 — CJTPAT for RD+

Running disparity	First character	Second character	Third character	Fourth character	Running disparity
RD+	D30.3 (7Eh)	D30.3 (7Eh)	D30.3 (7Eh)	D30.3 (7Eh)	RD+
	1000011100b	0111100011b	1000011100b	0111100011b	
The above dword of low density pattern is sent a total of 41 times					
RD+	D30.3 (7Eh)	D30.3 (7Eh)	D30.3 (7Eh)	D20.3 (74h)	RD-
	1000011100b	0111100011b	1000011100b	0010111100b	
Phase shift 11100001011					
RD-	D30.3 (7Eh)	D11.5 (ABh)	D21.5 (B5h)	D21.5 (B5h)	RD+
	0111100011b	1101001010b	1010101010b	1010101010b	
Phase shift 00011110100					
RD+	D21.5 (B5h)	D21.5 (B5h)	D21.5 (B5h)	D21.5 (B5h)	RD+
	1010101010b	1010101010b	1010101010b	1010101010b	
The above dword of high density pattern is sent a total of 12 times					
RD+	D21.5 (B5h)	D30.2 (5Eh)	D10.2 (4Ah)	D30.3 (7Eh)	RD+
	1010101010b	1000010101b	0101010101b	0111100011b	
Phase shift 01010000b and 10101111b					

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table A.2 shows the CJTPAT when there is negative running disparity (RD-) at the beginning of the pattern. The 8b and 10b values of each character are shown.

Table A.2 — CJTPAT for RD-

Running disparity	First character	Second character	Third character	Fourth character	Running disparity
RD-	D30.3 (7Eh)	D30.3 (7Eh)	D30.3 (7Eh)	D30.3 (7Eh)	RD-
	0111100011b	1000011100b	0111100011b	1000011100b	
The above dword of low density pattern is sent a total of 41 times					
RD-	D30.3 (7Eh)	D30.3 (7Eh)	D30.3 (7Eh)	D11.3 (6Bh)	RD+
	0111100011b	1000011100b	0111100011b	1101000011b	
Phase shift 00011110100b					
RD+	D30.3 (7Eh)	D20.2 (54h)	D10.2 (4Ah)	D10.2 (4Ah)	RD-
	1000011100b	0010110101b	0101010101b	0101010101b	
Phase shift 11100001011b					
RD-	D10.2 (4Ah)	D10.2 (4Ah)	D10.2 (4Ah)	D10.2 (4Ah)	RD-
	0101010101b	0101010101b	0101010101b	0101010101b	
The above dword of high density pattern is sent a total of 12 times					
RD-	D10.2 (4Ah)	D30.5 (BEh)	D21.5 (B5h)	D30.3 (7Eh)	RD-
	0101010101b	011110101 0b	10101010b	1000011100b	
Phase shift 10101111b and 01010000b					

To use CJTPAT as the payload in an SSP DATA frame, the 8b patterns for both RD+ and RD- shall be included.

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Table A.3 shows a pattern containing both CJTPAT for RD+ and CJTPAT for RD-. The 10b pattern resulting from encoding the 8b pattern contains the desired bit sequences for the phase shifts with both starting running disparities.

Table A.3 — CJTPAT for RD+ and RD-

First character	Second character	Third character	Fourth character	Notes
D30.3(7Eh)	D30.3(7Eh)	D30.3(7Eh)	D30.3(7Eh)	This dword is sent a total of 41 times.
...	...	...	...	
D30.3(7Eh)	D30.3(7Eh)	D30.3(7Eh)	D20.3(74h)	This dword is sent once.
D30.3(7Eh)	D11.5(ABh)	D21.5(B5h)	D21.5(B5h)	This dword is sent once.
D21.5(B5h)	D21.5(B5h)	D21.5(B5h)	D21.5(B5h)	This dword is sent a total of 12 times.
...	...	...	...	
D21.5(B5h)	D30.2(5Eh)	D10.2(4Ah)	D30.3(7Eh)	This dword is sent once.
D30.3(7Eh)	D30.3(7Eh)	D30.3(7Eh)	D30.3(7Eh)	This dword is sent a total of 41 times.
...	...	...	...	
D30.3(7Eh)	D30.3(7Eh)	D30.3(7Eh)	D11.3(6Bh)	This dword is sent once.
D30.3(7Eh)	D20.2(54h)	D10.2(4Ah)	D10.2(4Ah)	This dword is sent once.
D10.2(4Ah)	D10.2(4Ah)	D10.2(4Ah)	D10.2.(4Ah)	This dword is sent a total of 12 times.
...	...	...	...	
D10.2(4Ah)	D30.5(BEh)	D21.5(B5h)	D30.3(7Eh)	This dword is sent once.

When the CJTPAT is encapsulated in an SSP frame, the scrambler needs to be considered. By scrambling the desired 8b pattern prior to submitting it to the transmitter scrambler, the scrambling in the transmitter scrambler reverses the prior scrambling of the 8b pattern and the desired 10b pattern results. The 8b data dwords are scrambled by XORing the pattern with the expected scrambler dword output, taking into account the position of the 8b data dwords within the protocol frame.

Table A.4 shows CJTPAT embedded in a SSP DATA frame with a 24-byte header following the SOF.

The second column (8b data dword) in table A.4 lists the desired 8b pattern data that is to be 8b10b encoded.

The third column (Scrambler output dword) in table A.4 lists the output, in dword format, of the transmitter scrambler.

The fourth column (Scrambled 8b data dword) in table A.4 shows the result of XORing the 8b data with the scrambler output. The data in this column, if supplied to the transmitter scrambler, results in the desired 10b test pattern on the physical link.

The scrambler is re-initialized at the beginning of each frame (SOF) and the scrambler output is independent of the scrambled data. The insertion of ALIGNs within the frame should be avoided because of the possible disruption of the pattern on the physical link.

--`,,,`,,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table A.4 — CJTPAT scrambled in an SSP DATA frame (Sheet 1 of 4)

Frame contents	8b data dword	Scrambler output dword	Scrambled 8b data dword = 8b data dword XOR scrambler output dword
SOF	<primitive>	n/a	n/a
SSP DATA frame header	unknown	C2D2768Dh	unknown
	unknown	1F26B368h	unknown
	unknown	A508436Ch	unknown
	unknown	3452D354h	unknown
	unknown	8A559502h	unknown
	unknown	BB1ABE1Bh	unknown
SSP frame data	7E7E7E7Eh	FA56B73Dh	8428C943h
	7E7E7E7Eh	53F60B1Bh	2D887565h
	7E7E7E7Eh	F0809C41h	8EFEE23Fh
	7E7E7E7Eh	747FC34Ah	0A01BD34h
	7E7E7E7Eh	BE865291h	C0F82CEFh
	7E7E7E7Eh	7A6FA7B6h	0411D9C8h
	7E7E7E7Eh	3163E6D6h	4F1D98A8h
	7E7E7E7Eh	F036FE0Ch	8E488072h
SSP frame data	7E7E7E7Eh	1EF3EA29h	608D9457h
	7E7E7E7Eh	EB342694h	954A58EAh
	7E7E7E7Eh	53853B17h	2DFB4569h
	7E7E7E7Eh	E94ADC4Dh	9734A233h
	7E7E7E7Eh	5D200E88h	235E70F6h
	7E7E7E7Eh	6901EDD0h	177F93AEh
	7E7E7E7Eh	FA9E38DEh	84E046A0h
	7E7E7E7Eh	68DB4B07h	16A53579h
SSP frame data	7E7E7E7Eh	450A437Bh	3B743D05h
	7E7E7E7Eh	960DD708h	E873A976h
	7E7E7E7Eh	3F35E698h	414B98E6h
	7E7E7E7Eh	FE7698A5h	8008E6DBh
	7E7E7E7Eh	C80EF715h	B670896Bh
	7E7E7E7Eh	666090AFh	181EEED1h
	7E7E7E7Eh	FAF0D5CBh	848EABB5h
	7E7E7E7Eh	2B82009Fh	55FC7EE1h

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table A.4 — CJTPAT scrambled in an SSP DATA frame (Sheet 2 of 4)

Frame contents	8b data dword	Scrambler output dword	Scrambled 8b data dword = 8b data dword XOR scrambler output dword
SSP frame data	7E7E7E7Eh	0E317491h	704F0AEFh
	7E7E7E7Eh	76F46A1Eh	088A1460h
	7E7E7E7Eh	F46D6948h	8A131736h
	7E7E7E7Eh	7BCD8A93h	05B3F4Edh
	7E7E7E7Eh	1513AD7Eh	6B6DD300h
	7E7E7E7Eh	1E72FEEEh	600C8090h
	7E7E7E7Eh	A014AA3Bh	DE6AD445h
	7E7E7E7Eh	23AAD4E7h	5DD4AA99h
SSP frame data	7E7E7E7Eh	B0DC9E67h	CEA2E019h
	7E7E7E7Eh	E0A573FBh	9EDB0D85h
	7E7E7E7Eh	06CA944Fh	78B4EA31h
	7E7E7E7Eh	63E29212h	1D9CEC6Ch
	7E7E7E7Eh	4578626Dh	3B061C13h
	7E7E7E7Eh	53260C93h	2D5872EDh
	7E7E7E7Eh	3E592202h	40275C7Ch
	7E7E7E7Eh	2B6ECA63h	5510B41Dh
SSP frame data	7E7E7E7Eh	636A1F1Fh	1D146161h
	7E7E7E74h	35B5A9EDh	4BCBD799h
	7EABB5B5h	4AA2A0FDh	34091548h
	B5B5B5B5h	71AFE196h	C41A5423h
	B5B5B5B5h	E1D57B62h	5460CED7h
	B5B5B5B5h	55A0568Ah	E015E33Fh
	B5B5B5B5h	82D18968h	37643CDDh
	B5B5B5B5h	234CB4FFh	96F9014Ah
SSP frame data	B5B5B5B5h	83481E7Fh	36FDABCAh
	B5B5B5B5h	B21AE87Fh	07AF5DCAh
	B5B5B5B5h	A9C5EACDh	1C705F78h
	B5B5B5B5h	6201ACC3h	D7B41976h
	B5B5B5B5h	F60939CEh	43BC8C7Bh
	B5B5B5B5h	395F767Dh	8CEAC3C8h
	B5B5B5B5h	2FA55841h	9A10EDF4h
	B55E4A7Eh	836D4A7Ah	36330004h

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table A.4 — CJTPAT scrambled in an SSP DATA frame (Sheet 3 of 4)

Frame contents	8b data dword	Scrambler output dword	Scrambled 8b data dword = 8b data dword XOR scrambler output dword
SSP frame data	7E7E7E7Eh	388D587Ah	46F32604h
	7E7E7E7Eh	773DFF5Ch	09438122h
	7E7E7E7Eh	3C239CB3h	425DE2CDh
	7E7E7E7Eh	564D91A0h	2833EFDEh
	7E7E7E7Eh	43ED0BE1h	3D93759Fh
	7E7E7E7Eh	987429A7h	E60A57D9h
	7E7E7E7Eh	E52DDBA2h	9B53A5DCh
	7E7E7E7Eh	E78DC87Fh	99F3B601h
SSP frame data	7E7E7E7Eh	0AB8C669h	74C6B817h
	7E7E7E7Eh	64D083C9h	1AAEFDB7h
	7E7E7E7Eh	053DF93Ah	7B438744h
	7E7E7E7Eh	EEE9D9Eah	9097A794h
	7E7E7E7Eh	44BD3B97h	3AC345E9h
	7E7E7E7Eh	0FE24B8Ch	719C35F2h
	7E7E7E7Eh	F28D5694h	8CF328EAh
	7E7E7E7Eh	6310B6D9h	1D6EC8A7h
SSP frame data	7E7E7E7Eh	1792AECEh	69ECD0B0h
	7E7E7E7Eh	0A562EA1h	742850DFh
	7E7E7E7Eh	B048DF69h	CE36A117h
	7E7E7E7Eh	161A2878h	68645606h
	7E7E7E7Eh	5519CB51h	2B67B52Fh
	7E7E7E7Eh	19F5BE56h	678BC028h
	7E7E7E7Eh	EFFFB4B6h	9181CAC8h
	7E7E7E7Eh	B3826E72h	CDFC100Ch
SSP frame data	7E7E7E7Eh	E4722DDAh	9A0C53A4h
	7E7E7E7Eh	60BF5129h	1EC12F57h
	7E7E7E7Eh	248D90F5h	5AF3EE8Bh
	7E7E7E7Eh	4D06D21Ch	3378AC62h
	7E7E7E7Eh	7E96166Ch	00E86812h
	7E7E7E7Eh	5FAFE3B4h	21D19DCAh
	7E7E7E7Eh	506CB855h	2E12C62Bh
	7E7E7E7Eh	5BF03098h	258E4EE6h

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table A.4 — CJTPAT scrambled in an SSP DATA frame (Sheet 4 of 4)

Frame contents	8b data dword	Scrambler output dword	Scrambled 8b data dword = 8b data dword XOR scrambler output dword
SSP frame data	7E7E7E7Eh	46D4B6B3h	38AAC8CDh
	7E7E7E7Eh	051B9E11h	7B65E06Fh
	7E7E7E7Eh	015CC556h	7F22BB28h
	7E7E7E7Eh	E21035EFh	9C6E4B91h
	7E7E7E7Eh	56604D75h	281E330Bh
	7E7E7E7Eh	2E76675Ch	50081922h
	7E7E7E7Eh	071476F0h	796A088Eh
	7E7E7E7Eh	AFF087EBh	D18EF995h
SSP frame data	7E7E7E7Eh	1B62DB01h	651CA57Fh
	7E7E7E6Bh	23661F6Ch	5D186107h
	7E544A4Ah	F877B027h	8623FA6Dh
	4A4A4A4Ah	F5E389A2h	BFA9C3E8h
	4A4A4A4Ah	EEC73611h	A48D7C5Bh
	4A4A4A4Ah	4C04FB93h	064EB1D9h
	4A4A4A4Ah	E8D70F32h	A29D4578h
	4A4A4A4Ah	BFF03C54h	F5BA761Eh
SSP frame data	4A4A4A4Ah	E3403C01h	A90A764Bh
	4A4A4A4Ah	20FACA7Eh	6AB08034h
	4A4A4A4Ah	9942458Ch	D3080FC6h
	4A4A4A4Ah	37E2CB89h	7DA881C3h
	4A4A4A4Ah	5A1A9783h	1050DDC9h
	4A4A4A4Ah	CE48AA3Fh	8402E075h
	4A4A4A4Ah	06C9A761h	4C83ED2Bh
	4ABEB57Eh	06C03EABh	4C7E8BD5h
CRC	unknown	don’t care	unknown
EOF	<primitive>	N/A	N/A

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex B
(informative)

SAS to SAS phy reset sequence examples

Figure B.1 shows a speed negotiation between a phy A that supports only the G1 physical link rate attached to a phy B that only supports the G1 physical link rate. Both phys run:

1) the G1 speed negotiation window, supported by both phys; and
2) the G2 speed negotiation window, supported by neither phy.

Both phys then select G1 for the final speed negotiation window used to establish the negotiated physical link rate.

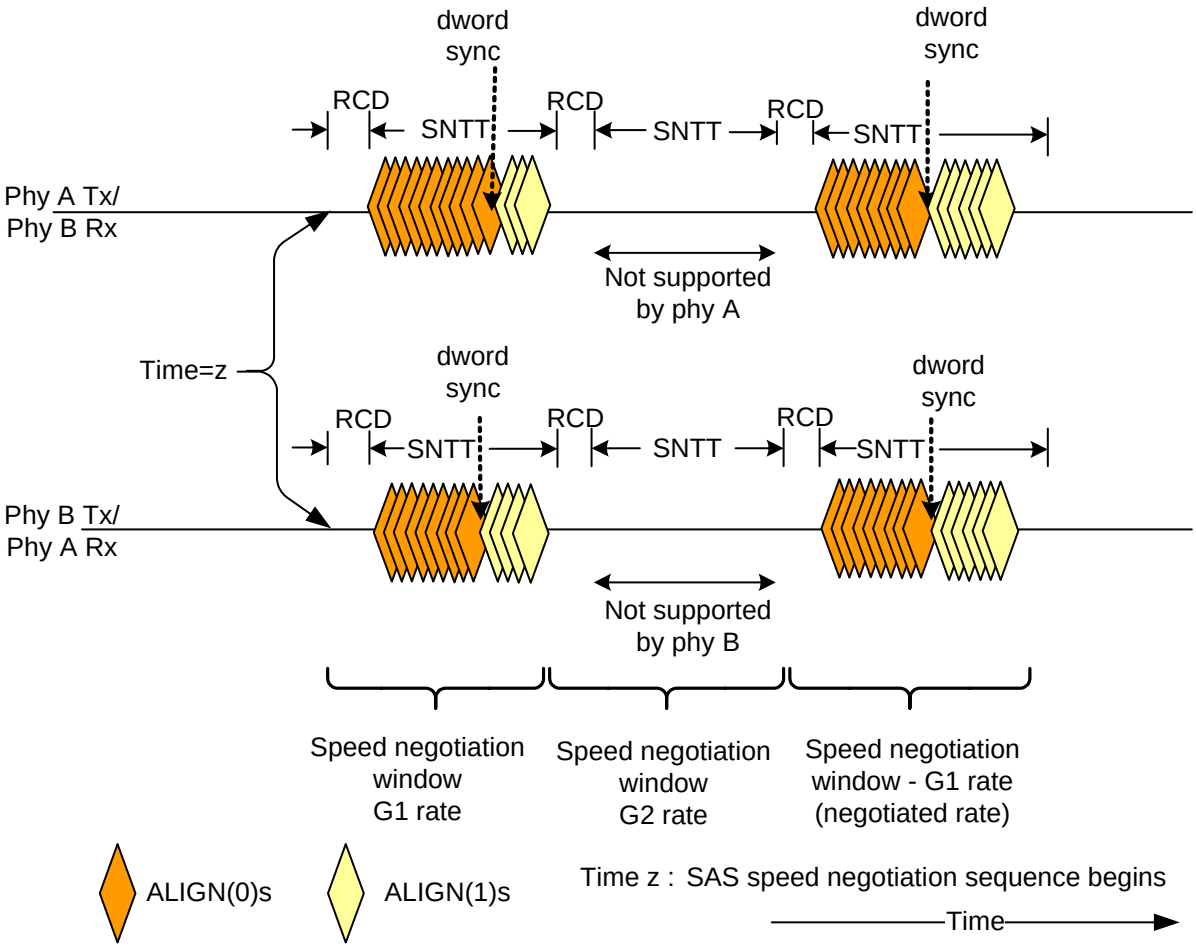


Figure B.1 — SAS speed negotiation sequence (phy A: G1 only, phy B: G1 only)

Figure B.2 shows a speed negotiation between a phy A that supports G1 through G3 physical link rates attached to a phy B that only supports G1 and G2 physical link rates. Both phys run:

1) the G1 speed negotiation window, supported by both phys;
2) the G2 speed negotiation window, supported by both phys; and
3) the G3 speed negotiation window, supported by phy A but not by phy B.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Both phys then select G2 for the final speed negotiation window used to establish the negotiated physical link rate.

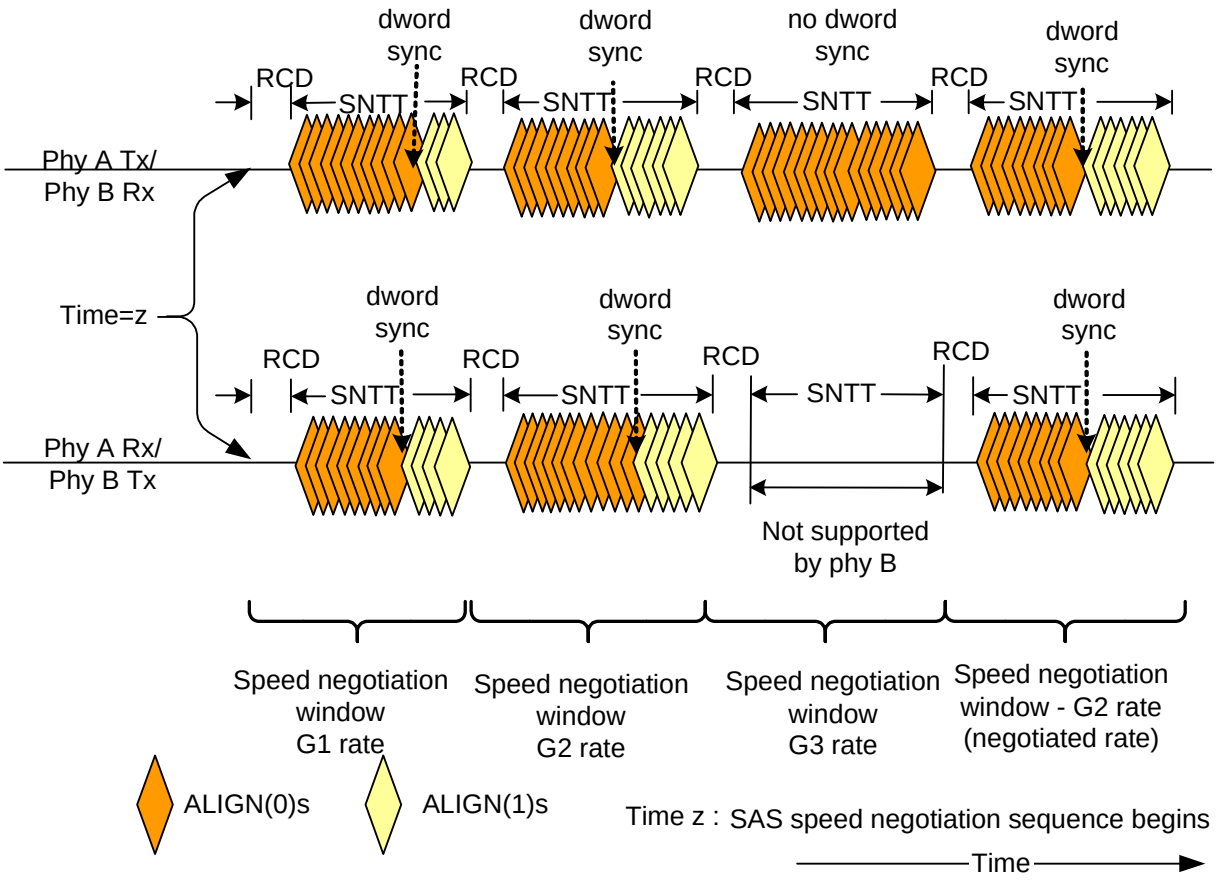


Figure B.2 — SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G1, G2)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex C
(informative)

CRC

C.1 CRC generator and checker implementation examples

Figure C.1 shows an example of a CRC generator.

