To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 6 November 2006
Subject: 05-381r7 SAS-2 Multiplexing

Revision history

Revision 0 (2 November 2005) First revision

Revision 1 (23 December 2005) Incorporated feedback from the December 2006 SAS protocol WG teleconference - removed the algorithm to decide whether or not to multiplex a link (left as vendor-specific), added more non-multiplexing specific changes to support 6 Gbps throughout all but the physical layer chapter.

Revision 2 (4 March 2006) Incorporated comments from January 2006 SAS protocol WG. Changed references to “ALIGNs/NOTIFYs” and its many variations into “deletable primitives”, which are defined as ALIGN, NOTIFY, and MUX.

Revision 3 (2 May 2006) Incorporated comments from March 2006 SAS protocol WG. Included global changes (where appropriate) of expander phy -> expander logical phy and SAS phy -> SAS logical phy. This revision does not include changes to the SL_IRM state machine to perform the multiplexing sequence.

Revision 4 (31 August 2006) Incorporated comments from July 2006 SAS protocol WG, including adding a REASON field to the IDENTIFY address frame. Included changes to SL_IRM state machines. Based on feedback from the SCSI Trade Association, dropped support for four-way multiplexing.

Revision 5 (7 September 2006) Incorporated comments from 5 September 2005 multiplexing teleconference, including dropping START_LOGICAL_LINK and including an alternate multiplexing negotiation method based on SNW-3 (labeled Option B). Moved the IDENTIFY address frame REASON field to a separate proposal 06-409 since it is useful even without multiplexing.

Revision 6 (23 October 2006) Incorporated comments from September 2006 SAS protocol WG - kept Option B only (SNW-3 based multiplexing negotiation). In the DISCOVER response and elsewhere, defined rates Bh - Fh as “reserved for future rates” rather than just “reserved” - application clients need to handle them as valid link rates, rather than treat them as errors. Application clients need to be prepared to be attached to future generation expanders. Changed the DISCOVER response NEGOTIATED PHYSICAL LINK RATE field to a NEGOTIATED LOGICAL LINK RATE field to provide better compatibility with SAS-1.1 application clients. Added a new NEGOTIATED PHYSICAL LINK RATE field in a different location for SAS-2 application clients to use. Added an ATTACHED REASON field to the DISCOVER response (should have been in 06-409).

Revision 6 (6 November 2006) Incorporated comments from November 2006 SAS protocol WG.

Related documents

sas2r02 - Serial Attached SCSI - 2 (SAS-2) revision 2
06-409r0 - SAS-2 IDENTIFY address frame REASON field (Rob Elliott, HP)
06-363r2 - SAS-2 SNW-3 bit definitions (Rob Elliott, HP)
06-324r1 - SAS-2 Modifications to speed negotiation (Amr Wasssal & Robert Watson, PMC-Sierra)

Overview

When Serial Attached SCSI was first conceived, it included the concept of time division multiplexing a physical link into two logical links when a 3 Gbps HBA is talking to multiple (SATA) 1.5 Gbps disk drives. This feature was removed before submittal to T10 to reduce protocol complexity. If a 3 Gbps HBA talks to a 1.5 Gbps disk drive, rate matching is used - deletable primitives (ALIGN/NOTIFYs) are inserted every other dword and half the bandwidth on the 3 Gbps is wasted.