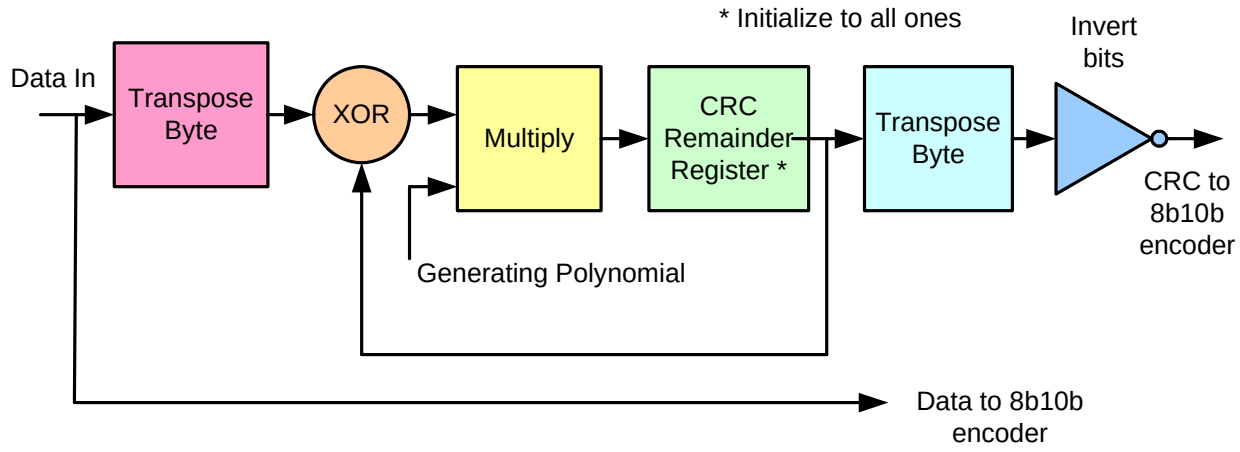


Figure C.1 — CRC generator example

Figure C.2 shows an example of a CRC checker.

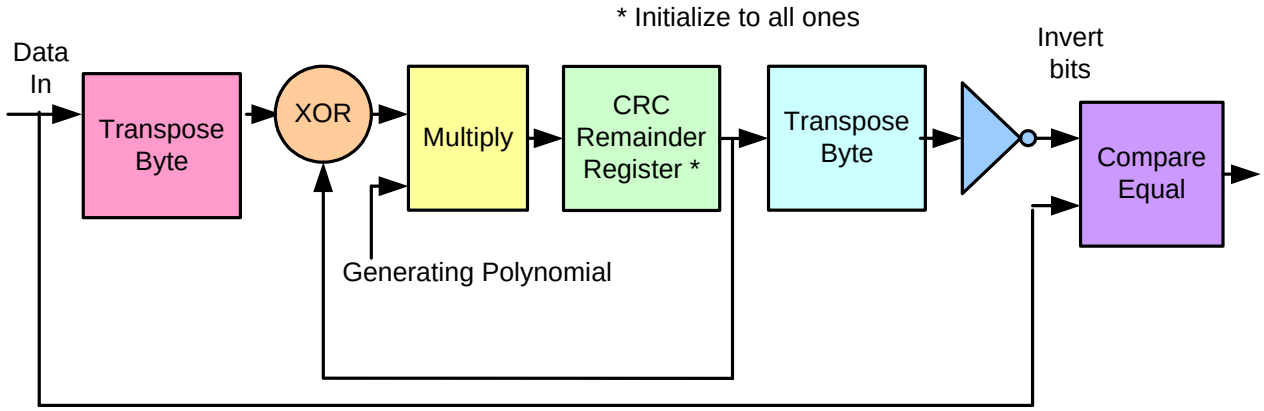


Figure C.2 — CRC checker example

C.2 CRC implementation in C

The following is an example C program that generates the value for the CRC field in frames. The inputs are the data dwords for the frame and the number of data dwords.

```
#include <stdio.h>

void main (void) {

      static unsigned long data_dwords[] = {
                  0x06D0B992L, 0x00B5DF59L, 0x00000000L,
                  0x00000000L, 0x1234FFFFL, 0x00000000L,
                  0x00000000L, 0x00000000L, 0x00000000L,
                  0x08000012L, 0x01000000L, 0x00000000L,
                  0x00000000L};  /* example data dwords */
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
      unsigned long calculate_crc(unsigned long *, unsigned long);
      unsigned long crc;

      crc = calculate_crc(data_dwords, 13);
      printf ("Example CRC is %x\n", crc);
}

/* returns crc value */
unsigned long calculate_crc(unsigned long *frame, unsigned long length) {
long poly = 0x04C11DB7L;
unsigned long crc_gen, x;
union {
      unsigned long lword;
      unsigned char byte[4];
} b_access;
static unsigned char xpose[] = {
                  0x0, 0x8, 0x4, 0xC, 0x2, 0xA, 0x6, 0xE,
                  0x1, 0x9, 0x5, 0xD, 0x3, 0xB, 0x7, 0xF};
unsigned int i, j, fb;

crc_gen = ~0;   /* seed generator with all ones */
for (i = 0; i < length; i++) {
      x = *frame++; /* get word */
        b_access.lword = x; /* transpose bits in byte */
        for (j = 0; j < 4; j++) {
            b_access.byte[j] = xpose[b_access.byte[j] >> 4] |
                        xpose[b_access.byte[j] & 0xF] << 4;
        }
        x = b_access.lword;

        for (j = 0; j < 32; j++) { /* serial shift register implementation */
            fb = ((x & 0x80000000L) > 0) ^ ((crc_gen & 0x80000000L) > 0);
            x <<= 1;
            crc_gen <<= 1;
            if (fb)
                  crc_gen ^= poly;
      }
}

b_access.lword = crc_gen; /* transpose bits in CRC */
for (j = 0; j < 4; j++) {
      b_access.byte[j] = xpose[b_access.byte[j] >> 4] |
                  xpose[b_access.byte[j] & 0xF] << 4;
}
crc_gen = b_access.lword;

return ~crc_gen; /* invert output */
}
```

C.3 CRC implementation with XORs

These equations implement the multiply function shown in figure C.1 and figure C.2. The ^ symbol represents an XOR operation.

```
crc00 = d00^d06^d09^d10^d12^d16^d24^d25^d26^d28^d29^d30^d31;
crc01 = d00^d01^d06^d07^d09^d11^d12^d13^d16^d17^d24^d27^d28;
crc02 = d00^d01^d02^d06^d07^d08^d09^d13^d14^d16^d17^d18^d24^d26^d30^d31;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
crc03 = d01^d02^d03^d07^d08^d09^d10^d14^d15^d17^d18^d19^d25^d27^d31;
crc04 = d00^d02^d03^d04^d06^d08^d11^d12^d15^d18^d19^d20^d24^d25^d29^d30^d31;
crc05 = d00^d01^d03^d04^d05^d06^d07^d10^d13^d19^d20^d21^d24^d28^d29;
crc06 = d01^d02^d04^d05^d06^d07^d08^d11^d14^d20^d21^d22^d25^d29^d30;
crc07 = d00^d02^d03^d05^d07^d08^d10^d15^d16^d21^d22^d23^d24^d25^d28^d29;
crc08 = d00^d01^d03^d04^d08^d10^d11^d12^d17^d22^d23^d28^d31;
crc09 = d01^d02^d04^d05^d09^d11^d12^d13^d18^d23^d24^d29;
crc10 = d00^d02^d03^d05^d09^d13^d14^d16^d19^d26^d28^d29^d31;
crc11 = d00^d01^d03^d04^d09^d12^d14^d15^d16^d17^d20^d24^d25^d26^d27^d28^d31;
crc12 = d00^d01^d02^d04^d05^d06^d09^d12^d13^d15^d17^d18^d21^d24^d27^d30^d31;
crc13 = d01^d02^d03^d05^d06^d07^d10^d13^d14^d16^d18^d19^d22^d25^d28^d31;
crc14 = d02^d03^d04^d06^d07^d08^d11^d14^d15^d17^d19^d20^d23^d26^d29;
crc15 = d03^d04^d05^d07^d08^d09^d12^d15^d16^d18^d20^d21^d24^d27^d30;
crc16 = d00^d04^d05^d08^d12^d13^d17^d19^d21^d22^d24^d26^d29^d30;
crc17 = d01^d05^d06^d09^d13^d14^d18^d20^d22^d23^d25^d27^d30^d31;
crc18 = d02^d06^d07^d10^d14^d15^d19^d21^d23^d24^d26^d28^d31;
crc19 = d03^d07^d08^d11^d15^d16^d20^d22^d24^d25^d27^d29;
crc20 = d04^d08^d09^d12^d16^d17^d21^d23^d25^d26^d28^d30;
crc21 = d05^d09^d10^d13^d17^d18^d22^d24^d26^d27^d29^d31;
crc22 = d00^d09^d11^d12^d14^d16^d18^d19^d23^d24^d26^d27^d29^d31;
crc23 = d00^d01^d06^d09^d13^d15^d16^d17^d19^d20^d26^d27^d29^d31;
crc24 = d01^d02^d07^d10^d14^d16^d17^d18^d20^d21^d27^d28^d30;
crc25 = d02^d03^d08^d11^d15^d17^d18^d19^d21^d22^d28^d29^d31;
crc26 = d00^d03^d04^d06^d10^d18^d19^d20^d22^d23^d24^d25^d26^d28^d31;
crc27 = d01^d04^d05^d07^d11^d19^d20^d21^d23^d24^d25^d26^d27^d29;
crc28 = d02^d05^d06^d08^d12^d20^d21^d22^d24^d25^d26^d27^d28^d30;
crc29 = d03^d06^d07^d09^d13^d21^d22^d23^d25^d26^d27^d28^d29^d31;
crc30 = d04^d07^d08^d10^d14^d22^d23^d24^d26^d27^d28^d29^d30;
crc31 = d05^d08^d09^d11^d15^d23^d24^d25^d27^d28^d29^d30^d31;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.4 CRC examples

Table C.1 shows several CRC examples. Data is shown in dwords, from first to last.

Table C.1 — CRC examples

Data	CRC
00010203h 04050607h 08090A0Bh 0C0D0E0Fh 10111213h 14151617h 18191A1Bh 1C1D1E1Fh	8A7E2691h
00000001h 00000000h 00000000h 00000000h 00000000h 00000000h 00000000h 00000000h	898C0D7Ah
00000000h 00000000h 00000000h 00000000h 00000000h 00000000h 00000000h 00000001h	3B650D6Eh
06D0B992h 00B5DF59h 00000000h 00000000h 1234FFFFh 00000000h 00000000h 00000000h 00000000h 08000012h 01000000h 00000000h 00000000h	3F4F1C26h

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex D
(informative)

SAS address hashing

D.1 SAS address hashing overview

See 4.2.2 for a description of hashed SAS addresses and the algorithm used to create them.

D.2 Hash collision probability

The following are Monte-Carlo simulations evaluating the probability of collision in a system containing 128 addressable SAS ports. Four models were used for the models for the simulations:

a) random model;
b) sequential mode;
c) lots model; and
d) three lots model.

The random model uses a system with 128 randomly chosen 64-bit integers as SAS addresses.

The sequential model uses a system with 128 sequentially-assigned SAS addresses starting from a random 64-bit base.

The lots model uses:

a) Two sequentially assigned SAS addresses with unique company IDs and random vendor-specific identifiers;
b) 125 randomly drawn SAS addresses from a 10 000-unit production lot. The vendor-specific identifiers within the lot were assigned by 10 SAS address-writers, randomly drawn from a pool of 4 096 possible SAS address-writers. Each SAS address-writer assigns vendor-specific identifiers sequentially within its own subset of the vendor-specific identifiers, starting from a randomly chosen base at the beginning of the production run; and
c) One randomly chosen SAS address (representing a replacement unit) with another unique company ID.

The three lots model uses:

a) Two sequentially assigned SAS addresses with unique company IDs and random vendor-specific identifiers;
b) 125 randomly drawn SAS addresses from three 10 000-unit lots. The vendor-specific identifiers within each lot were assigned by 10 SAS address-writers, randomly drawn from a pool of 4 096 possible SAS address-writers for that vendor. Each SAS address-writer assigns vendor-specific identifiers sequentially within its own subset of the vendor-specific identifiers, starting from a randomly chosen base at the beginning of the production run. Each of the three lots has a different company ID; and
c) One randomly chosen SAS address (representing a replacement unit) with another unique company ID.

Table D.1 lists the results of Monte-Carlo simulation.

Table D.1 — Monte-Carlo simulation results

SAS address model	Trials	Collisions	Average collisions per system
lots	2 000 000 000	45 063	0,000 022 531 5
three lots	2 000 000 000	662 503	0,000 331 251 5
random	10 000 000	4 882	0,000 488 2
sequential	10 000 000	0	0

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

D.3 Hash generation

One way to implement the hashing encoder in hardware is to use serial shift registers as shown in figure D.1. For error correction purposes, the number of data bits is limited to 39. For hashing purposes, the circuit shown serves as a divider. Because the period of this generator polynomial is 63, any binary sequence of length exceeding 63 is treated as a 63-bit sequence with (bit 63) × L + k added to (bit k modulo 2) for k = 0, 1, …, 62 and any integer L. Therefore, using this generator polynomial to hash a 64-bit address is equivalent to hashing a 63-bit sequence with bit 63 added modulo 2 to bit 0. With this wrapping, a binary sequence of any length is treated as an equivalent binary sequence of 63 bits, which, in turn, is treated as a degree-62 polynomial. After feeding this equivalent degree-62 polynomial into the circuit shown, the shift register contains the remainder from dividing the degree-62 input polynomial by the generator polynomial. This remainder is the hashed result.

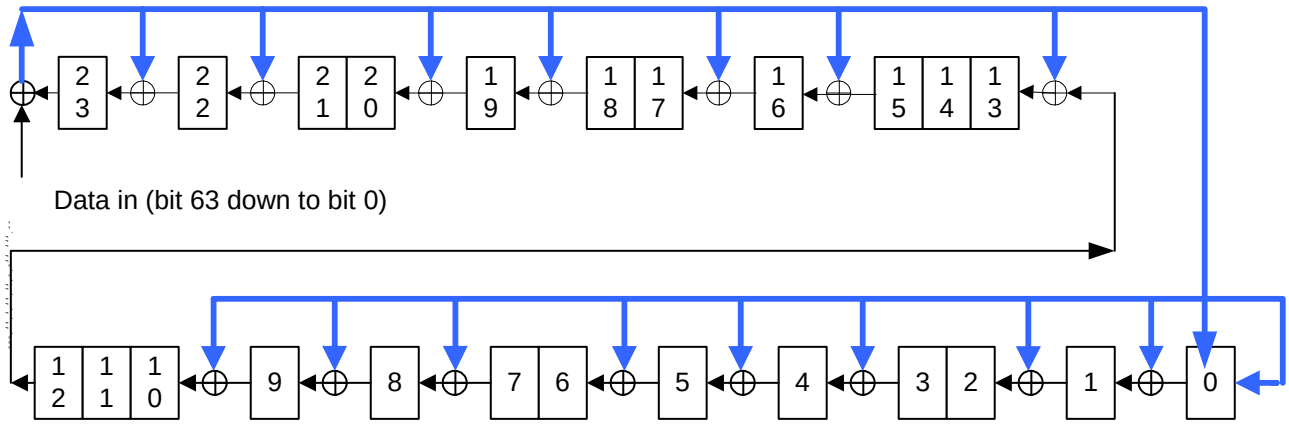


Figure D.1 — BCH(69, 39, 9) code generator

D.4 Hash implementation in C

The following is an example C program that generates a 24-bit hashed value from a 64-bit value.

```
typedef unsigned int uint32_t;
uint32_t hash(uint32_t upperbits, uint32_t lowerbits)
{
      const unsigned distance_9_poly = 0x01DB2777;
      uint32_t msb = 0x01000000;
      uint32_t moving_one, leading_bit;
      int i;
      unsigned regg;
      regg = 0;
      moving_one = 0x80000000;
      for (i = 31; i >= 0; i--) {
            leading_bit = 0;
            if (moving_one & upperbits) leading_bit = msb;
            regg <<= 1;
            regg ^= leading_bit;
            if (regg & msb) regg ^= distance_9_poly;
            moving_one >>= 1;
      }
      moving_one = 0x80000000;
      for (i = 31; i >= 0; i--) { // note lower limit of i = 0;
            leading_bit = 0;
            if (moving_one & lowerbits) leading_bit = msb;
            regg <<= 1;
            regg ^= leading_bit;
            if (regg & msb) regg ^= distance_9_poly;
            moving_one >>= 1;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        }
        return regg & 0x00FFFFFF;
    }
```

D.5 Hash implementation with XORs

These equations generate the 24-bit hashed SAS address for the SSP frame header from a 64-bit SAS address. The ^ symbol represents an XOR.

```
hash00=d00^d01^d03^d05^d07^d09^d10^d11^d12^d15^d16^d17^d18^d19^d20^d21^d22^
      d23^d24^d25^d28^d30^d31^d33^d34^d36^d38^d39^d63;
hash01=d00^d02^d03^d04^d05^d06^d07^d08^d09^d13^d15^d26^d28^d29^d30^d32^d33^
      d35^d36^d37^d38^d40^d63;
hash02=d00^d04^d06^d08^d11^d12^d14^d15^d17^d18^d19^d20^d21^d22^d23^d24^d25^
      d27^d28^d29^d37^d41^d63;
hash03=d01^d05^d07^d09^d12^d13^d15^d16^d18^d19^d20^d21^d22^d23^d24^d25^d26^
      d28^d29^d30^d38^d42;
hash04=d00^d01^d02^d03^d05^d06^d07^d08^d09^d11^d12^d13^d14^d15^d18^d26^d27^
      d28^d29^d33^d34^d36^d38^d43^d63;
hash05=d00^d02^d04^d05^d06^d08^d11^d13^d14^d17^d18^d20^d21^d22^d23^d24^d25^
      d27^d29^d31^d33^d35^d36^d37^d38^d44^d63;
hash06=d00^d06^d10^d11^d14^d16^d17^d20^d26^d31^d32^d33^d37^d45^d63;
hash07=d01^d07^d11^d12^d15^d17^d18^d21^d27^d32^d33^d34^d38^d46;
hash08=d00^d01^d02^d03^d05^d07^d08^d09^d10^d11^d13^d15^d17^d20^d21^d23^d24^
      d25^d30^d31^d35^d36^d38^d47^d63;
hash09=d00^d02^d04^d05^d06^d07^d08^d14^d15^d17^d19^d20^d23^d26^d28^d30^d32^
      d33^d34^d37^d38^d48^d63;
hash10=d00^d06^d08^d10^d11^d12^d17^d19^d22^d23^d25^d27^d28^d29^d30^d35^d36^
      d49^d63;
hash11=d01^d07^d09^d11^d12^d13^d18^d20^d23^d24^d26^d28^d29^d30^d31^d36^d37^
      d50;
hash12=d02^d08^d10^d12^d13^d14^d19^d21^d24^d25^d27^d29^d30^d31^d32^d37^d38^
      d51;
hash13=d00^d01^d05^d07^d10^d12^d13^d14^d16^d17^d18^d19^d21^d23^d24^d26^d32^
      d34^d36^d52^d63;
hash14=d01^d02^d06^d08^d11^d13^d14^d15^d17^d18^d19^d20^d22^d24^d25^d27^d33^
      d35^d37^d53;
hash15=d02^d03^d07^d09^d12^d14^d15^d16^d18^d19^d20^d21^d23^d25^d26^d28^d34^
      d36^d38^d54;
hash16=d00^d01^d04^d05^d07^d08^d09^d11^d12^d13^d18^d23^d25^d26^d27^d28^d29^
      d30^d31^d33^d34^d35^d36^d37^d38^d55^d63;
hash17=d00^d02^d03^d06^d07^d08^d11^d13^d14^d15^d16^d17^d18^d20^d21^d22^d23^
      d25^d26^d27^d29^d32^d33^d35^d37^d56^d63;
hash18=d01^d03^d04^d07^d08^d09^d12^d14^d15^d16^d17^d18^d19^d21^d22^d23^d24^
      d26^d27^d28^d30^d33^d34^d36^d38^d57;
hash19=d00^d01^d02^d03^d04^d07^d08^d11^d12^d13^d21^d27^d29^d30^d33^d35^d36^
      d37^d38^d58^d63;
hash20=d00^d02^d04^d07^d08^d10^d11^d13^d14^d15^d16^d17^d18^d19^d20^d21^d23^
      d24^d25^d33^d37^d59^d63;
hash21=d01^d03^d05^d08^d09^d11^d12^d14^d15^d16^d17^d18^d19^d20^d21^d22^d24^
      d25^d26^d34^d38^d60;
hash22=d00^d01^d02^d03^d04^d05^d06^d07^d11^d13^d24^d26^d27^d28^d30^d31^d33^
      d34^d35^d36^d38^d61^d63;
hash23=d00^d02^d04^d06^d08^d09^d10^d11^d14^d15^d16^d17^d18^d19^d20^d21^d22^
      d23^d24^d27^d29^d30^d32^d33^d35^d37^d38^d62^d63;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

D.6 Hash examples

Table D.2 shows examples using simple SAS addresses as input values. Two of the input values hash to the same value.

Table D.2 — Hash results for simple SAS addresses

64-bit input value	24-bit hashed value
00000000_00000000h	000000h
00000000_00000001h	DB2777h
FFFFFFFF_FFFFFFFFh	DB2777h

Table D.3 shows examples using realistic SAS addresses as input values.

Table D.3 — Hash results for realistic SAS addresses

64-bit input value	24-bit hashed value
50010753_4F0CFC88h	D0B992h
50010B92_B3CBF639h	B5DF59h
5002037E_157FEC63h	B064F7h
50004CF6_FBCE3889h	88FF12h
50020374_C4657EC7h	F36570h
50010D92_A016E450h	9F9571h
50002A58_850ACC66h	64B6B9h
50008C7B_EE7910DEh	8D6135h
500508BD_C22CAC94h	86ECF1h
500805F3_334B0AD3h	752AB2h
500A0B8A_FAA6A820h	5543A7h
500805E6_BCC55C68h	463DEDh

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table D.4 shows examples using a walking ones pattern to generate the input values.

Table D.4 — Hash results for a walking ones pattern

64-bit input value	24-bit hashed value	64-bit input value	24-bit hashed value
0000000000000001h	DB2777h	0000000100000000h	8232C2h
0000000000000002h	6D6999h	0000000200000000h	DF42F3h
0000000000000004h	DAD332h	0000000400000000h	65A291h
0000000000000008h	6E8113h	0000000800000000h	CB4522h
0000000000000010h	DD0226h	0000001000000000h	4DAD33h
0000000000000020h	61233Bh	0000002000000000h	9B5A66h
0000000000000040h	C24676h	0000004000000000h	ED93BBh
0000000000000080h	5FAB9Bh	0000008000000000h	000001h
0000000000000100h	BF5736h	0000010000000000h	000002h
0000000000000200h	A5891Bh	0000020000000000h	000004h
0000000000000400h	903541h	0000040000000000h	000008h
0000000000000800h	FB4DF5h	0000080000000000h	000010h
0000000000001000h	2DBC9Dh	0000100000000000h	000020h
0000000000002000h	5B793Ah	0000200000000000h	000040h
0000000000004000h	B6F274h	0000400000000000h	000080h
0000000000008000h	B6C39Fh	0000800000000000h	000100h
0000000000010000h	B6A049h	0001000000000000h	000200h
0000000000020000h	B667E5h	0002000000000000h	000400h
0000000000040000h	B7E8BDh	0004000000000000h	000800h
0000000000080000h	B4F60Dh	0008000000000000h	001000h
0000000000100000h	B2CB6Dh	0010000000000000h	002000h
0000000000200000h	BEB1ADh	0020000000000000h	004000h
0000000000400000h	A6442Dh	0040000000000000h	008000h
0000000000800000h	97AF2Dh	0080000000000000h	010000h
0000000001000000h	F4792Dh	0100000000000000h	020000h
0000000002000000h	33D52Dh	0200000000000000h	040000h
0000000004000000h	67AA5Ah	0400000000000000h	080000h
0000000008000000h	CF54B4h	0800000000000000h	100000h
0000000010000000h	458E1Fh	1000000000000000h	200000h
0000000020000000h	8B1C3Eh	2000000000000000h	400000h
0000000040000000h	CD1F0Bh	4000000000000000h	800000h
0000000080000000h	411961h	8000000000000000h	DB2777h

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table D.5 shows examples using a walking zeros pattern to generate the input values.

Table D.5 — Hash results for a walking zeros pattern

64-bit input value	24-bit hashed value	64-bit input value	24-bit hashed value
FFFFFFFFFFFFFFFEh	000000h	FFFFFFFEFFFFFFFFh	5915B5h
FFFFFFFFFFFFFFFDh	B64EEEh	FFFFFFFDFFFFFFFFh	046584h
FFFFFFFFFFFFFFFBh	01F445h	FFFFFFFBFFFFFFFFh	BE85E6h
FFFFFFFFFFFFFFF7h	B5A664h	FFFFFFF7FFFFFFFFh	106255h
FFFFFFFFFFFFFFEFh	062551h	FFFFFFEFFFFFFFFFh	968A44h
FFFFFFFFFFFFFFDFh	BA044Ch	FFFFFFDFFFFFFFFFh	407D11h
FFFFFFFFFFFFFFBFh	196101h	FFFFFFBFFFFFFFFFh	36B4CCh
FFFFFFFFFFFFFF7Fh	848CECh	FFFFFF7FFFFFFFFFh	DB2776h
FFFFFFFFFFFFEFFFh	647041h	FFFFFEFFFFFFFFFFh	DB2775h
FFFFFFFFFFFFDFFFh	7EAE6Ch	FFFFFDFFFFFFFFFFh	DB2773h
FFFFFFFFFFFFBFFFh	4B1236h	FFFFFBFFFFFFFFFFh	DB277Fh
FFFFFFFFFFFF7FFFh	206A82h	FFFFF7FFFFFFFFFFh	DB2767h
FFFFFFFFFFFEFFFFh	F69BEAh	FFFFEFFFFFFFFFFFh	DB2757h
FFFFFFFFFFFDFFFFh	805E4Dh	FFFFDFFFFFFFFFFFh	DB2737h
FFFFFFFFFFFBFFFFh	6DD503h	FFFFBFFFFFFFFFFFh	DB27F7h
FFFFFFFFFFF7FFFFh	6DE4E8h	FFFF7FFFFFFFFFFFh	DB2677h
FFFFFFFFFFEFFFFFh	6D873Eh	FFFEFFFFFFFFFFFFh	DB2577h
FFFFFFFFFFDFFFFFh	6D4092h	FFFDFFFFFFFFFFFFh	DB2377h
FFFFFFFFFFBFFFFFh	6CCFCAh	FFFBFFFFFFFFFFFFh	DB2F77h
FFFFFFFFFF7FFFFFh	6FD17Ah	FFF7FFFFFFFFFFFFh	DB3777h
FFFFFFFFFEFFFFFFh	69EC1Ah	FFEFFFFFFFFFFFFFh	DB0777h
FFFFFFFFFDFFFFFFh	6596DAh	FFDFFFFFFFFFFFFFh	DB6777h
FFFFFFFFFBFFFFFFh	7D635Ah	FFBFFFFFFFFFFFFFh	DBA777h
FFFFFFFFF7FFFFFFh	4C885Ah	FF7FFFFFFFFFFFFFh	DA2777h
FFFFFFFFEFFFFFFFh	2F5E5Ah	FEFFFFFFFFFFFFFFh	D92777h
FFFFFFFFDFFFFFFFh	E8F25Ah	FDFFFFFFFFFFFFFFh	DF2777h
FFFFFFFFBFFFFFFFh	BC8D2Dh	FBFFFFFFFFFFFFFFh	D32777h
FFFFFFFF7FFFFFFFh	1473C3h	F7FFFFFFFFFFFFFFh	CB2777h
FFFFFFFEFFFFFFFFh	9EA968h	EFFFFFFFFFFFFFFFh	FB2777h
FFFFFFFDFFFFFFFFh	503B49h	DFFFFFFFFFFFFFFFh	9B2777h
FFFFFFFBFFFFFFFFh	16387Ch	BFFFFFFFFFFFFFFFh	5B2777h
FFFFFFF7FFFFFFFFh	9A3E16h	7FFFFFFFFFFFFFFFh	000000h

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Annex E
(informative)

Scrambling

E.1 Scrambler implementation example

Figure E.1 shows an example of a data scrambler. This example generates the value to XOR with the dword input with two 16 bit parallel multipliers. 16 bits wide is the maximum width for the multiplier as the generating polynomial is 16 bits.

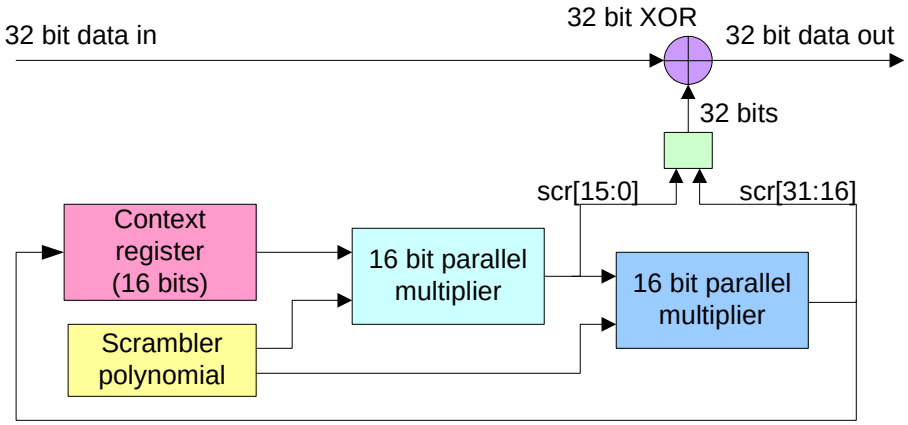


Figure E.1 — Scrambler

The generator polynomial is:

$$G(x) = x^{16} + x^{15} + x^{13} + x^4 + 1$$

For all implementations, the context register is initialized to produce a first dword output of C2D2768Dh for a dword input of all zeros.

E.2 Scrambler implementation in C

The following is an example C program that generates the scrambled data dwords for transmission. The inputs are the data dword to scramble and control indication to reinitialize the residual value (e.g., following an SOF).

```
#include <stdio.h>

unsigned long scramble(int reset, unsigned long dword);
void main(void)
{
int i;

for (i  = 0; i < 12; i++)
      printf(" %08X \n",scramble(i==0, 0));/* scramble all 0s */
}

#define poly 0xA011
unsigned long scramble(int reset, unsigned long dword) {
static unsigned short scramble;
int i;

  if (reset)
     scramble = 0xFFFF;
  for (i = 0; i < 32; i++)  /* serial shift register implementation */
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
  {
      dword ^= (scramble & 0x8000)? (1 << i):0;
      scramble = (scramble << 1) ^ ((scramble & 0x8000)? poly:0);
  }
  return dword;
}
```