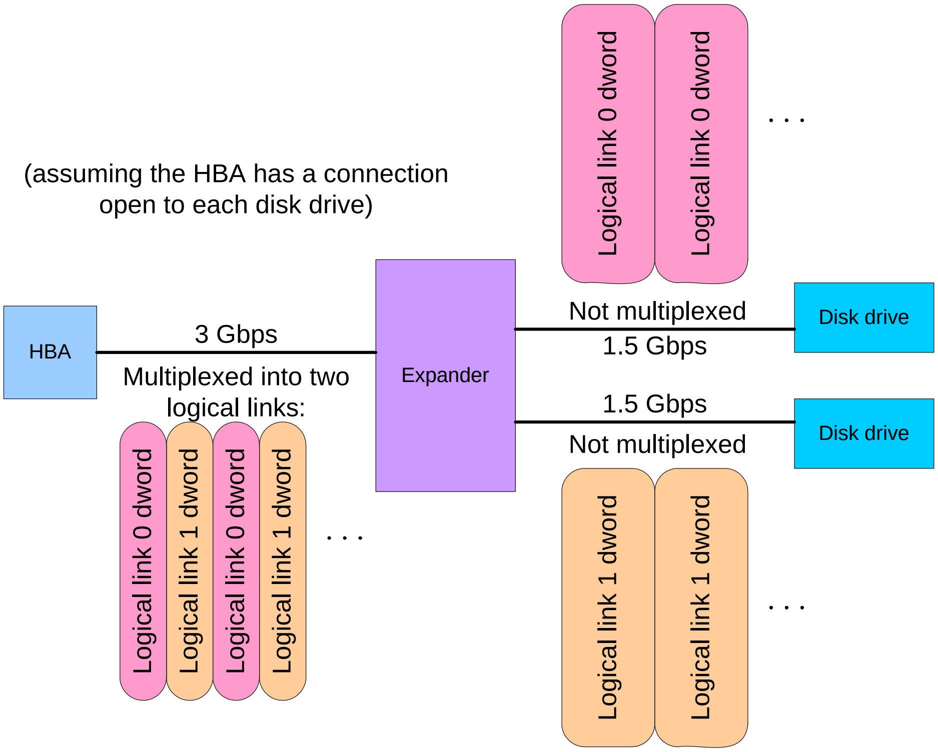


Figure 1 — Multiplexing overview

Multiplexing reclaims that bandwidth by transmitting dwords for another connection rather than ALIGN/NOTIFYs.

Key points of the proposal:

a) Multiplexing is optional and can be done by any phy faster than 1.5 Gbps
b) Multiplexing can be done by both end devices (both initiators and targets) and expander devices
c) Static multiplexing (not dynamic)
   A) One-way multiplexing (means no multiplexing)
   B) Two-way multiplexing (e.g. a 3 Gbps physical link into two 1.5 Gbps logical links, or a 6 Gbps physical link into two 3 Gbps logical links)
   C) No nesting (e.g. no 6 Gbps physical link to one 3 Gbps and two 1.5 Gbps logical links)
d) Define new MUX (LOGICAL LINK 0)/(LOGICAL LINK 1) primitives
   A) Periodically resend MUX to confirm logical link numbers (mainly for logic analyzers)
   B) MUXing happens outside elasticity buffers, so no ALIGN/NOTIFY during MUX exchange
e) Negotiate multiplexing during SNW-3 (suggested by Jeff Gauvin, LSI Logic)
   A) SNW-3 exchanges REQUESTED LOGICAL LINK RATE fields
   B) If both sides agree, start multiplexing immediately
   C) Transmit an IDENTIFY address frame on each logical link
   D) Transmit MUX before each IDENTIFY address frame to identify its logical link number
f) Rerun link reset sequence if multiplexing level needs to be changed
g) Rerun link reset sequence on loss of dword synchronization (since which dword belongs to which logical link is uncertain)
h) No modifications to the SL_CC state machine; it just works on logical phys rather than physical phys
i) Clock skew management
   A) Multiplex after clock skew management ALIGN/NOTIFY insertion
   B) Demultiplex before the elasticity buffers
   C) Multiplex every other dword, not every other non-ALIGN/NOTIFY
   D) ALIGN/NOTIFY frequency within each logical link must equal that of a physical link at the same rate

j) SMP functions
   A) In PHY CONTROL, provide field to specify how many logical links a phy should request in the IDENTIFY address frame
   B) In DISCOVER, report current multiplexing status (enabled/disabled), outgoing IDENTIFY content, incoming IDENTIFY content

k) The discover process algorithm to decide whether or not to request multiplexing on a physical link is left vendor-specific. There is no standard way to resolve whether high-speed targets should have priority to make high-speed connections vs. optimizing for more lower-speed targets.

Additionally, non-multiplexing specific changes are included to support 6 Gbps throughout the protocol layers.

Alternatives

One alternative is to let the expander handle everything by buffering; e.g., let HBAs speak to expanders at 6 Gbps, expanders speak to drives at 3 Gbps, and have the expander terminate the connections on each side and store-and-forward multiple frames accumulated during the connection. This is very complicated for the expander and may require protocol changes to optimize performance (see 06-422 and 06-386 for such a proposal). The SAS connection-based fabric is not well suited for a packet-switched approach - telephone networks do this type of conversion, but voice connections have low bandwidth requirements.

Another alternative is a dynamic rather than static multiplexing scheme. This would let connections with different connection rates share the same physical link without requiring redoing the link reset sequence. After an OPEN (3 Gbps) is sent, every other dword is available to carry another connection, not just rate matching ALIGN/NOTIFYs. Problems that would have to be solved with a dynamic scheme include:

a) maintain proper ALIGN/NOTIFY insertion rates
b) keep the ability to do rate matching within a 3 Gbps connection
c) avoid starvation of 6 Gbps connection requests by 3 Gbps connections that keep slipping in

Suggested changes to SAS-2

Changes to chapter 3 (definitions)

Add logical phy and logical link terms.