E.3 Scrambler implementation with XORs

These equations generate the scrambled dwords to XOR with dwords before transmission and dword reception to recover the original data. The ^ symbol represents an XOR operation. The initialized value for d[15:0] is F0F6h (i.e., 0xF0F6) in this example.

```
scr0=d15^d13^d4^d0;
scr1=d15^d14^d13^d5^d4^d1^d0;
scr2=d14^d13^d6^d5^d4^d2^d1^d0;
scr3=d15^d14^d7^d6^d5^d3^d2^d1;
scr4=d13^d8^d7^d6^d3^d2^d0;
scr5=d14^d9^d8^d7^d4^d3^d1;
scr6=d15^d10^d9^d8^d5^d4^d2;
scr7=d15^d13^d11^d10^d9^d6^d5^d4^d3^d0;
scr8=d15^d14^d13^d12^d11^d10^d7^d6^d5^d1^d0;
scr9=d14^d12^d11^d8^d7^d6^d4^d2^d1^d0;
scr10=d15^d13^d12^d9^d8^d7^d5^d3^d2^d1;
scr11=d15^d14^d10^d9^d8^d6^d3^d2^d0;
scr12=d13^d11^d10^d9^d7^d3^d1^d0;
scr13=d14^d12^d11^d10^d8^d4^d2^d1;
scr14=d15^d13^d12^d11^d9^d5^d3^d2;
scr15=d15^d14^d12^d10^d6^d3^d0;
scr16=d11^d7^d1^d0;
scr17=d12^d8^d2^d1;
scr18=d13^d9^d3^d2;
scr19=d14^d10^d4^d3;
scr20=d15^d11^d5^d4;
scr21=d15^d13^d12^d6^d5^d4^d0;
scr22=d15^d14^d7^d6^d5^d4^d1^d0;
scr23=d13^d8^d7^d6^d5^d4^d2^d1^d0;
scr24=d14^d9^d8^d7^d6^d5^d3^d2^d1;
scr25=d15^d10^d9^d8^d7^d6^d4^d3^d2;
scr26=d15^d13^d11^d10^d9^d8^d7^d5^d3^d0;
scr27=d15^d14^d13^d12^d11^d10^d9^d8^d6^d1^d0;
scr28=d14^d12^d11^d10^d9^d7^d4^d2^d1^d0;
scr29=d15^d13^d12^d11^d10^d8^d5^d3^d2^d1;
scr30=d15^d14^d12^d11^d9^d6^d3^d2^d0;
scr31=d12^d10^d7^d3^d1^d0;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

E.4 Scrambler examples

Table E.1 shows several scrambler examples. Data is shown in dwords, from first to last.

Table E.1 — Scrambler examples

Data	Scrambled output
06D0B992h	C402CF1Fh
00B5DF59h	1F936C31h
00000000h	A508436Ch
00000000h	3452D354h
1234FFFFh	98616AFDh
00000000h	BB1ABE1Bh
00000000h	FA56B73Dh
00000000h	53F60B1Bh
00000000h	F0809C41h
08000012h	7C7FC358h
01000000h	BF865291h
00000000h	7A6FA7B6h
00000000h	3163E6D6h
3F4F1C26h	CF79E22Ah
00000000h	C2D2768Dh
00000000h	1F26B368h
00000000h	A508436Ch
00000000h	3452D354h
00000000h	8A559502h
00000000h	BB1ABE1Bh
00000000h	FA56B73Dh
00000000h	53F60B1Bh
00000000h	F0809C41h
00000000h	747FC34Ah
00000000h	BE865291h
00000000h	7A6FA7B6h
00000000h	3163E6D6h
B00F2BCCh [a]	4039D5C0h

[a] The last dword represents a CRC dword.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex F
(informative)

ATA architectural notes

F.1 STP differences from Serial ATA (SATA)

Some of the differences of STP compared with SATA are:

a) STP adds addressing of multiple SATA devices. Each SATA device is assigned a SAS address by its attached expander device with STP/SATA bridge functionality. The STP initiator port understands addressing more than one STP target port;
b) STP allows multiple STP initiator ports to share access to a SATA device using an active/standby mode called affiliations (see 7.17.3);
c) interface power management is not supported;
d) far-end analog loopback testing is not supported;
e) far-end retimed loopback testing is not supported;
f) near-end analog loopback testing is not supported;
g) use of SATA_CONT is required; and
h) BIST Activate frames are not supported.

F.2 STP differences from Serial ATA II

The following features of Serial ATA II are specifically excluded from SAS STP or handled differently in a SAS domain:

a) extended differential voltages;
b) enclosure services;
c) staggered device spinup;
d) drive activity indication;
e) drive hot-plug improvements; and
f) power management improvements.

F.3 Affiliation policies

F.3.1 Affiliation policies overview

SATA is based on a model that assumes a SATA device is controlled by a single SATA host, and does not address the notion of multiple SATA hosts having the ability to access any given SATA device.

With STP/SATA bridges, SATA devices are cast into an environment where multiple STP initiator ports, assuming the role of SATA hosts in a SATA domain, have access to the same SATA devices in the SAS domain. The SATA protocol used inside STP connections does not account for the possibility that more than one STP initiator port might be vying for access to the SATA device. Affiliations provide a way to ensure contention for a SATA device does not result in incoherent access to the SATA device when commands from different STP initiator ports collide at the SATA device.

To prevent a SATA device from confusing commands from one STP initiator port with commands from another, an STP initiator port needs a means to maintain exclusive access to a SATA or STP device for the duration of the processing of a command.

For example, consider the case where an STP initiator port establishes a connection to send a command (perhaps a read), and then closes the connection while the SATA device (e.g., a disk drive) retrieves the data (e.g., performs a seek operation to the track containing the data). If, after the connection is closed, another STP initiator port is allowed to establish a connection and send another command, the SATA device would no longer have a means to determine which STP initiator port should receive the data when the device requests the connection to send the data for the first command. This is because, unlike SCSI target devices, SATA devices have no notion of multiple SATA hosts.

The consequences are worse for write commands since the result could be wrong data written to media, with the original data being overwritten and permanently lost.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Affiliation provides a means for an STP initiator port to establish atomic access to a SATA device across the processing of a command or series of commands to the SATA device, without requiring the STP initiator port to maintain a connection open to the STP target port for the duration of command processing.

F.3.2 Affiliation policy for static STP initiator port to STP target port mapping

Affiliations should not be used to enforce policies establishing fixed associations between STP initiator ports and STP target ports.

F.3.3 Affiliation policy with SATA queued commands and multiple STP initiator ports

STP initiator ports using queued commands when other STP initiator ports may be accessing the same STP target port should, at vendor-specific intervals, allow commands to complete and release the affiliation to allow other STP initiator ports access to the STP target port.

F.3.4 Applicability of affiliation for STP target ports

Affiliation may or may not be necessary for STP target ports depending on whether the STP target port tracks the STP initiator port's SAS address on each command received. If the STP target port has the means to manage and track commands from each STP initiator port independently, then affiliations are not necessary because the STP target port is capable of associating each information transfer with the appropriate STP initiator port, and is capable of establishing a connection to the appropriate STP initiator port when sending information back for a command.

An STP target port that behaves the same as a SATA device, in that it maintains only a single ATA task file register image to be shared among all STP initiator ports, supports affiliations in order to provide a way for STP initiator ports to maintain exclusive access to the STP target port while commands remain outstanding. In this model, an STP target port is capable of establishing connections to an STP initiator port, but is only capable of remembering the SAS address of the last STP initiator port to establish a connection, and therefore is only capable of requesting a connection back to that same STP initiator port.

See 10.4.3.7 for an explanation of how an STP target port reports support for affiliations.

--`,,,`,,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex G
(informative)

Expander device handling of connections

G.1 Expander device handling of connections overview

This annex provides examples of how expander devices process connection requests.

Figure G.1 shows the topology used by examples in this annex.

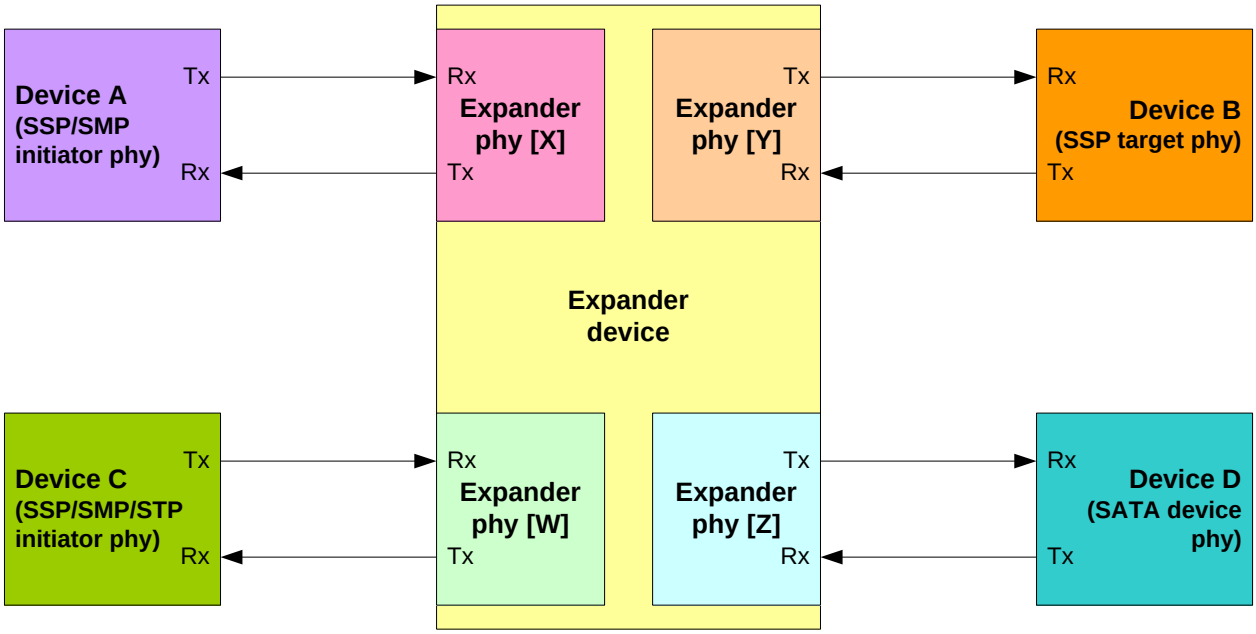


Figure G.1 — Example topology

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table G.1 defines the column headers used within the figures contained within this annex.

Table G.1 — Column descriptions for connection examples

Column header	Description
Phy [W] Rx	Expander phy [W] Receive from device C
Phy [W] Tx	Expander phy [W] Transmit to device C
Phy [W] XL state	Expander phy [W] XL state machine state (see 7.15)
Phy [W] XL req/rsp	Expander phy [W] XL requests and responses (see 4.6.6)
Phy [W] XL cnf/ind	Expander phy [W] XL confirmations and indications (see 4.6.6)
Phy [X] Rx	Expander phy [X] Receive from device A
Phy [X] Tx	Expander phy [X] Transmit to device A
Phy [X] XL state	Expander phy [X] XL state machine state (see 7.15)
Phy [X] XL req/rsp	Expander phy [X] XL requests and responses (see 4.6.6)
Phy [X] XL cnf/ind	Expander phy [X] XL confirmations and indications (see 4.6.6)
Phy [Y] XL cnf/ind	Expander phy [Y] XL confirmations and indications (see 4.6.6)
Phy [Y] XL req/rsp	Expander phy [Y] XL requests and responses (see 4.6.6)
Phy [Y] XL state	Expander phy [Y] XL state machine state (see 7.15)
Phy [Y] Tx	Expander phy [Y] Transmit to device B
Phy [Y] Rx	Expander phy [Y] Receive from device B
Phy [Z] XL cnf/ind	Expander phy [Y] XL confirmations and indications (see 4.6.6)
Phy [Z] XL req/rsp	Expander phy [Y] XL requests and responses (see 4.6.6)
Phy [Z] XL state	Expander phy [Y] XL state machine state (see 7.15)
Phy [Z] Tx	Expander phy [Y] Transmit to device D
Phy [Z] Rx	Expander phy [Y] Receive from device D

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

G.2 Connection request - OPEN_ACCEPT

Figure G.2 shows the establishment of a successful connection between two end devices.

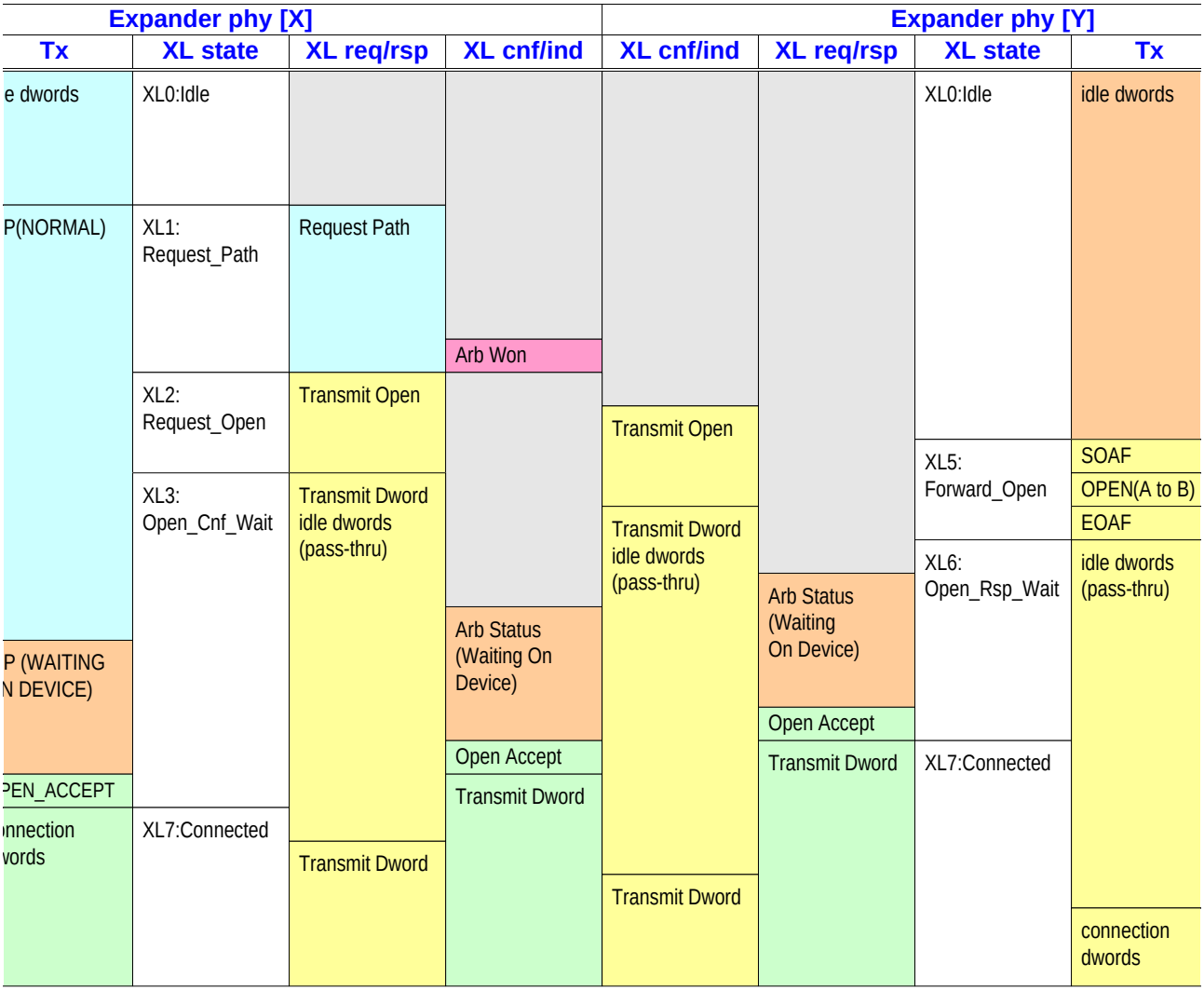


Figure G.2 — Connection request - OPEN_ACCEPT

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.3 Connection request - OPEN_REJECT by end device

Figure G.3 shows failure to establish a connection due to rejection of the connection request by an end device.

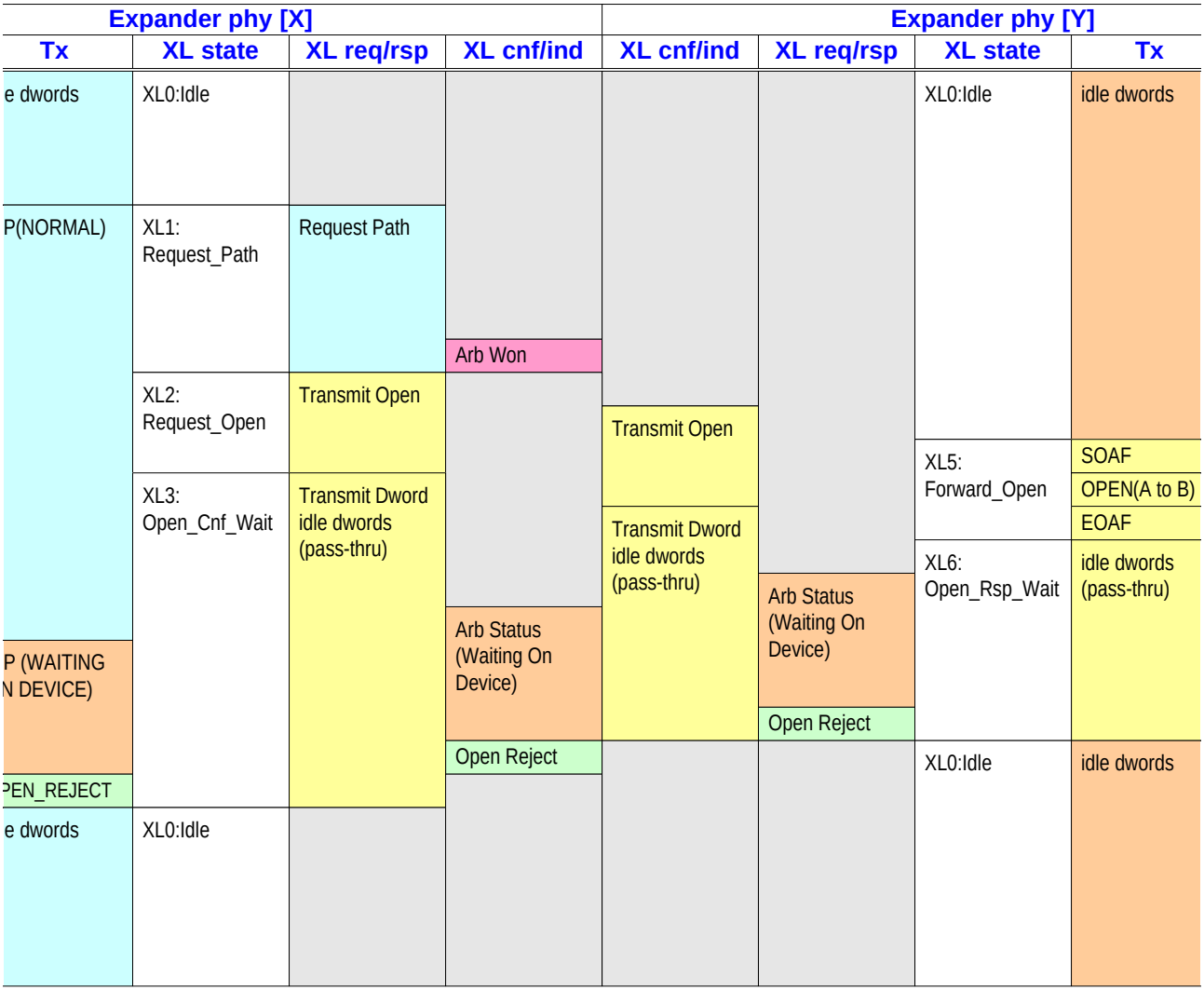


Figure G.3 — Connection request - OPEN_REJECT by end device

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.4 Connection request - OPEN_REJECT by expander device

Figure G.4 shows failure to establish a connection due to rejection of the connection request by an expander device.

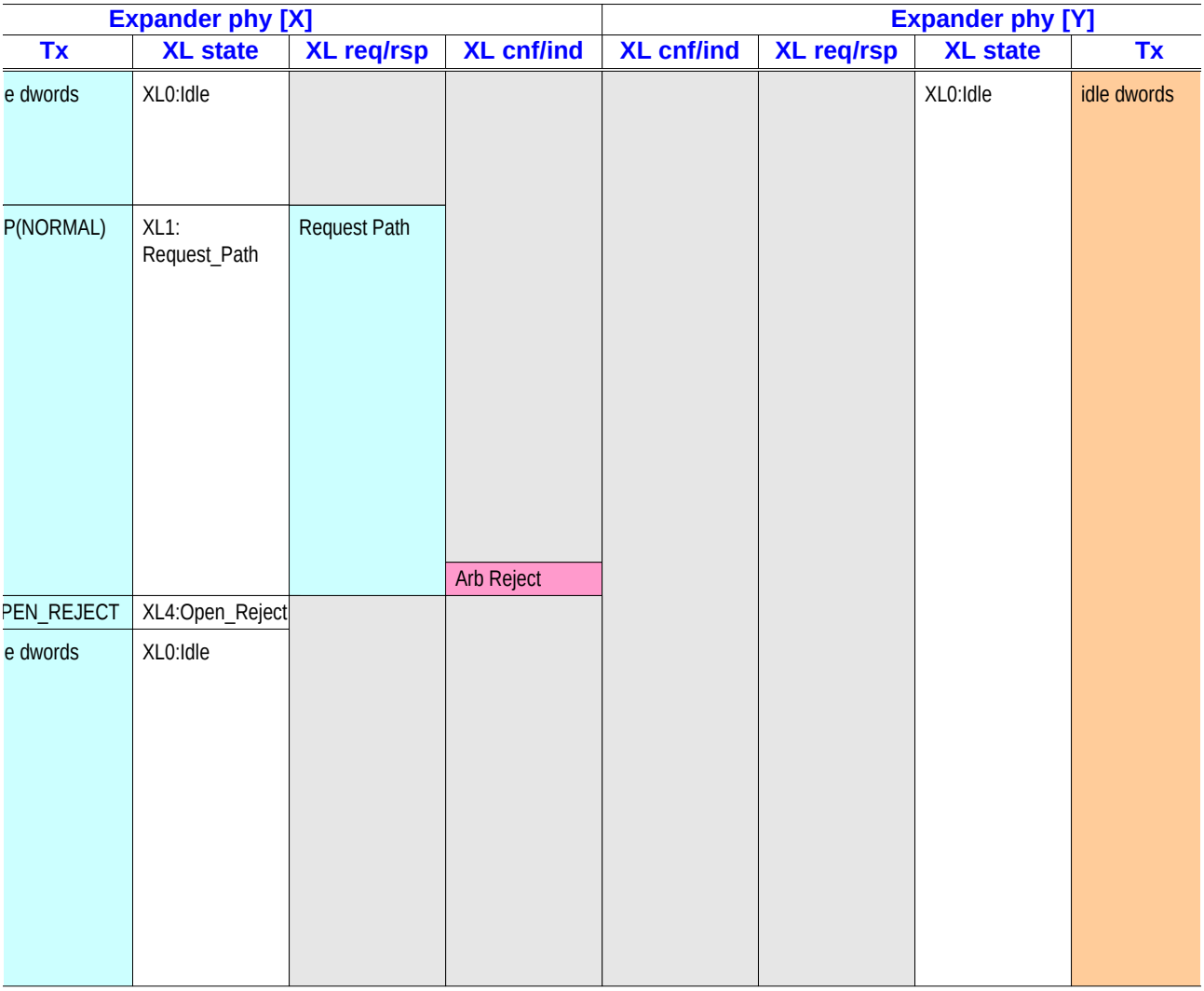


Figure G.4 — Connection request - OPEN_REJECT by expander device

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

G.5 Connection request - arbitration lost

Figure G.5 shows two end devices attempting to establish a connection at the same time. This example assumes that the OPEN (A to B) address frame has higher priority than the OPEN (B to A) address frame and therefore device A wins arbitration and device B loses arbitration.

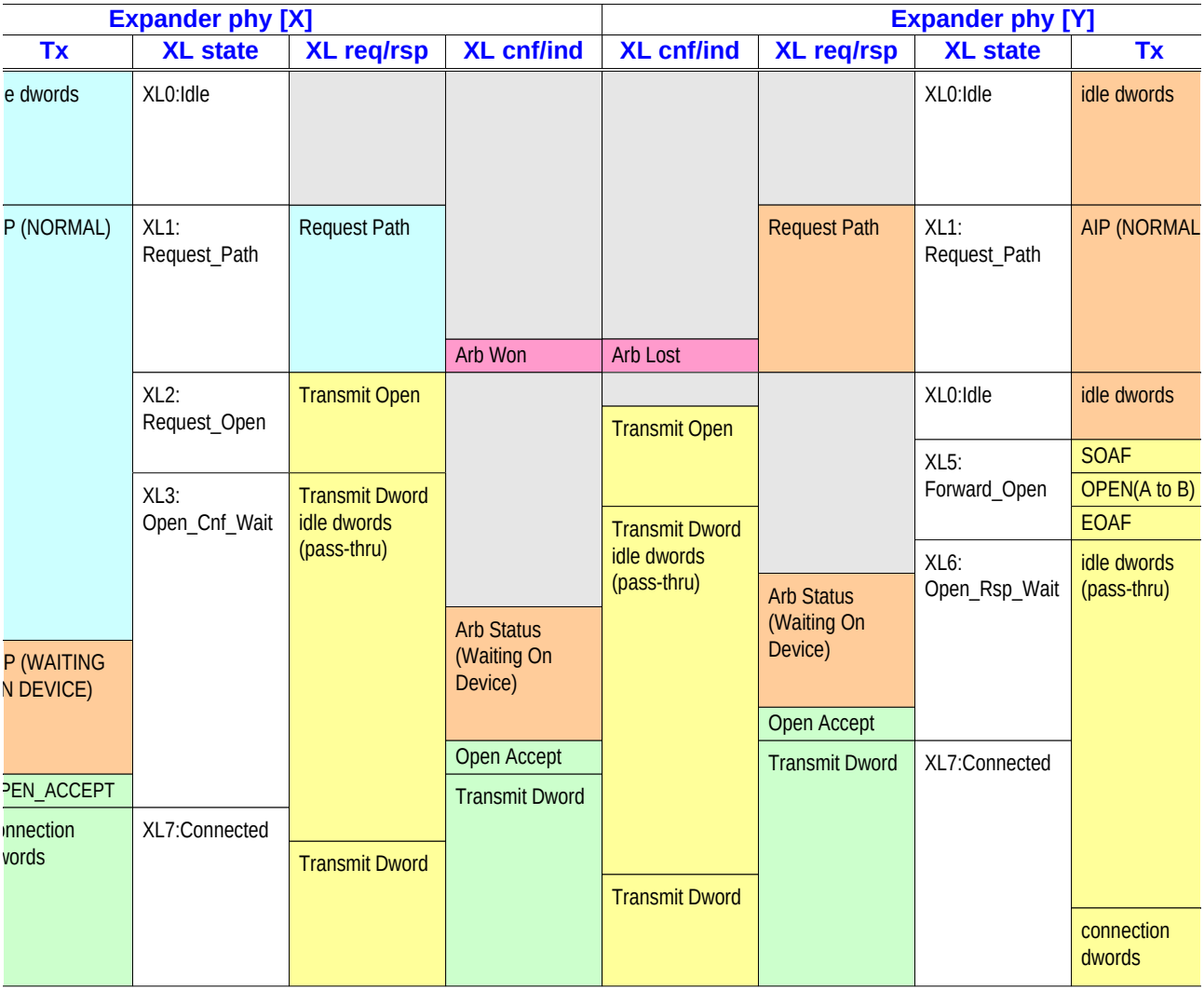


Figure G.5 — Connection request - arbitration lost

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.6 Connection request - backoff and retry

Figure G.6 shows a higher priority OPEN address frame (B to C) received by a phy which has previously forwarded an OPEN address frame (A to B) whose source (A) differs from the winning destination (C). In this case expander phy [X] is required to back off and retry path arbitration (see 7.15.9).

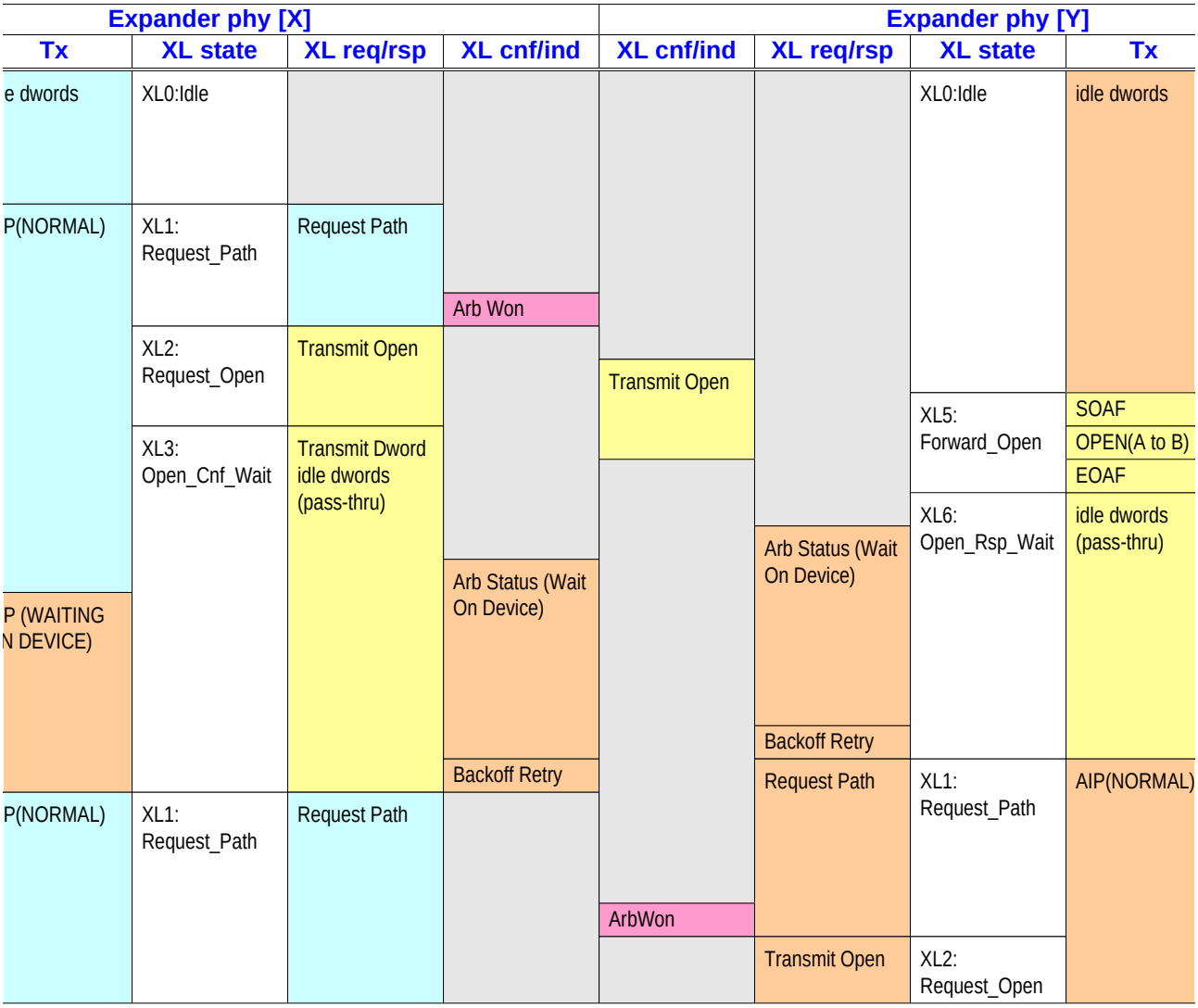


Figure G.6 — Connection request - backoff and retry

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.7 Connection request - backoff and reverse path

Figure G.7 shows a higher priority OPEN address frame (B to A) received by a phy which has previously forwarded an OPEN address frame (A to B) whose source (A) matches the winning destination (A). In this case expander phy [Y] forwards the higher priority OPEN to expander phy [X] (see 7.15.9).

Expander phy [X]				Expander phy [Y]			
Tx	XL state	XL req/rsp	XL cnf/ind	XL cnf/ind	XL req/rsp	XL state	Tx
e dwords	XL0:Idle					XL0:Idle	idle dwords
P(NORMAL)	XL1: Request_Path	Request Path	ArbWon				
	XL2: Request_Open	Transmit Open		Transmit Open		XL5: Forward_Open	SOAF
							OPEN(A to B)
	XL3: Open_Cnf_Wait	Transmit Dword idle dwords (pass-thru)					EOAF
			Arb Status (Wait On Device)		Arb Status (Wait On Device)	XL6: Open_Rsp_Wait	idle dwords (pass-thru)
P (WAITING N DEVICE)							
			Backoff Reverse		Backoff Reverse		
AF	XL5: Forward_Open		Transmit Open		Transmit Open	XL2: Request_Open	AIP(NORMAL)
PEN(B to A)							
AF						XL3: Open_Cnf_Wait	
e dwords ass-thru)	XL6: Open_Rsp_Wait						
		Arb Status - wait on device		Arb Status - wait on device			AIP (WAITING ON DEVICE)

Figure G.7 — Connection request - backoff and reverse path

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.8 Connection close - single step

Figure G.8 shows an end device initiating the closing of a connection by transmitting CLOSE, followed by another end device responding with CLOSE at a later time.

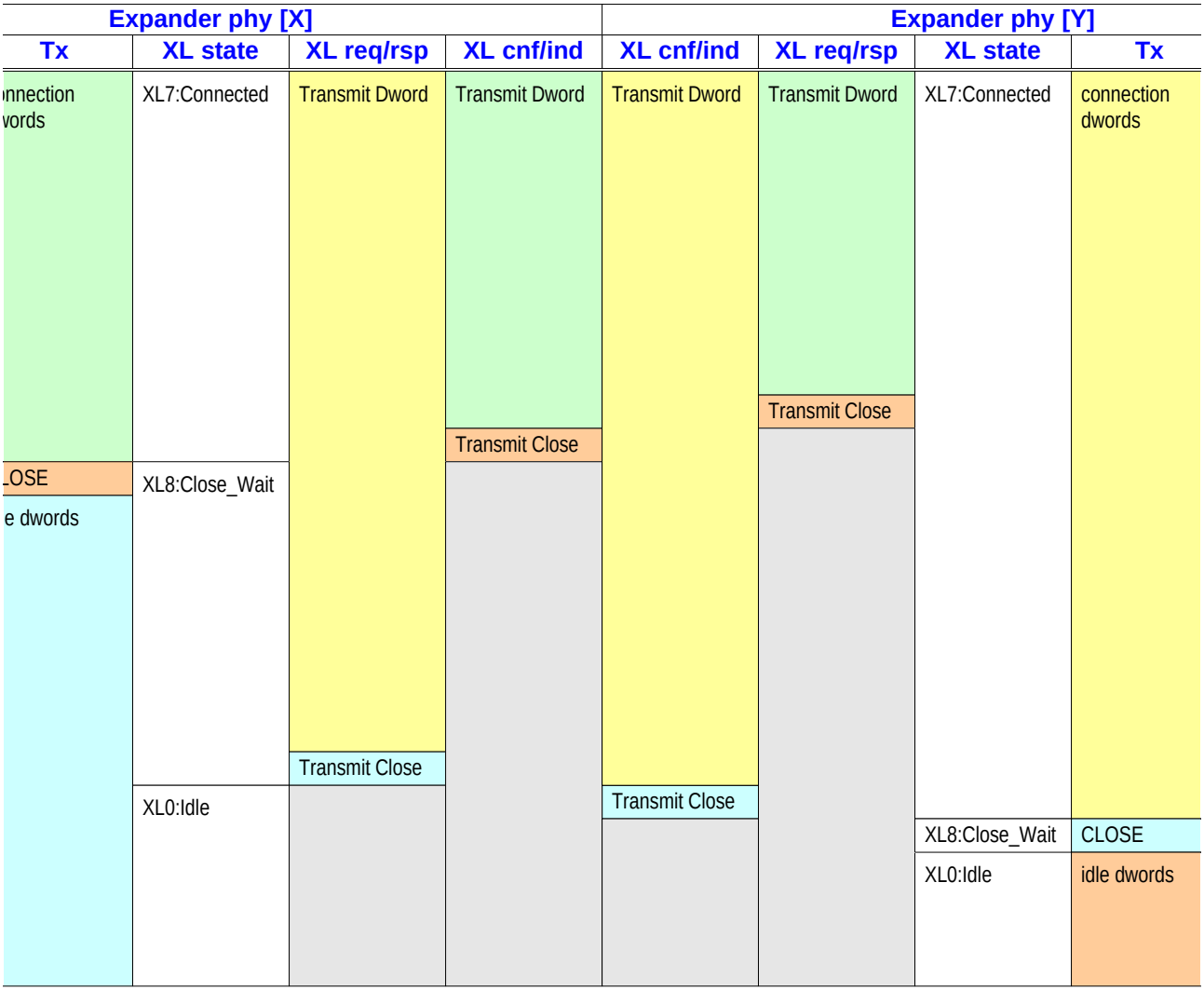


Figure G.8 — Connection close - single step

--`,,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.9 Connection close - simultaneous

Figure G.9 shows two end devices simultaneously transmitting CLOSE to each other.

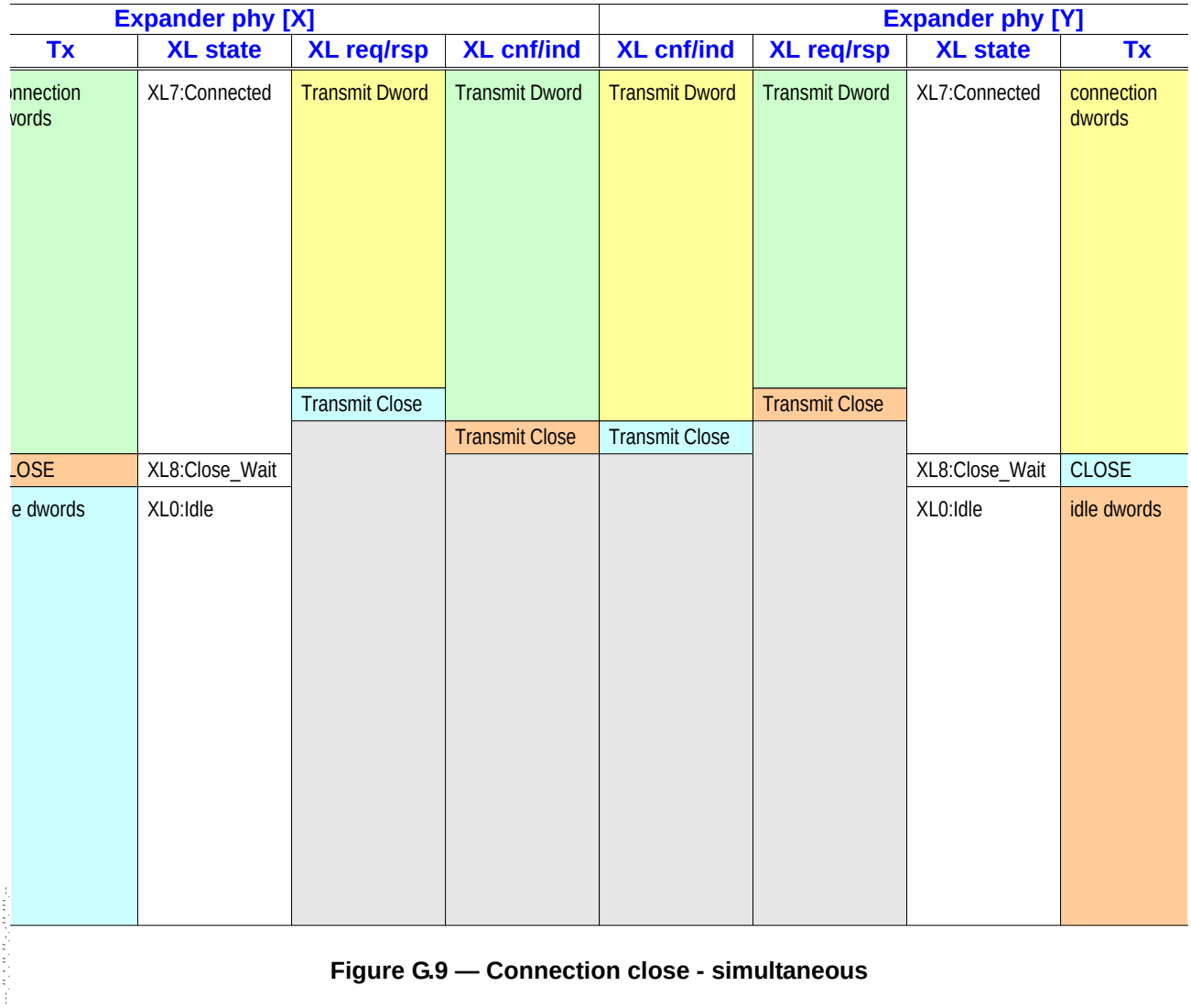


Figure G.9 — Connection close - simultaneous

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.10 BREAK handling during path arbitration

Figure G.10 shows an expander device responding to the reception of a BREAK during path arbitration.

Expander phy [X]				Expander phy [Y]			
Tx	XL state	XL req/rsp	XL cnf/ind	XL cnf/ind	XL req/rsp	XL state	Tx
e dwords	XL0:Idle					XL0:Idle	idle dwords
P (NORMAL)	XL1: Request_Path	Request Path					
REAK	XL9:Break						
e dwords	XL0:Idle						

Figure G.10 — BREAK handling during path arbitration

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

G.11 BREAK handling during connection

Figure G.11 shows an expander device responding to the reception of a BREAK during a connection.

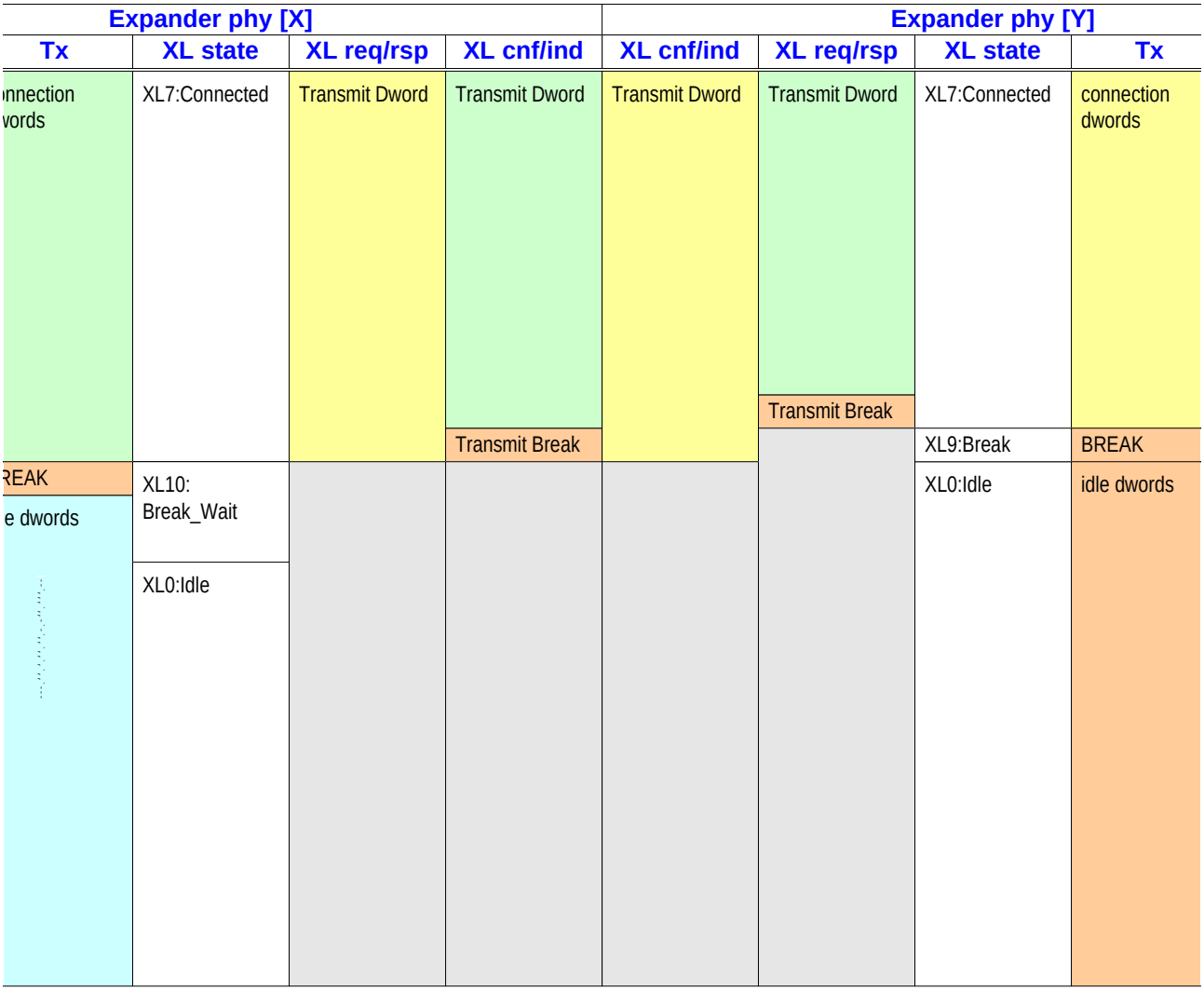
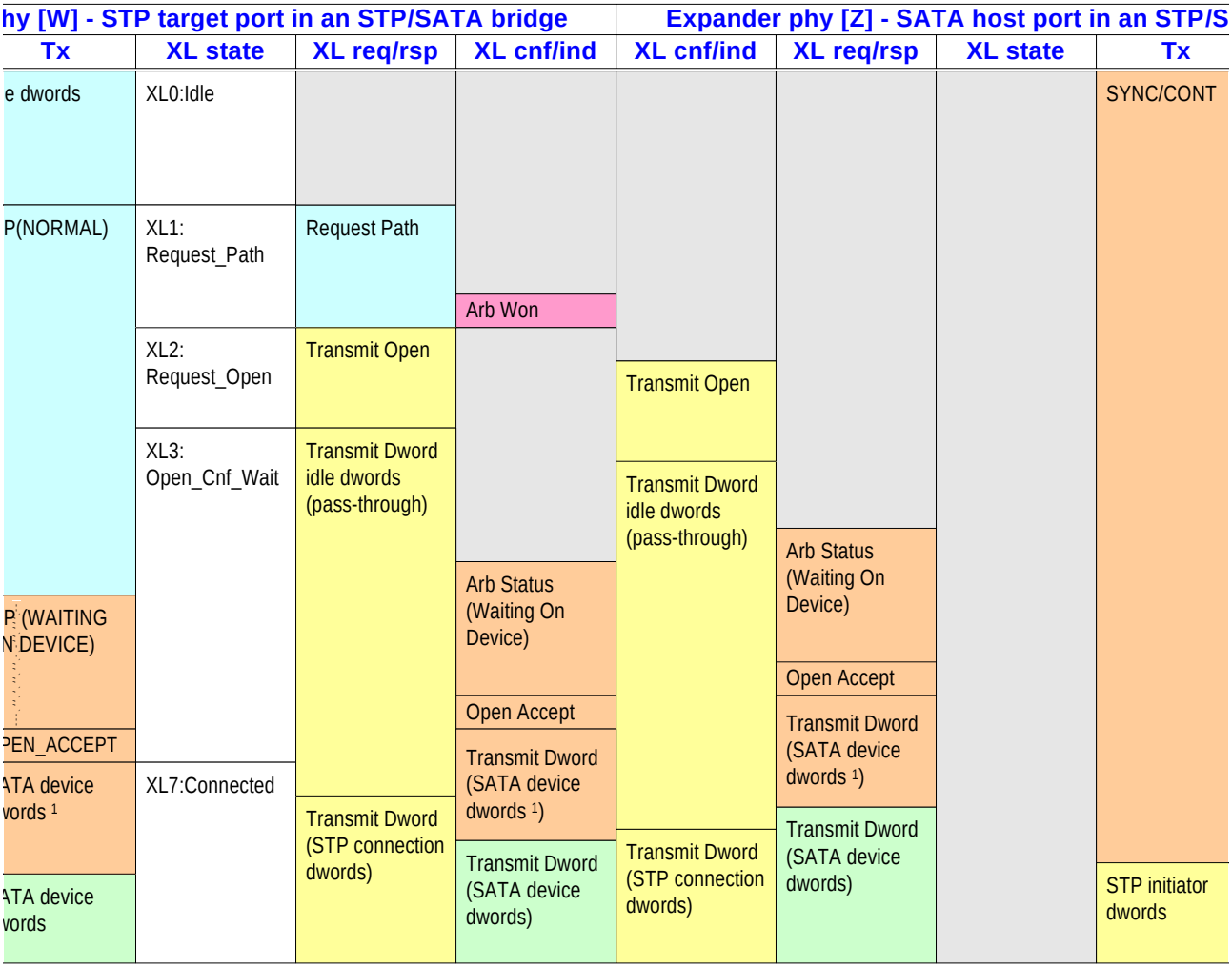


Figure G.11 — BREAK handling during a connection

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.12 STP connection - originated by STP initiator port

Figure G.12 shows an STP initiator port originating a connection to an STP target port in an STP/SATA bridge.



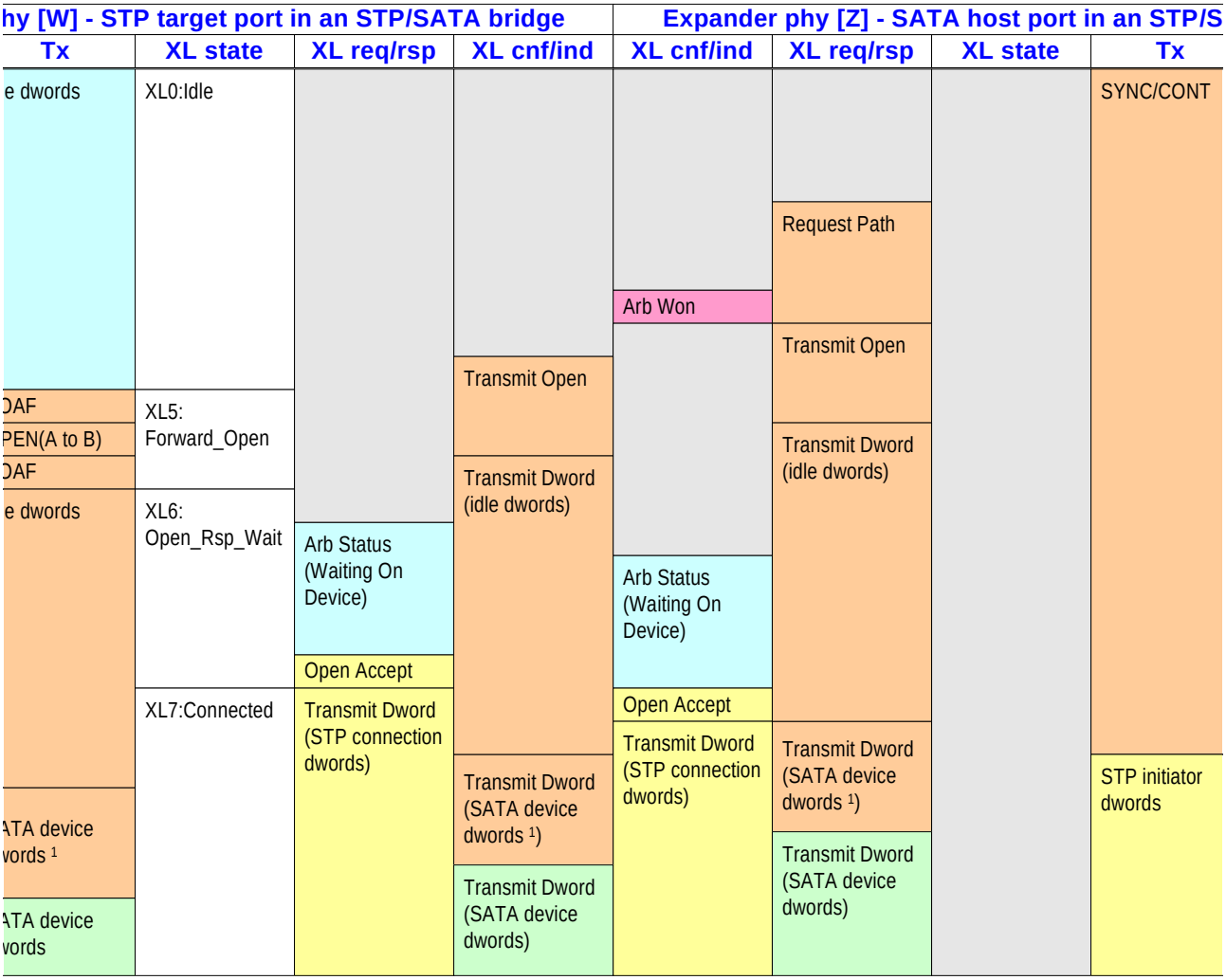
dge duplicates the dword stream which is being received from the SATA device before forwarding dwords - th TA primitive is correctly forwarded to the STP initiator port.

Figure G.12 — STP connection - originated by STP initiator port

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.13 STP connection - originated by STP target port in an STP/SATA bridge

Figure G.13 shows an STP target port in an STP/SATA bridge originating a connection on behalf of a SATA device which is requesting to transmit a frame.



dge duplicates the dword stream which is being received from the SATA device before forwarding dwords - th TA primitive is correctly forwarded to the STP initiator port.

Figure G.13 — STP connection - originated by STP target port in an STP/SATA bridge

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.14 STP connection close - originated by STP initiator port

Figure G.14 shows an STP initiator port closing a connection to an STP target port in an STP/SATA bridge.

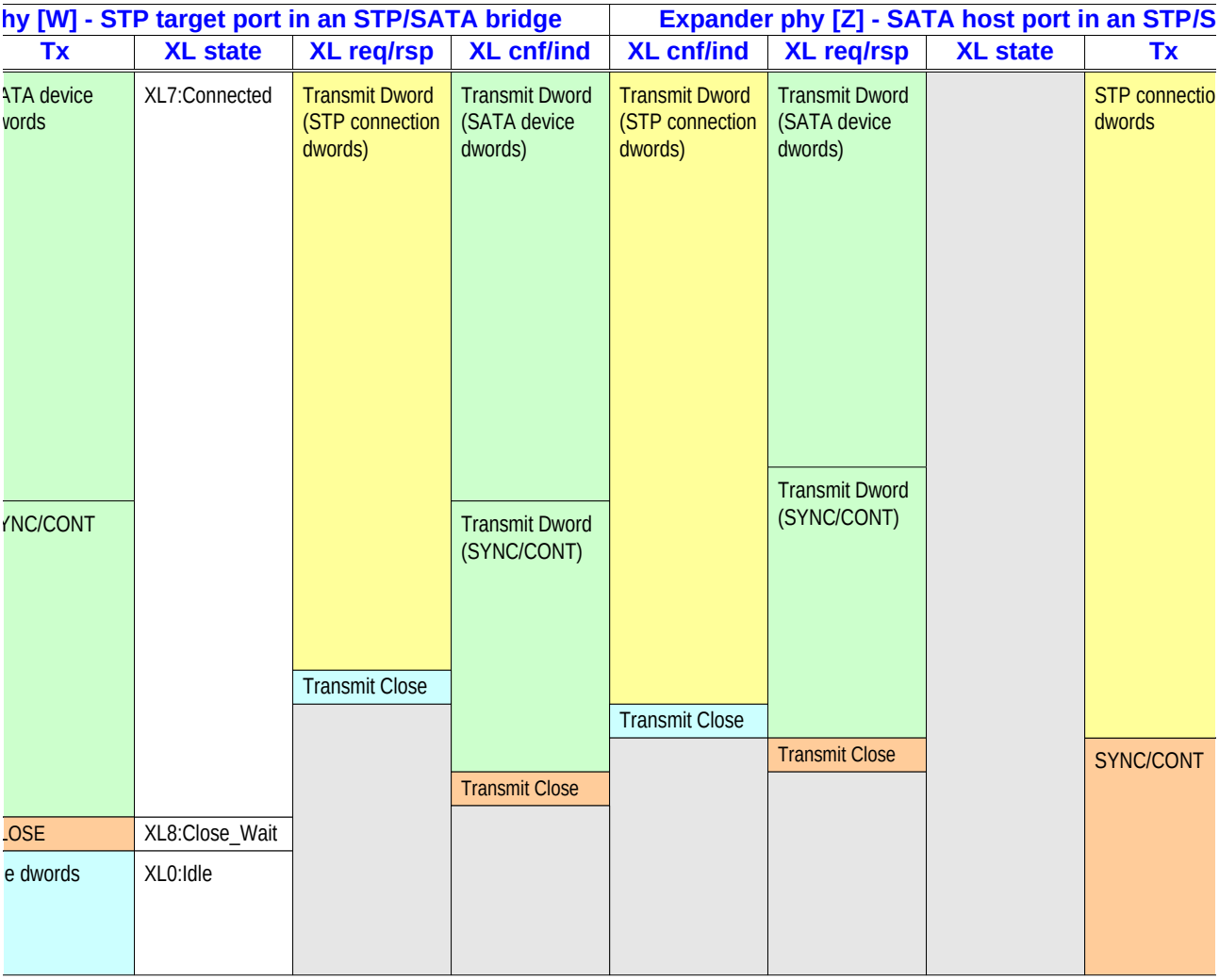


Figure G.14 — STP connection close - originated by STP initiator port

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.15 STP connection close - originated by STP target port in an STP/SATA bridge

Figure G.15 shows an STP target port in an STP/SATA bridge closing an STP connection.

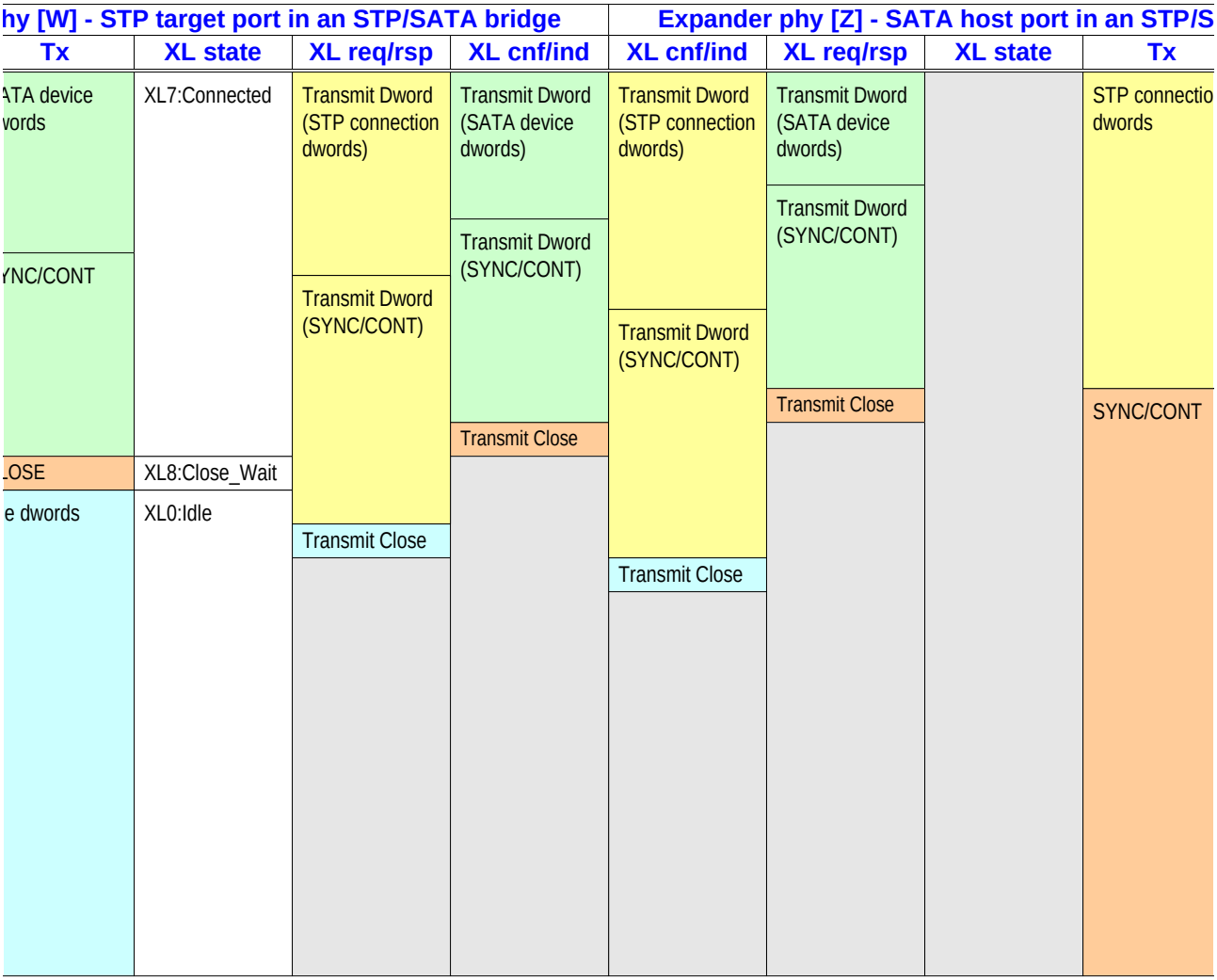


Figure G.15 — STP connection close - originated by STP target port in an STP/SATA bridge

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.16 Pathway blocked and pathway recovery example

Figure G.16 shows a topology used to illustrate pathway recovery.

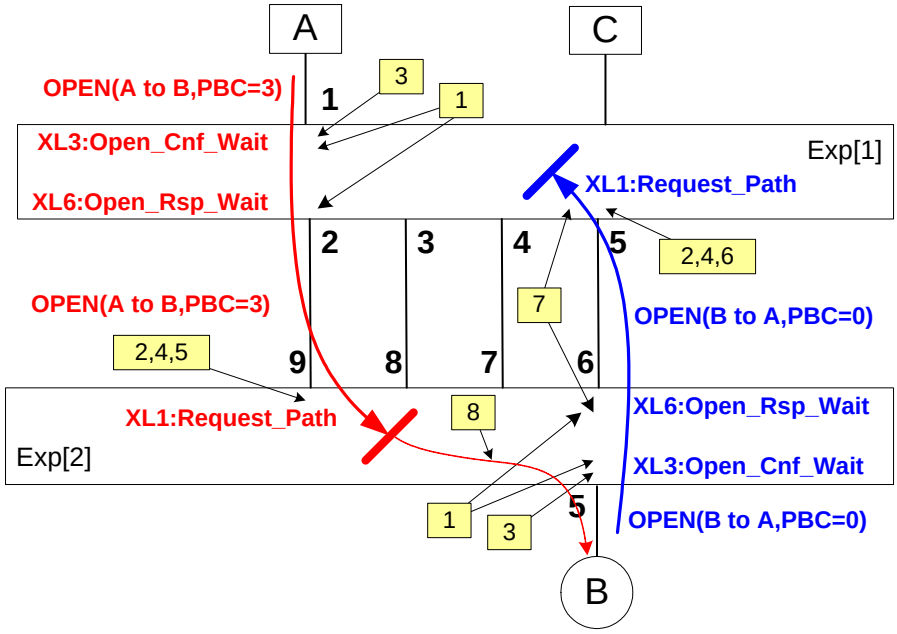


Figure G.16 — Partial pathway recovery

The sequence of events used to identify pathway blockage and to perform pathway recovery are as follows:

1) Exp[1].Phy[1,2] and Exp[2].Phy[5,6] send Phy Status (Partial Pathway) responses to ECM to indicate that they contain partial pathways;
2) Exp[1].Phy[5] and Exp[2].Phy[9] receive Arbitrating (Waiting On Partial) confirmations from ECM which cause them to transmit AIP (WAITING ON PARTIAL);
3) AIP(WAITING ON PARTIAL) received by Exp[1].Phy[2] and Exp[2].Phy[6] then forwarded to Exp[1].Phy[1] and Exp[2].Phy[5] as an Arb Status (Waiting On Partial) confirmation. Exp[1].Phy[1] and Exp[2].Phy[5] send Phy Status (Blocked On Partial) responses to ECM as confirmation that they are blocked waiting on a partial pathway in another expander device;
4) Exp[1].Phy[5] and Exp[2].Phy[9] receive Arbitrating (Blocked On Partial) confirmations from ECM while all destination phys send Phy Status (Blocked On Partial) responses which cause them to run their Partial Pathway Timeout timers;
5) Partial Pathway Timeout timer expires in Exp[2].Phy[9] – this causes a request to the ECM to resolve pathway blockage. Pathway recovery priority for this phy is not lower than all phys within the destination port which are also blocked. ECM does not provide an Arb Reject (Pathway Blocked) confirmation to Exp[2].Phy[9] so this expander phy waits for pathway resolution to occur elsewhere in the topology;
6) Partial Pathway Timeout timer expires in Exp[1].Phy[5] – this causes a request to the ECM to resolve pathway blockage. Pathway recovery priority for this expander phy is lower than all expander phys within the destination port which are also blocked. ECM provides an Arb Reject (Pathway Blocked) confirmation to Exp[1].Phy[5] which instructs this expander phy to reject the connection request using OPEN_REJECT (PATHWAY BLOCKED);
7) OPEN_REJECT (PATHWAY BLOCKED) tears down partial pathway all the way to the originating end device (Device B); and
8) Partial pathway established between Exp[2].Phy[9] and Exp[2].Phy[5] and OPEN(A to B) is delivered to device B.

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex H
(informative)

Primitive encoding

This annex describes a set of the K28.5-based primitive encodings whose 40-bit values (after 8b10b encoding with either starting running disparity) have a Hamming distance (i.e., the number of bits different in two patterns) of at least 8. All the primitive encodings in 7.2 were selected from this list. Unassigned encodings may be used by future versions of this standard.

Table H.1 — Primitives with Hamming distance of 8 (Sheet 1 of 3)

1st	2nd	3rd	4th	Assignment
K28.5	D01.3	D01.3	D01.3	ALIGN (2)
K28.5	D01.4	D01.4	D01.4	ACK
K28.5	D01.4	D02.0	D31.4	RRDY (RESERVED 0)
K28.5	D01.4	D04.7	D24.0	NAK (RESERVED 1)
K28.5	D01.4	D07.3	D30.0	CREDIT_BLOCKED
K28.5	D01.4	D16.7	D07.3	NAK (RESERVED 2)
K28.5	D01.4	D24.0	D16.7	RRDY (NORMAL)
K28.5	D01.4	D27.4	D04.7	NAK (CRC ERROR)
K28.5	D01.4	D30.0	D02.0	RRDY (RESERVED 1)
K28.5	D01.4	D31.4	D29.7	NAK (RESERVED 0)
K28.5	D02.0	D01.4	D29.7	ERROR
K28.5	D02.0	D02.0	D02.0	HARD_RESET
K28.5	D02.0	D04.7	D01.4	CLOSE (RESERVED 1)
K28.5	D02.0	D07.3	D04.7	CLOSE (CLEAR AFFILIATION)
K28.5	D02.0	D16.7	D31.4	
K28.5	D02.0	D24.0	D07.3	BREAK
K28.5	D02.0	D29.7	D16.7	
K28.5	D02.0	D30.0	D27.4	CLOSE (NORMAL)
K28.5	D02.0	D31.4	D30.0	CLOSE (RESERVED 0)
K28.5	D04.7	D01.4	D24.0	BROADCAST (RESERVED 1)
K28.5	D04.7	D02.0	D01.4	BROADCAST (CHANGE)
K28.5	D04.7	D04.7	D04.7	BROADCAST (RESERVED 2)
K28.5	D04.7	D07.3	D29.7	BROADCAST (RESERVED 0)
K28.5	D04.7	D16.7	D02.0	BROADCAST (RESERVED 3)
K28.5	D04.7	D24.0	D31.4	BROADCAST (RESERVED CHANGE 0)
K28.5	D04.7	D27.4	D07.3	BROADCAST (RESERVED CHANGE 1)
K28.5	D04.7	D29.7	D30.0	BROADCAST (RESERVED 4)
K28.5	D04.7	D31.4	D27.4	
K28.5	D07.0	D07.0	D07.0	ALIGN (1)
K28.5	D07.3	D01.4	D31.4	
K28.5	D07.3	D02.0	D04.7	

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table H.1 — Primitives with Hamming distance of 8 (Sheet 2 of 3)

1st	2nd	3rd	4th	Assignment
K28.5	D07.3	D04.7	D30.0	
K28.5	D07.3	D07.3	D07.3	
K28.5	D07.3	D24.0	D29.7	
K28.5	D07.3	D27.4	D16.7	
K28.5	D07.3	D29.7	D27.4	
K28.5	D07.3	D30.0	D24.0	
K28.5	D07.3	D31.4	D02.0	
K28.5	D10.2	D10.2	D27.3	ALIGN (0)
K28.5	D16.7	D01.4	D02.0	
K28.5	D16.7	D02.0	D07.3	
K28.5	D16.7	D04.7	D31.4	
K28.5	D16.7	D16.7	D16.7	OPEN_ACCEPT
K28.5	D16.7	D24.0	D27.4	
K28.5	D16.7	D27.4	D30.0	
K28.5	D16.7	D29.7	D24.0	
K28.5	D16.7	D30.0	D04.7	
K28.5	D16.7	D31.4	D01.4	
K28.5	D24.0	D01.4	D16.7	
K28.5	D24.0	D02.0	D29.7	
K28.5	D24.0	D04.7	D07.3	SOF
K28.5	D24.0	D07.3	D31.4	EOAF
K28.5	D24.0	D16.7	D27.4	EOF
K28.5	D24.0	D24.0	D24.0	
K28.5	D24.0	D27.4	D02.0	
K28.5	D24.0	D29.7	D04.7	
K28.5	D24.0	D30.0	D01.4	SOAF
K28.5	D27.3	D27.3	D27.3	ALIGN (3)
K28.5	D27.4	D01.4	D07.3	AIP (RESERVED WAITING ON PARTIAL)
K28.5	D27.4	D04.7	D02.0	
K28.5	D27.4	D07.3	D24.0	AIP (WAITING ON CONNECTION)
K28.5	D27.4	D16.7	D30.0	AIP (RESERVED 1)
K28.5	D27.4	D24.0	D04.7	AIP (WAITING ON PARTIAL)
K28.5	D27.4	D27.4	D27.4	AIP (NORMAL)
K28.5	D27.4	D29.7	D01.4	AIP (RESERVED 2)
K28.5	D27.4	D30.0	D29.7	AIP (WAITING ON DEVICE)
K28.5	D27.4	D31.4	D16.7	AIP (RESERVED 0)
K28.5	D29.7	D02.0	D30.0	OPEN_REJECT (RESERVED CONTINUE 0)
K28.5	D29.7	D04.7	D27.4	OPEN_REJECT (RESERVED STOP 1)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table H.1 — Primitives with Hamming distance of 8 (Sheet 3 of 3)

1st	2nd	3rd	4th	Assignment
K28.5	D29.7	D07.3	D16.7	OPEN_REJECT (RESERVED INITIALIZE 1)
K28.5	D29.7	D16.7	D04.7	OPEN_REJECT (PATHWAY BLOCKED)
K28.5	D29.7	D24.0	D01.4	OPEN_REJECT (RESERVED CONTINUE 1)
K28.5	D29.7	D27.4	D24.0	OPEN_REJECT (RETRY)
K28.5	D29.7	D29.7	D29.7	OPEN_REJECT (NO DESTINATION)
K28.5	D29.7	D30.0	D31.4	OPEN_REJECT (RESERVED INITIALIZE 0)
K28.5	D29.7	D31.4	D07.3	OPEN_REJECT (RESERVED STOP 0)
K28.5	D30.0	D01.4	D04.7	DONE (ACK/NAK TIMEOUT)
K28.5	D30.0	D02.0	D16.7	
K28.5	D30.0	D07.3	D27.4	DONE (CREDIT TIMEOUT)
K28.5	D30.0	D16.7	D01.4	DONE (RESERVED 0)
K28.5	D30.0	D24.0	D02.0	
K28.5	D30.0	D27.4	D29.7	DONE (RESERVED TIMEOUT 0)
K28.5	D30.0	D29.7	D31.4	DONE (RESERVED 1)
K28.5	D30.0	D30.0	D30.0	DONE (NORMAL)