3.1 Definitions

3.1.1 attached SAS address: The SAS address (see 3.1.165) of the attached phy (e.g., received by a ~~physical~~logical phy in the incoming IDENTIFY address frame during the initialization sequence (see 4.1.2)), or the SAS address of the STP target port in an STP/SATA bridge (see 4.6.2).

3.1.2 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.

3.1.3 deletable primitive: An ALIGN (see 7.2.5.2), NOTIFY (see 7.2.5.9), or MUX (see 7.2.5.n), which may be deleted by a receiver instead of being placed into its elasticity buffer. See 7.3.

3.1.4 expander logical phy: An expander phy (see 3.1.5) or a multiplexed portion of an expander phy (see 3.1.5). See 4.x.

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

3.1.6 logical link: A physical link (see 3.1.12) or a multiplexed portion of a physical link. See 4.x.

3.1.xxx logical link rate: A physical link rate established as a result of speed negotiation and multiplexing negotiation between two phys.

3.1.7 logical phy: A phy (see 3.1.11) or a multiplexed portion of a phy. See 4.x.

3.1.8 multiplexing: Dividing a physical link into two logical links, and dividing a phy into two logical phys, by interleaving dwords. See 4.xx.

3.1.129 negotiated physical link rate: The current operational physical link rate~~ established as a result of speed negotiation between two phys~~ (see 3.1.xxx).

3.1.9 partial pathway: The set of ~~physical~~logical links participating in a connection request that have not yet conveyed a connection response. See 4.1.9.

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

3.1.11 phy: A object in a device that is used to interface to other devices (e.g., an expander phy (see 3.1.5) or a SAS phy (see 3.1.19)). See 4.1.2.

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

3.1.xxx physical link rate: A physical link rate established as a result of speed negotiation between two phys.

3.1.13 physical phy: A phy (see 3.1.11) that contains a transceiver (see 3.1.241) and electrically interfaces to a physical link to communicate with another physical phy. See 4.1.2.

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

3.1.15 rate: Data transfer rate of a physical link (e.g., 1,5 Gbps~~ or~~, 3~~,0~~ Gbps, or 6 Gbps).

3.1.16 SAS initiator phy: A logical phy (see 3.1.7) in a SAS initiator device.

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

3.1.18 SAS logical phy: A SAS phy or a multiplexed portion of a SAS phy (see 3.1.17). See 4.x.

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

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

3.1.21 SAS target phy: A logical phy (see 3.1.7) in a SAS target device.

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

3.1.23 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.

3.1.24 SMP initiator phy: A SAS initiator phy (see 3.1.16) in an SMP initiator port.

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

3.1.26 SMP phy: A SAS logical phy (see 3.1.18) in an SMP port.

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

3.1.28 SMP target phy: A SAS target phy (see 3.1.21) in an SMP target port.

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

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

3.1.31 SSP initiator phy: A SAS initiator phy (see 3.1.16) in an SSP initiator port.

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

3.1.33 SSP phy: A SAS logical phy (see 3.1.18) in an SSP port.

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

3.1.35 SSP target phy: A SAS target phy (see 3.1.21) in an SSP target port.

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

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

3.1.38 STP initiator phy: A SAS initiator phy (see 3.1.16) in an STP initiator port.

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

3.1.40 STP phy: A SAS logical phy (see 3.1.18) in an STP port.

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

3.1.42 STP primitive: A primitive used only inside STP connections and on SATA physical links. See 7.2.2.

3.1.43 STP target phy: A SAS target phy (see 3.1.21) in an STP target port.

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

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

3.1.46 unit interval (UI): The normalized, dimensionless, nominal duration of a signal transmission bit (e.g., 666,$\overline{6}$ ps at 1,5 Gbps~~ and~~, 333,$\overline{3}$ ps at 3~~,0~~ Gbps, and 166,6 ps at 6 Gbps). Unit interval is a measure of time that has been normalized such that 1 UI is equal to 1/baud seconds.

3.1.47 virtual phy: A phy (see 3.1.11) that interfaces with a vendor-specific interface to another virtual phy inside the same device. See 4.1.2.