K28.5	D30.0	D31.4	D24.0	DONE (RESERVED TIMEOUT 1)
K28.5	D31.3	D01.3	D07.0	NOTIFY (RESERVED 1)
K28.5	D31.3	D07.0	D01.3	NOTIFY (RESERVED 0)
K28.5	D31.3	D10.2	D10.2	NOTIFY (RESERVED 2)
K28.5	D31.3	D31.3	D31.3	NOTIFY (ENABLE SPINUP)
K28.5	D31.4	D01.4	D30.0	OPEN_REJECT (RESERVED ABANDON 3)
K28.5	D31.4	D02.0	D27.4	OPEN_REJECT (RESERVED ABANDON 0)
K28.5	D31.4	D04.7	D29.7	OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)
K28.5	D31.4	D07.3	D02.0	OPEN_REJECT (RESERVED ABANDON 2)
K28.5	D31.4	D16.7	D24.0	OPEN_REJECT (WRONG DESTINATION)
K28.5	D31.4	D27.4	D01.4	OPEN_REJECT (STP RESOURCES BUSY)
K28.5	D31.4	D29.7	D07.3	OPEN_REJECT (PROTOCOL NOT SUPPORTED)
K28.5	D31.4	D30.0	D16.7	OPEN_REJECT (RESERVED ABANDON 1)
K28.5	D31.4	D31.4	D31.4	OPEN_REJECT (BAD DESTINATION)

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex I
(informative)

Messages between state machines

I.1 Messages between phy layer and other layers

Table I.1 lists the requests from the management application layer and the link layer to the phy layer.

Table I.1 — Requests from management application layer or link layer to phy layer

Phy layer	← Request ––	Application layer or link layer
SP	Enter Partial	MA
	Enter Slumber	
	Exit Partial	
	Exit Slumber	
	Power on, hard reset, or Management Reset	
SP_DWS	Power on, hard reset, or Management Reset	MA
SP	Stop SNTT	SL_IR

Table I.2 lists the confirmations from the phy layer to the link layer.

Table I.2 — Confirmations from phy layer to link layer

Phy layer	–– Confirmation →	Link layer
SP	Phy Layer Not Ready	XL (in expander phy)
	SATA Spinup Hold	
SP	Start SL_IR Receiver	SL_IR
	Phy Layer Not Ready	
	Phy Layer Ready (SAS)	
	Phy Layer Ready (SATA)	

I.2 Messages between link layer, port layer, and management application layer for all protocols

Table I.3 lists the requests between the link layer and the port layer for all protocols.

Table I.3 — Requests between link layer and port layer

Link layer	← Request ––	Port layer
SL	Open Connection	PL
	Stop Arb	

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table I.4 lists the confirmations between the link layer and the port layer for all protocols.

Table I.4 — Confirmations between link layer and port layer

Link layer	–– Confirmation →	Port layer
SL	Open Failed (Retry)	PL
	Connection Opened (SSP, Source Opened)	
	Connection Opened (STP, Source Opened)	
	Connection Opened (SMP, Source Opened)	
	Connection Opened (SSP, Destination Opened)	
	Connection Opened (STP, Destination Opened)	
	Connection Opened (SMP, Destination Opened)	
	Connection Closed (Break Received)	
	Connection Closed (Normal)	
	Connection Closed (Close Timeout)	
	Connection Closed (Break Timeout)	

Table I.5 lists the requests from the management application layer to the link layer for all protocols.

Table I.5 — Requests from management application layer to link layer

Link layer	← Request ––	Application layer
SL_IR	Tx HARD_RESET	MA
	Tx IDENTIFY Address Frame	
SL	Transmit Broadcast (type)	MA

Table I.6 lists the confirmations between the link layer and the expander function or management application layer for all protocols.

Table I.6 — Confirmations between link layer and port layer, link layer, or application layer

Link layer	–– Confirmation →	Port layer, link layer, or application layer
SL	Change Received	MA
SL_IR	Phy Enabled	MA
	Phy Disabled	
	Address Frame Failed	
	HARD_RESET Transmitted	
	Identify Timeout	
	Identification Sequence Complete	
SL_IR	HARD_RESET Received	PL
	Phy Enabled	
	Phy Disabled	

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

I.3 Messages between link layer, port layer, and transport layer for SSP

Table I.7 lists the requests between the link layer, port layer, and SSP transport layer.

Table I.7 — Requests between link layer, port layer, and transport layer for SSP

Link layer	← Request ––	Port layer	← Request ––	Transport layer
SL		PL	Cancel	ST
	Accept_Reject Opens (Accept SSP)		Accept_Reject Opens (Accept SSP)	
	Accept_Reject Opens (Reject SSP)		Accept_Reject Opens (Reject SSP)	
SSP		PL	Transmit Frame (Interlocked)	ST
			Transmit Frame (Non-Interlocked)	
	Tx Frame (Balance Required)			
	Tx Frame (Balance Not Required)			
	Close Connection			

Table I.8 lists the confirmations from the port layer to the SSP transport layer.

Table I.8 — Confirmations from port layer to transport layer for SSP

Port layer	–– Confirmation →	Transport layer
PL	HARD_RESET Received	ST

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table I.9 lists the confirmations between the SL link layer, port layer, and SSP transport layer.

Table I.9 — Confirmations between SL link layer, port layer, and SSP transport layer

Link layer	–– Confirmation →	Port layer	–– Confirmation →	Transport layer
SL	Open Failed (Port Layer Request)	PL	Transmission Status (Cancel Acknowledge)	ST
	Open Failed (Wrong Destination)		Transmission Status (Wrong Destination)	
	Open Failed (Connection Rate Not Supported)		Transmission Status (Connection Rate Not Supported)	
	Open Failed (Protocol Not Supported)		Transmission Status (Protocol Not Supported)	
	Open Failed (No Destination)		Transmission Status (No Destination) or Transmission Status (I_T Nexus Loss)	
	Open Failed (Bad Destination)		Transmission Status (Bad Destination)	
	Open Failed (STP Resources Busy)		Transmission Status (STP Resources Busy)	
	Open Failed (Open Timeout Occurred)		Transmission Status (Open Timeout Occurred) or Transmission Status (I_T Nexus Loss)	
	Open Failed (Break Received)		Transmission Status (Break Received)	
	Open Failed (Pathway Blocked)			
	Connection Closed (any reason)		no confirmation or Transmission Status (Connection Lost Without ACK/NAK)	

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table I.10 lists the confirmations between the SSP link layer, port layer, and SSP transport layer.

Table I.10 — Confirmations between SSP link layer, port layer, and SSP transport layer

Link layer	–– Confirmation →	Port layer	–– Confirmation →	Transport layer
SSP	Frame Transmitted	PL	Transmission Status (Frame Transmitted)	ST
	ACK/NAK Timeout		Transmission Status (ACK/NAK Timeout)	
	Credit Timeout			
	DONE Transmitted			
	DONE Timeout			
	ACK Received		Transmission Status (ACK Received)	
	ACK Transmitted		ACK Transmitted	
	NAK Received		Transmission Status (NAK Received)	
	DONE Received			
	Frame Received (ACK/NAK Balanced)		Frame Received (ACK/NAK Balanced)	
	Frame Received (ACK/NAK Not Balanced)		Frame Received (ACK/NAK Not Balanced)	
			Transmission Status (No Phys In Port)	

I.4 Messages between link layer, port layer, and transport layer for SMP

Table I.11 lists the requests between the link layer, port layer, and SMP transport layer.

Table I.11 — Requests between SL/SMP link layer, port layer, and SMP transport layer

Link layer	← Request ––	Port layer	← Request ––	Transport layer
SL	Accept_Reject Opens (Accept SMP)	PL	Accept_Reject Opens (Accept SMP)	MT
	Accept_Reject Opens (Reject SMP)		Accept_Reject Opens (Reject SMP)	
SMP	Tx Frame	PL	Transmit Frame	MT
	SMP Transmit Break		SMP Transmit Break	

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table I.12 lists the confirmations between the link layer, port layer, and SMP transport layer.

Table I.12 — Confirmations between link layer, port layer, and SMP transport layer

Link layer	–– Confirmation →	Port layer	–– Confirmation →	Transport layer
SMP	Frame Transmitted	PL	Transmission Status (Frame Transmitted)	MT
	Frame Received		Frame Received	
	Frame Received (SMP Failure)		Frame Received (SMP Failure)	
SL	Open Failed (Wrong Destination)	PL	Transmission Status (Wrong Destination)	MT
	Open Failed (Connection Rate Not Supported)		Transmission Status (Connection Rate Not Supported)	
	Open Failed (Protocol Not Supported)		Transmission Status (Protocol Not Supported)	
	Open Failed (No Destination)		Transmission Status (No Destination)	
	Open Failed (Bad Destination)		Transmission Status (Bad Destination)	
	Open Failed (STP Resources Busy)		Transmission Status (STP Resources Busy)	
	Open Failed (Open Timeout Occurred)		Transmission Status (Open Timeout Occurred)	
	Open Failed (Break Received)		Transmission Status (Break Received)	
	Open Failed (Pathway Blocked)			
	Connection Closed (any reason)		Connection Closed	

I.5 Messages from transport layer to application layer for SSP

Table I.13 lists the requests and responses from the SCSI application layer to the SSP transport layer.

Table I.13 — Requests and responses from SCSI application layer to SSP transport layer

Transport layer	← Request or response ––	Application layer
ST_I (SSP initiator port)	Send SCSI Command	SA (SCSI initiator device)
	Send Task Management Request	
	Accept_Reject OPENs (Accept SSP)	
	Accept_Reject OPENs (Reject SSP)	
ST_T (SSP target port)	Send Command Complete	SA (SCSI target device)
	Task Management Function Executed	
	Send Data-In	
	Receive Data-Out	
	Accept_Reject OPENs (Accept SSP)	
	Accept_Reject OPENs (Reject SSP)	

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table I.14 lists the confirmations and indications from the SSP transport layer to the SCSI application layer.

Table I.14 — Confirmations and indications from SSP transport layer to SCSI application layer

Transport layer	–– Confirmation or indication →	Application layer
ST_I (SSP initiator port)	Command Complete Received	SA (SCSI initiator device)
	Received Task Management Function - Executed	
	Nexus Lost	
	Transport Reset	
ST_T (SSP target port)	SCSI Command Received	SA (SCSI target device)
	Task Management Request Received	
	Data-In Delivered	
	Data-Out Received	
	Nexus Lost	
	Transport Reset	

I.6 Messages from transport layer to application layer for SMP

Table I.15 lists the requests from the management application layer to the SMP transport layer.

Table I.15 — Requests from management application layer to SMP transport layer

Transport layer	← Request ––	Application layer
MT_IP (SMP initiator port)	Send SMP Function Request	MA (SAS initiator device)
MT_TP (SMP target port)	Send SMP Response	MA (SAS target device)
	Accept_Reject OPENs (Accept SMP)	
	Accept_Reject OPENs (Reject SMP)	

Table I.16 lists the confirmations from the SMP transport layer to the management application layer.

Table I.16 — Confirmations from SMP transport layer to management application layer

Transport layer	–– Confirmation →	Application layer
MT_IP (SMP initiator port)	Open Failed	MA (SAS initiator device)
	SMP Frame Receive Timeout	
	SMP Frame Transmit Receive Failure	
	Received SMP Function Complete	
MT_TP (SMP target port)	SMP Function Received	MA (SAS target device)
	SMP Connection Closed	

--`,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex J
(informative)

Discover process example implementation

J.1 Discover process example implementation overview

This annex includes a C program implementing the discover process. Table J.1 describes the source files.

Table J.1 — C program files

Filename	Description
SASDiscoverSimulation.h	header file
SASDiscoverSimulation.cpp	C source file

J.2 Header file

The following is the C header file for the discover process.