3.2 Symbols and abbreviations

See 2.1 for abbreviations of standards bodies (e.g., ISO). Units and abbreviations used in this standard:

Abbreviation	Meaning
AA	ATA application layer (see 10.3)
A.C.	alternating current
...	...
G1	generation 1 physical link rate (1,5 Gbps)
G2	generation 2 physical link rate (3~~,0~~ Gbps)
G3	generation 3 physical link rate (6 Gbps)

Abbreviation	Meaning
G~~3~~4	generation ~~3~~4 physical link rate (defined in a future version of this standard)
Gbps	gigabits per second (10^9 bits per second)
Gen1i	SATA generation 1 physical link rate (1,5 Gbps)(see SATAII-PHY)
Gen1x	SATA generation 1 physical link rate (1,5 Gbps), extended length (see SATAII-PHY)
Gen2i	SATA generation 2 physical link rate (3~~,0~~ Gbps)(see SATAII-PHY)
Gen2x	SATA generation 2 physical link rate (3~~,0~~ Gbps), extended length (see SATAII-PHY)
...	...
SL_IR	link layer identification, and hard reset state machines (see 7.9.5)
...	...

Changes to chapter 4 (General)

Add the concept of logical phys and logical links to the model.

4 General

4.1 Architecture

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).

Figure 2 shows two phys attached with a physical link.

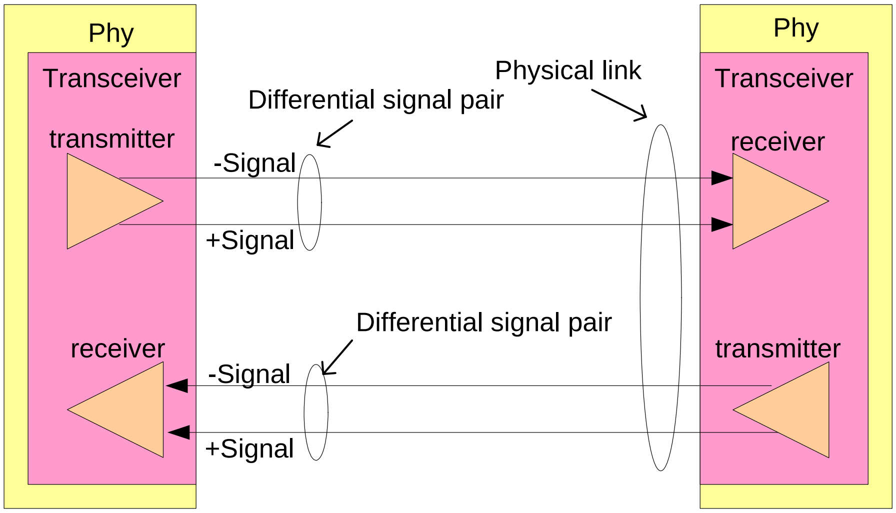


Figure 2 — Physical links and phys

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

A device (i.e., a SAS device (see 4.1.4) or expander device (see 4.1.5)) contains one or more phys.

Each phy has:

a) a SAS address (see 4.2.2), inherited from the SAS port (see 4.1.3) or expander device;

b) a phy identifier (see 4.2.7) which is unique within the device;
c) optionally, support for being an SSP initiator phy;
d) optionally, support for being an STP initiator phy;
e) optionally, support for being an SMP initiator phy;
f) optionally, support for being an SSP target phy;
g) optionally, support for being an STP target phy; and
h) optionally, support for being an SMP target phy.

A phy may be used as one or two logical phys based on multiplexing (see 7.xx).

During the identification sequence (see 7.9), a logical phy:

a) transmits an IDENTIFY address frame including the device type (i.e., end device, edge expander device, or fanout expander device) of the device containing the logical phy, the SAS address of the SAS port or expander device containing the logical phy, phy identifier, SSP initiator phy capability, STP initiator phy capability, SMP initiator phy capability, SSP target phy capability, STP target phy capability, and SMP target phy capability.
b) receives an IDENTIFY address frame containing the same set of information from the attached logical phy, including the attached device type, attached SAS address, attached phy identifier, attached SSP initiator phy capability, attached STP initiator phy capability, attached SMP initiator phy capability, attached SSP target phy capability, attached STP target phy capability, and attached SMP target phy capability.

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~~LOGICAL 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 10.4.3.11), and Phy Control and Discover subpage (see 10.2.7.2.3). 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 3 defines the phy classes, showing the relationships between the following classes:

a) phy;
b) SAS phy;
c) expander phy;
d) SAS initiator phy;
e) SAS target phy;
f) SSP phy;
g) STP phy; and
h) SMP phy.

SATA phys are also referenced in this standard but are defined by SATA (see ATA/ATAPI-7 V3).