```
// SASDiscoverSimulation.h

// assume the maximum number of phys in an expander device is 128
#define MAXIMUM_EXPANDER_PHYS           128

// assume the maximum number of indexes is 128
#define MAXIMUM_EXPANDER_INDEXES        128

// limit to 8 initiators for this example
#define MAXIMUM_INITIATORS              8

// defines for address frame types
#define ADDRESS_IDENTIFY_FRAME          0x00
#define ADDRESS_OPEN_FRAME              0x01

// defines for SMP frame types
#define SMP_REQUEST_FRAME               0x40
#define SMP_RESPONSE_FRAME              0x41

// defines for SMP request functions
#define REPORT_GENERAL                  0x00
#define REPORT_MANUFACTURER_INFORMATION 0x01
#define DISCOVER                        0x10
#define REPORT_PHY_ERROR_LOG            0x11
#define REPORT_PHY_SATA                 0x12
#define REPORT_ROUTE_INFORMATION        0x13
#define CONFIGURE_ROUTE_INFORMATION     0x90
#define PHY_CONTROL                     0x91

// defines for the protocol bits
#define SATA                            0x01
#define SMP                             0x02
#define STP                             0x04
#define SSP                             0x08

enum SMPFunctionResult
{
  SMP_FUNCTION_ACCEPTED = 0,
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  UNKNOWN_SMP_FUNCTION,
  SMP_FUNCTION_FAILED
};

// DeviceTypes
enum DeviceTypes
{
  NO_DEVICE = 0,
  END_DEVICE,
  EDGE_EXPANDER_DEVICE,
  FANOUT_EXPANDER_DEVICE
};

// RoutingAttribute
enum RoutingAttribute
{
  DIRECT = 0,
  SUBTRACTIVE,
  TABLE
};

// RouteFlag
enum DisableRouteEntry
{
  ENABLED = 0,
  DISABLED
};

// PhyLinkRate(s)
enum PhysicalLinkRate
{
  RATE_UNKNOWN = 0,
  PHY_DOES_NOT_EXIST,
  PHY_DISABLED,
  PHY_FAILED,
  SPINUP_HOLD_OOB,
  GBPS_1_5 = 8,
  GBPS_3_0
};

// PhyOperation
enum PhyOperation
{
  NOP = 0,
  LINK_RESET,
  HARD_RESET,
  DISABLE,
  CLEAR_ERROR_LOG = 5,
  CLEAR_AFFILIATION
};

// provide the simple type definitions
typedef unsigned char byte;
typedef unsigned short word;
typedef unsigned long dword;
typedef unsigned _int64 quadword;

// the structures assume a char bitfield is valid, this is compiler
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
// dependent
defines would be more portable, but less descriptive

// the Identify frame is exchanged following OOB, for this
// code it contains the identity information for the attached device
// and the initiator application client
struct Identify
{
  // byte 0
  byte AddressFrame:4;              // ADDRESS_IDENTIFY_FRAME
  byte DeviceType:3;               // END_DEVICE
                                //
  byte RestrictedByte0Bit7:1;

  // byte 1
  byte RestrictedByte1;

  // byte 2
  union
  {
    struct
    {
      byte RestrictedByte2Bit0:1;
      byte SMPInitiator:1;
      byte STPInitiator:1;
      byte SSPInitiator:1;
      byte ReservedByte2Bit4_7:4;
    };
    byte InitiatorBits;
  };

  // byte 3
  union
  {
    struct
    {
      byte RestrictedByte3Bit0:1;
      byte SMPTarget:1;
      byte STPTarget:1;
      byte SSPTarget:1;
      byte ReservedByte3Bit4_7:4;
    };
    byte TargetBits;
  };

  // byte 4-11
  byte RestrictedByte4_11[8];

  // byte 12-19
  quadword SASAddress;

  // byte 20-23
  byte RestrictedByte20_23[4];

  // byte 24-27
  byte ReservedByte24_27[4];

  // byte 28-31
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  dword CRC;
};

// the Open address frame is used to send open requests
struct OpenAddress
{
  // byte 0
  byte AddressFrame:4;            // ADDRESS_OPEN_FRAME
  byte Protocol:3;                // SMP
                                  // STP
                                  // SSP
  byte Initiator:1;

  // byte 1
  byte ConnectionRate:4;          // GBPS_1_5
                                  // GBPS_3_0
  byte Features:4;

  // byte 2-3
  word InitiatorConnectionTag;

  // byte 4-11
  quadword DestinationSASAddress;

  // byte 12-19
  quadword SourceSASAddress;

  // byte 20
  byte CompatibleFeatures;

  // byte 21
  byte PathwayBlockedCount;

  // byte 22-23
  word ArbitrationWaitTime;

  // byte 24-27
  byte MoreCompatibleFeatures[4];

  // byte 28-31
  dword CRC[4];
};

// request specific bytes for a general input function
struct SMPRequestGeneralInput
{
  // byte 4-7
  dword CRC;
};

// request specific bytes for a phy input function
struct SMPRequestPhyInput
{
  // byte 4-7
  byte IgnoredByte4_7[4];

  // byte 8
  byte ReservedByte8;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  // byte 9
  byte PhyIdentifier;

  // byte 10
  byte IgnoredByte10;

  // byte 11
  byte ReservedByte11;

  // byte 12-15
  dword CRC;
};

// the ConfigureRouteInformation structure is used to provide the
// expander route entry for the expander route table, it is intended
// to be referenced by the SMPRequestConfigureRouteInformation struct
struct ConfigureRouteInformation
{
  // byte 12
  byte IgnoredByte12Bit0_6:7;
  byte DisableRouteEntry:1;     // if a routing error is detected
                                // then the route is disabled by
                                // setting this bit

  // byte 13-15
  byte IgnoredByte13_15[3];

  // byte 16-23
  quadword RoutedSASAddress;    // identical to the AttachedSASAddress
                                // found through discovery

  // byte 24-35
  byte IgnoredByte24_35[12];

  // byte 36-39
  byte ReservedByte36_39[4];
};

// request specific bytes for SMP ConfigureRouteInformation function
struct SMPRequestConfigureRouteInformation
{
  // byte 4-5
  byte ReservedByte4_5[2];

  // byte 6-7
  word ExpanderRouteIndex;

  // byte 8
  byte ReservedByte8;

  // byte 9
  byte PhyIdentifier;

  // byte 10-11
  byte ReservedByte10_11[2];

  // byte 12-39
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  struct ConfigureRouteInformation Configure;

  // byte 40-43
  dword CRC;
};

// the PhyControlInformation structure is used to provide the
// expander phy control values, it is intended
// to be referenced by the SMPRequestPhyControl struct
struct PhyControlInformation
{
  // byte 12-31
  byte IgnoredByte12_31[20];

  // byte 32
  byte IgnoredByte32Bit0_3:4;
  byte ProgrammedMinimumPhysicalLinkRate:4;

  // byte 33
  byte IgnoredByte33Bit0_3:4;
  byte ProgrammedMaximumPhysicalLinkRate:4;

  // byte 34-35
  byte IgnoredByte34_35[2];

  // byte 36
  byte PartialPathwayTimeoutValue:4;
  byte ReservedByte36Bit4_7:4;

  // byte 37-39
  byte ReservedByte37_39[3];
};

// request specific bytes for SMP Phy Control function
struct SMPRequestPhyControl
{
  // byte 4-7
  byte IgnoredByte4_7[4];

  // byte 8
  byte ReservedByte8;

  // byte 9
  byte PhyIdentifier;

  // byte 10
  byte PhyOperation;

  // byte 11
  byte ReservedByte11;

  // byte 12-39
  struct PhyControlInformation Control;

  // byte 40-43
  dword CRC;
};
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
// generic structure referencing an SMP Request,
// initialize before use
struct SMPRequest
{
  // byte 0
  byte SMPFrameType;              // always SMP_REQUEST_FRAME

  // byte 1
  byte Function;                  // REPORT_GENERAL
                                  // REPORT_MANUFACTURER_INFORMATION
                                  // DISCOVER
                                  // REPORT_PHY_ERROR_LOG
                                  // REPORT_PHY_SATA
                                  // REPORT_ROUTE_INFORMATION
                                  // CONFIGURE_ROUTE_INFORMATION
                                  // PHY_CONTROL

  // byte 2-3
  byte ReservedByte2_3[2];

  // bytes 4-n
  union
  {
    struct SMPRequestGeneralInput ReportGeneral;
    struct SMPRequestGeneralInput ReportManufacturerInformation;
    struct SMPRequestPhyInput Discover;
    struct SMPRequestPhyInput ReportPhyErrorLog;
    struct SMPRequestPhyInput ReportPhySATA;
    struct SMPRequestPhyInput ReportRouteInformation;
    struct SMPRequestConfigureRouteInformation ConfigureRouteInformation;
    struct SMPRequestPhyControl PhyControl;
  } Request;
};

// request specific bytes for SMP Report General response, intended to be
// referenced by SMPResponse
struct SMPResponseReportGeneral
{
  // byte 4-5
  word ExpanderChangeCount;

  // byte 6-7
  word ExpanderRouteIndexes;

  // byte 8
  byte ReservedByte8;

  // byte 9
  byte NumberOfPhys;

  // byte 10
  byte ConfigurableRouteTable:1;
  byte ReservedByte10Bit1_7:7;

  // byte 11
  byte ReservedByte11;

  // byte 12-15
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  dword CRC;
};

// request specific bytes for SMP Report Manufacturer Information response,
// intended to be referenced by SMPResponse
struct SMPResponseReportManufacturerInformation
{
  // byte 4-7
  byte IgnoredByte4_7[4];

  // byte 8
  byte ReservedByte8;

  // byte 9-10
  byte IgnoredByte9_10[2];

  // byte 11
  byte ReservedByte11;

  // byte 12-19
  byte VendorIdentification[8];

  // byte 20-35
  byte ProductIdentification[16];

  // byte 36-39
  byte ProductRevisionLevel[4];

  // byte 40-59
  byte VendorSpecific[20];

  // byte 60-63
  dword CRC;
};

// the Discover structure is used to retrieve expander port information
// it is intended to be referenced by the SMPResponseDiscover structure
struct Discover
{
  // byte 12
  byte RoutingAttribute:4;
  byte AttachedDeviceType:3;
  byte IgnoredByte12Bit7:1;

  // byte 13
  byte NegotiatedPhysicalLinkRate:4;
  byte ReservedByte13Bit4_7:4;

  // byte 14
  union
  {
    struct
    {
      byte AttachedSATAHost:1;
      byte AttachedSMPInitiator:1;
      byte AttachedSTPInitiator:1;
      byte AttachedSSPInitiator:1;
      byte ReservedByte14Bit4_7:4;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    };
    byte InitiatorBits;
  };

  // byte 15
  union
  {
    struct
    {
      byte AttachedSATADevice:1;
      byte AttachedSMPTarget:1;
      byte AttachedSTPTarget:1;
      byte AttachedSSPTarget:1;
      byte ReservedByte15Bit4_7:4;
    };
    byte TargetBits;
  };

  // byte 16-23
  quadword AttachedSASAddress;

  // byte 24-31
  quadword SASAddress;

  // byte 32
  byte HardwareMinimumPhysicalLinkRate:4;
  byte ProgrammedMinimumPhysicalLinkRate:4;

  // byte 33
  byte HardwareMaximumPhysicalLinkRate:4;
  byte ProgrammedMaximumPhysicalLinkRate:4;

  // byte 34-35
  byte VendorSpecific[2];

  // byte 36
  byte PartialPathwayTimeoutValue:4;
  byte IgnoredByte36Bit4_6:3;
  byte InternalPhy:1;

  // byte 37
  byte ReservedByte37;

  // byte 38          proposal 03-089r0, previously Reserved
  byte PhyChangeCount;

  // byte 39
  byte ReservedByte39;

  // byte 40-43
  dword CRC;
};

// response specific bytes for SMP Discover, intended to be referenced by
// SMPResponse
struct SMPResponseDiscover
{
  // byte 4-7
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  byte IgnoredByte4_7;

  // byte 8
  byte ReservedByte8;

  // byte 9
  byte PhyIdentifier;

  // byte 10-11
  byte ReservedByte10_11;

  // byte 12-43
  struct Discover Result;
};

// response specific bytes for SMP Report Phy Error Log, intended to be
// referenced by SMPResponse
struct SMPResponseReportPhyErrorLog
{
  // byte 4-7
  byte IgnoredByte4_7;

  // byte 8
  byte ReservedByte8;

  // byte 9
  byte PhyIdentifier;

  // byte 10
  byte IgnoredByte10;

  // byte 11
  byte ReservedByte11;

  // byte 12-15
  dword InvalidDwordCount;

  // byte 16-19
  dword DisparityErrorCount;

  // byte 20-23
  dword LossOfDwordSynchronizationCount;

  // byte 24-27
  dword PhyResetProblemCount;

  // byte 28-31
  dword CRC;
};

// this structure describes the Register Device to Host FIS defined in the
// SATA specification
struct RegisterDeviceToHostFIS
{
  // byte 24
  byte FISType;

  // byte 25
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    byte ReservedByte25Bit0_5:6;
    byte Interrupt:1;
    byte ReservedByte25Bit7:1;

    // byte 26
    byte Status;

    // byte 27
    byte Error;

    // byte 28
    byte SectorNumber;

    // byte 29
    byte CylLow;

    // byte 30
    byte CylHigh;

    // byte 31
    byte  DevHead;

    // byte 32
    byte SectorNumberExp;

    // byte 33
    byte CylLowExp;

    // byte 34
    byte CylHighExp;

    // byte 35
    byte ReservedByte35;

    // byte 36
    byte SectorCount;

    // byte 37
    byte SectorCountExp;

    // byte 38-43
    byte ReservedByte38_43[6];
};

// response specific bytes for SMP Report Phy SATA, intended to be
// referenced by SMPResponse
struct SMPResponseReportPhySATA
{
    // byte 4-7
    byte IgnoredByte4_7;

    // byte 8
    byte ReservedByte8;

    // byte 9
    byte PhyIdentifier;

    // byte 10
```

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  byte IgnoredByte10;

  // byte 11
  byte AffilationValid:1;
  byte ReservedByte11Bit1_7:7;

  // byte 12-15
  byte ReservedByte12_15[4];

  // byte 16-32
  quadword STPSASAddress;

  // byte 24-43
  struct RegisterDeviceToHostFIS FIS;

  // byte 44-47
  byte ReservedByte44_47[4];

  // byte 48-55
  quadword AffiliatedSTPInitiatorSASAddress;

  // byte 56-59
  dword CRC;
};

struct ReportRouteInformation
{
  // byte 12
  byte IgnoredByte12Bit0_6:7;
  byte RouteEntryDisabled:1;

  // byte 13-15
  byte IgnoredByte13_15[3];

  // byte 16-23
  quadword RoutedSASAddress;

  // byte 24-35
  byte IgnoredByte24_35[12];

  // byte 36-39
  byte ReservedByte36_39[4];
};

// response specific bytes for SMP Report Route Information, intended to be
// referenced by SMPResponse
struct SMPResponseReportRouteInformation
{
  // byte 4-5
  byte IgnoredByte4_5;

  // byte 6-7
  word ExpanderRouteIndex;

  // byte 8
  byte ReservedByte8;

  // byte 9
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    byte PhyIdentifier;

    // byte 10
    byte IgnoredByte10;

    // byte 11
    byte ReservedByte11;

    // byte 12-39
    struct ReportRouteInformation Result;

    // byte 40-43
    dword CRC;
};

// response specific bytes for SMP Configure Route Information,
// intended to be referenced by SMPResponse
struct SMPResponseConfigureRouteInformation
{
    // byte 4-7
    dword CRC;
};

// response specific bytes for SMP Phy Control,
// intended to be referenced by SMPResponse
struct SMPResponsePhyControl
{
    // byte 4-7
    dword CRC;
};

// generic structure referencing an SMP Response
// initialize before use
struct SMPResponse
{
    // byte 0
    byte SMPFrameType;              // always 41h for SMP responses

    // byte 1
    byte Function;

    // byte 2
    byte FunctionResult;

    // byte 3
    byte ReservedByte3;

    // bytes 4-n
    union
    {
        struct SMPResponseReportGeneral ReportGeneral;
        struct SMPResponseReportManufacturerInformation ReportManufacturerInformation;
        struct SMPResponseDiscover Discover;
        struct SMPResponseReportPhyErrorLog ReportPhyErrorLog;
        struct SMPResponseReportPhySATA ReportPhySATA;
        struct SMPResponseReportRouteInformation ReportRouteInformation;
        struct SMPResponseConfigureRouteInformation ConfigureRouteInformation;
        struct SMPResponsePhyControl PhyControl;
```

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  } Response;

};

// this structure is how this simulation obtains its knowledge about the
// initiator port that is doing the discover, it is not defined as part of
// the standard...
struct ApplicationClientKnowledge
{
  quadword SASAddress;
  byte NumberOfPhys;
  byte InitiatorBits;
  byte TargetBits;
};

// the TopologyTable structure is the summary of the information gathered
// during the discover process, the table presented here is not concerned
// about memory resources consumed, production code would be more concerned
// about specifying necessary elements explictly
struct TopologyTable
{
  struct TopologyTable *Next;

  // information from REPORT_GENERAL
  struct SMPResponseReportGeneral Device;

  // information from DISCOVER
  struct SMPResponseDiscover Phy[MAXIMUM_EXPANDER_PHYS];

  // route index for the configuration chain functions
  word RouteIndex[MAXIMUM_EXPANDER_PHYS];

  //
  // in production code there would also be links to the necessary device
  // information like end device; vendor, model, serial number, etc.
  // the gathering of that type of information is not done here...
  //
};
```

J.3 Source file

The following is the C source file for the discover process.

```
// SASDiscoverSimulation.cpp
//
// this is a simple simulation and code implementation of the initiator
// based expander discovery and configuration

// there is no attempt to handle phy errors, arbitration issues, etc.
// production level implementation would have to handle errors appropriately

// structure names used are equivalent to those referenced in the
// SAS document

// basic assumptions
//
// 1. change primitives initate a rediscovery/configuration sequence
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
// 2. table locations for SASAddresses are deterministic for a specific
//    topology only, when the topology changes, the location of a SASAddress
//    in an ASIC table cannot be assumed
// 3. a complete discovery level occurs before the configuration of the
//    level begins, multiple passes are required as the levels of expanders
//    encountered between the intiator and the end devices is increased
// 4. configuration of a single expander occurs before proceeding to
//    subsequent expanders attached
// 5. the Attached structure is filled in following OOB and is available
//    from the initialization routines
// 6. the Iam structure is provide by the application client

#include <malloc.h>
#include <memory.h>

// include the SAS structures
#include “SASDiscoverSimulation.h”

// limit to 4 levels of expanders
#define ADDRESS_CHAIN_LEVELS          4

// used to limit the redundant configuration cycles
#define MAXIMUM_ROUTE_ENTRIES (ADDRESS_CHAIN_LEVELS * MAXIMUM_EXPANDER_PHYS)

// loaded by the application client, in this simulation it is provided
// in a text file, SAS-02-359Example.ini
struct ApplicationClientKnowledge Iam[MAXIMUM_INITIATORS] = { 0 };

// obtained following OOB from the attached phy, in this simulation
// it is provided in a text file, SAS-02-359Example.ini
struct Identify Attached[MAXIMUM_INITIATORS] = { 0 };

// buffers used to request and return SMP data
struct SMPRequest SMPRequestFrame = { 0 };
struct SMPResponse SMPResponseFrame = { 0 };

// resulting discover information ends up in this table
struct TopologyTable *SASDomain[MAXIMUM_INITIATORS] = { 0 };

// this is the function used to send an SMPRequest and get a response back
extern byte SMPRequest(byte PhyIdentifier,
                       quadword Source,
                       quadword Destination,
                       struct SMPRequest *SMPRequestFrame,
                       struct SMPResponse *SMPResponseFrame,
                       byte Function,
                       ...);

// this is the configuration chain it contains the list of expander between
// the programming initiator and the discover extent, each expander in the
// chain is configured when a new SASaddress is encounter from the top
// of the chain to the bottom
static struct ConfigurationChain
{
   quadword SASAddress;
   struct TopologyTable *Expander;
   byte PhyIdentifier;
} TableChain[ADDRESS_CHAIN_LEVELS] = { 0 };
```

--`,,`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
static int ChainIndex = 0;
static int ChainEntry = 0;

// this is the upstream chain and contains the list of expanders that have
// already been traversed upstream, this reduces the number of times we
// go through the configuration
static quadword UpstreamChain[ADDRESS_CHAIN_LEVELS] = { 0 };

// this the route entry chain it contains the list of route entries that
// have already been configured, this reduces duplication of configuration
// requests
static struct RouteEntryChain
{
   quadword SASAddress;
   word RouteIndex;
   byte PhyIdentifier;
} ConfiguredChain[MAXIMUM_ROUTE_ENTRIES] = { 0 };

// this function gets the report general and discover information for
// a specific expander
struct TopologyTable *DiscoverExpander(byte PhyIdentifier,
                                       quadword SourceSASAddress,
                                       quadword DestinationSASAddress,
                                       struct TopologyTable *Expander)
{
   struct TopologyTable *expander;
   byte phyCount;
   int error = 1;

   // allocate space to retrieve the expander information
   expander = (struct TopologyTable *)
              calloc(1,
                     sizeof(struct TopologyTable));

   // make sure we only do this if the allocation is successful
   if(expander)
   {
      // get the report general information for the expander
      SMPRequest(PhyIdentifier,
                 SourceSASAddress,
                 DestinationSASAddress,
                 &SMPRequestFrame,
                 &SMPResponseFrame,
                 REPORT_GENERAL);

      // don't worry about too much in the 'else' case for this example
      if(SMPResponseFrame.FunctionResult == SMP_FUNCTION_ACCEPTED)
      {
         // copy the result into the topology table
         memcpy((void *)&(expander->Device),
                (void *)&SMPResponseFrame.Response.ReportGeneral,
                sizeof(struct SMPResponseReportGeneral));

         if(expander->Device.NumberOfPhys <= MAXIMUM_EXPANDER_PHYS)
         {
            // now walk through all the phys of the expander
            for(phyCount = 0;
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                (phyCount < expander->Device.NumberOfPhys);
                phyCount++)
            {
               // get the discover information for each phy
               SMPRequest(PhyIdentifier,
                          SourceSASAddress,
                          DestinationSASAddress,
                          &SMPRequestFrame,
                          &SMPResponseFrame,
                          DISCOVER,
                          phyCount);

               // don't worry about the 'else' case for this example
               if(SMPResponseFrame.FunctionResult == SMP_FUNCTION_ACCEPTED)
               {
                  // clear the error flag
                  error = 0;

                  // copy the result into the topology table
                  memcpy((void *)&(expander->Phy[phyCount]),
                         (void *)&SMPResponseFrame.Response.Discover,
                         sizeof(struct SMPResponseDiscover));
               }
               else
               {
                  // something happened so just bailout on this expander
                  error = 1;

                  // release the memory we allocated for this...
                  free(expander);
                  break;
               }
            }
         }
      }

      // if we did not have an error, then link in the information from
      // this expander into the topology table
      if(!error &&
         Expander)
      {
         Expander->Next = expander;
      }
   }

   // the expander pointer is the error return, a null indicates something
   // bad happened...
   return(expander);
}

// this routine adds a SASAddress to the configuration chain
void PushSASAddress(quadword SASAddress,
                    struct TopologyTable *Expander,
                    byte PhyIdentifier)
{
   TableChain[ChainIndex].SASAddress = SASAddress;
   TableChain[ChainIndex].Expander = Expander;
   TableChain[ChainIndex].PhyIdentifier = PhyIdentifier;
```

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   if(ChainIndex < ADDRESS_CHAIN_LEVELS)
   {
      ChainIndex++;
   }
}

// this routine removes a SASAddress from the configuration chain
void PopSASAddress(void)
{
   if(ChainIndex)
   {
      ChainIndex--;
   }

   TableChain[ChainIndex].SASAddress = 0;
   TableChain[ChainIndex].Expander = 0;
   TableChain[ChainIndex].PhyIdentifier = 0;
}

// this routine resets the ChainEntry and return the first entry on
// the Chain
quadword FirstSASAddress(void)
{
   ChainEntry = 0;

   return(TableChain[ChainEntry].SASAddress);
}

// this routine returns the next entry on the Chain
quadword NextSASAddress(void)
{
   if(ChainEntry < ChainIndex)
   {
      ChainEntry++;
   }

   return(TableChain[ChainEntry].SASAddress);
}

// this routine gets the route index from the expander structure
word IndexForSASAddress(void)
{
   struct TopologyTable *expander = TableChain[ChainEntry].Expander;
   word index = 0;

   index = expander->RouteIndex[TableChain[ChainEntry].PhyIdentifier]++;

   return(index);
}

// this routine gets the phy identifier from the configuration chain
byte PhyForSASAddress(void)
{
   return(TableChain[ChainEntry].PhyIdentifier);
}

// this routine keeps track of which SASAddresses have already been
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
// traversed upstream to avoid duplicate configuration passes
byte AlreadyTraversed(quadword SASAddress)
{
   byte chain = 0;
   byte itsHere = 0;

   while(UpstreamChain[chain])
   {
      if(UpstreamChain[chain] == SASAddress)
      {
         itsHere = 1;
         break;
      }

      if(chain < ADDRESS_CHAIN_LEVELS)
      {
         chain++;
      }
   }

   UpstreamChain[chain] = SASAddress;

   return(itsHere);
}

// this routine keeps track of which expanders have been configured to
// avoid duplicate configuration requests
byte AlreadyConfigured(quadword SASAddress,
                       word RouteIndex,
                       byte PhyIdentifier)
{
   word chain = 0;
   byte itsHere = 0;

   while(ConfiguredChain[chain].SASAddress)
   {
      if((ConfiguredChain[chain].SASAddress == SASAddress) &&
         (ConfiguredChain[chain].RouteIndex == RouteIndex) &&
         (ConfiguredChain[chain].PhyIdentifier == PhyIdentifier))
      {
         itsHere = 1;
         break;
      }

      if(chain < MAXIMUM_ROUTE_ENTRIES)
      {
         chain++;
      }
   }

   ConfiguredChain[chain].SASAddress = SASAddress;
   ConfiguredChain[chain].RouteIndex = RouteIndex;
   ConfiguredChain[chain].PhyIdentifier = PhyIdentifier;

   return(itsHere);
}

// this function is recursive, it discovers then configures as necessary
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
// the expanders it finds that are “downstream”
struct TopologyTable *DiscoverAndConfigure(byte PhyIdentifier,
                                           quadword SourceSASAddress,
                                           struct TopologyTable *Start,
                                           byte Upstream)
{
   struct TopologyTable *expander = Start;
   struct TopologyTable *nextExpander;
   struct TopologyTable *attachedDeviceSet = 0;

   struct SMPResponseDiscover *discover;
   struct SMPResponseDiscover *configure;

   quadword sasAddress;
   byte phyCount;
   word indexOffset;
   byte phyOffset;

   // if the upstream flag is set, then we’re moving upstream looking for
   // the boundary of the device set
   if(Upstream)
   {
      // walk through all the phys of the expander
      for(phyCount = 0;
          (phyCount < expander->Device.NumberOfPhys);
          phyCount++)
      {
         // this is just a pointer helper
         discover = &(expander->Phy[phyCount]);

         // look for phys with edge or fanout devices attached...
         if(((discover->Result.RoutingAttribute == SUBTRACTIVE) &&
             (discover->Result.AttachedDeviceType == EDGE_EXPANDER_DEVICE))
||
            (discover->Result.AttachedDeviceType == FANOUT_EXPANDER_DEVICE))
         {
            if(!AlreadyTraversed(discover->Result.AttachedSASAddress))
            {
               nextExpander
               = DiscoverExpander(PhyIdentifier,
                                  SourceSASAddress,
                                  discover->Result.AttachedSASAddress,
                                  expander);

               // if we successfully got the information from the next
               // expander then link it in the topology table...
               if(nextExpander)
               {
                  // if the attached device has a subtractive phy, then
                  // stop going upstream, we have two expander device sets
                  // connected without a fanout expander, save the address
                  // of the attachedDeviceSet, and do the configuration of
                  // it as a “separate domain”...
                  if(nextExpander->Phy[phyCount].Result.RoutingAttribute
                     == SUBTRACTIVE)
                  {
                     attachedDeviceSet = nextExpander;
                  }
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                  // the attached device does not have a subtractive phy,
                  // continue to move upstream until we hit the edge of the
                  // device set...
                  else
                  {
                     // go upstream to the next expander
                     attachedDeviceSet = DiscoverAndConfigure
                                         (PhyIdentifier,
                                          SourceSASAddress,
                                          nextExpander,
                                          1);

                     expander->Next = nextExpander;
                  }
               }
            }
         }
      }
   }

   // this is a maximal decent traversal with a configuration stage
   // at each transition to a new level, if a configuration is required
   // by the expander

   // walk through all the phys of the expander looking for table routes
   // only, we'll do configuration as we find them...
   for(phyCount = 0;
       (phyCount < expander->Device.NumberOfPhys);
       phyCount++)
   {
      // this is just a pointer helper
      discover = &(expander->Phy[phyCount]);

      // look for phys with edge or fanout devices attached...
      if((discover->Result.RoutingAttribute == TABLE) &&
         ((discover->Result.AttachedDeviceType == EDGE_EXPANDER_DEVICE) ||
          (discover->Result.AttachedDeviceType == FANOUT_EXPANDER_DEVICE)))
      {
         // add the expander to the configuration chain
         PushSASAddress(discover->Result.SASAddress,
                        expander,
                        phyCount);

         nextExpander = DiscoverExpander
                        (PhyIdentifier,
                         SourceSASAddress,
                         discover->Result.AttachedSASAddress,
                         expander);

         // if we successfully got the information from the next expander
         // then link it in the topology table...
         if(nextExpander)
         {
            byte nextPhyCount = 0;

            // loop through the phys of the nextExpander, gathering
            // the SASAddresses for this expander
            for(nextPhyCount = 0;
```

--`,,,`,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                (nextPhyCount < nextExpander->Device.NumberOfPhys);
                nextPhyCount++)
            {
               // configure the expanders between the initiator and
               // the destination
               for(sasAddress = FirstSASAddress();
                   sasAddress;
                   sasAddress = NextSASAddress())
               {
                  word routeIndex = IndexForSASAddress();
                  byte phyIdentifier = PhyForSASAddress();

                  // this avoids duplicate configure cycles as we walk the
                  // topology
                  if(!AlreadyConfigured(sasAddress,
                                        routeIndex,
                                        phyIdentifier))
                  {
                     struct SMPResponseDiscover *nextDiscover;

                     // this is just a helper pointer to reduce line length
                     nextDiscover = &(nextExpander->Phy[nextPhyCount]);

                     // configure the indexes for the nextExpander in the
                     // AttachedExpander
                     SMPRequest(PhyIdentifier,
                                SourceSASAddress,
                                sasAddress,
                                &SMPRequestFrame,
                                &SMPResponseFrame,
                                CONFIGURE_ROUTE_INFORMATION,
                                routeIndex,
                                phyIdentifier,
                                0,
                                nextDiscover->Result.AttachedSASAddress);
                  }
               }
            }

            // descend to the next expander
            DiscoverAndConfigure(PhyIdentifier,
                                 SourceSASAddress,
                                 nextExpander,
                                 0);

            // remove the last expander from the configuration chain
            PopSASAddress();

            expander->Next = nextExpander;
         }
      }
   }

   return(attachedDeviceSet);
}

// this routine appends the leaf to the tree domain
void ConcatenateDomains(struct TopologyTable *Tree,
```

--`,,,`,,,-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                          struct TopologyTable *Leaf)
{
   while(Tree)
   {
      if(Tree->Next == 0)
      {
         Tree->Next = Leaf;
         break;
      }

      Tree = Tree->Next;
   }
}

// this routine clears the state variables that control execution, this
// allows recursion to work, and allows the different initiator passes
// to work...
void InitializeForRecursion(void)
{
   word chain = 0;

   for(chain = 0;
       chain < MAXIMUM_ROUTE_ENTRIES;
       chain++)
   {
      if(chain < ADDRESS_CHAIN_LEVELS)
      {
         TableChain[chain].SASAddress = 0;
         TableChain[chain].Expander = 0;
         TableChain[chain].PhyIdentifier = 0;

         UpstreamChain[chain] = 0;
      }

      ConfiguredChain[chain].SASAddress = 0;
      ConfiguredChain[chain].RouteIndex = 0;
      ConfiguredChain[chain].PhyIdentifier = 0;
   }

   ChainIndex = 0;
   ChainEntry = 0;
}

// the application client for the initiator device would make a call to
// this function to begin the discover process...
// to simplify the setup for the simulation, the DiscoverProcess gets
// the Initiator number to allow multiple initiators...
void DiscoverProcess(byte Initiator)
{
   // clear the state variables
   InitializeForRecursion();

   // check to see if an expander is attached

   // this example ignores the other phys on initiators with multiple phys,
   // the results would be the same, just the determination of wide ports
   // and independent SASDomains would need to be handled...
   if((Attached[Initiator].DeviceType == EDGE_EXPANDER_DEVICE) ||
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      (Attached[Initiator].DeviceType == FANOUT_EXPANDER_DEVICE))
   {
      // expander is attached, so begin to walk the topology, building
      // the necessary tables for each expander
      SASDomain[Initiator] = DiscoverExpander
                             (0,
                              Iam[Initiator].SASAddress,
                              Attached[Initiator].SASAddress,
                              0);
   }

   // make sure we got the information from the attached expander before
   // going on...
   if(SASDomain[Initiator])
   {
      struct TopologyTable *attachedDeviceSet;

      // go upstream on the subtractive phys until we discover that we are
      // attached to another subtractive phy or a fanout expander
      // then begin the discover process from that point, this works
      // because any new address that we find naturally moves upstream
      // due to the subtractive addressing method
      // if during the discover and configuration cycle, it is determined
      // that there are two device sets connected, then a second discover
      // and configuration cycle is required for the other device set
      attachedDeviceSet = DiscoverAndConfigure(0,
                                               Iam[Initiator].SASAddress,
                                               SASDomain[Initiator],
                                               1);

      // if two device sets are connected, then the attached device set
      // has to be discovered and configured separately, but there is no
      // need to do the upstream check...
      if(attachedDeviceSet)
      {
         DiscoverAndConfigure(0,
                              Iam[Initiator].SASAddress,
                              attachedDeviceSet,
                              0);

         // put the domains together
         ConcatenateDomains(SASDomain[Initiator],
                            attachedDeviceSet);
      }
   }
}
```

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex K
(informative)

SAS icon

The SAS icon is shown in figure K.1. This icon should be included on all connectors to devices compliant with this standard.

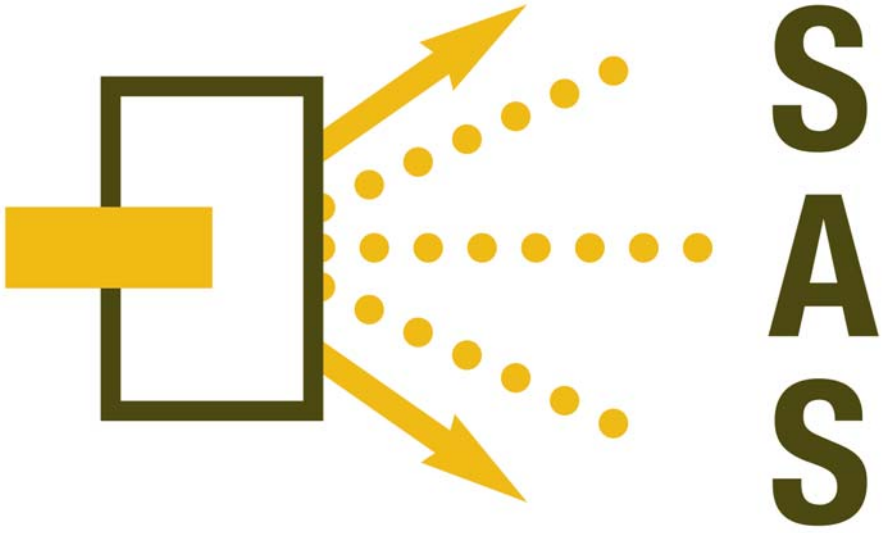


Figure K.1 — SAS icon

NOTE 35 - Contact the SCSI Trade Association at http://www.scsita.org for versions of the SAS icon in various graphics formats.

--`,,`,,-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISBN 2-8318-7742-3



ICS 35.200

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND

Copyright International Organization for Standardization
Reproduced by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale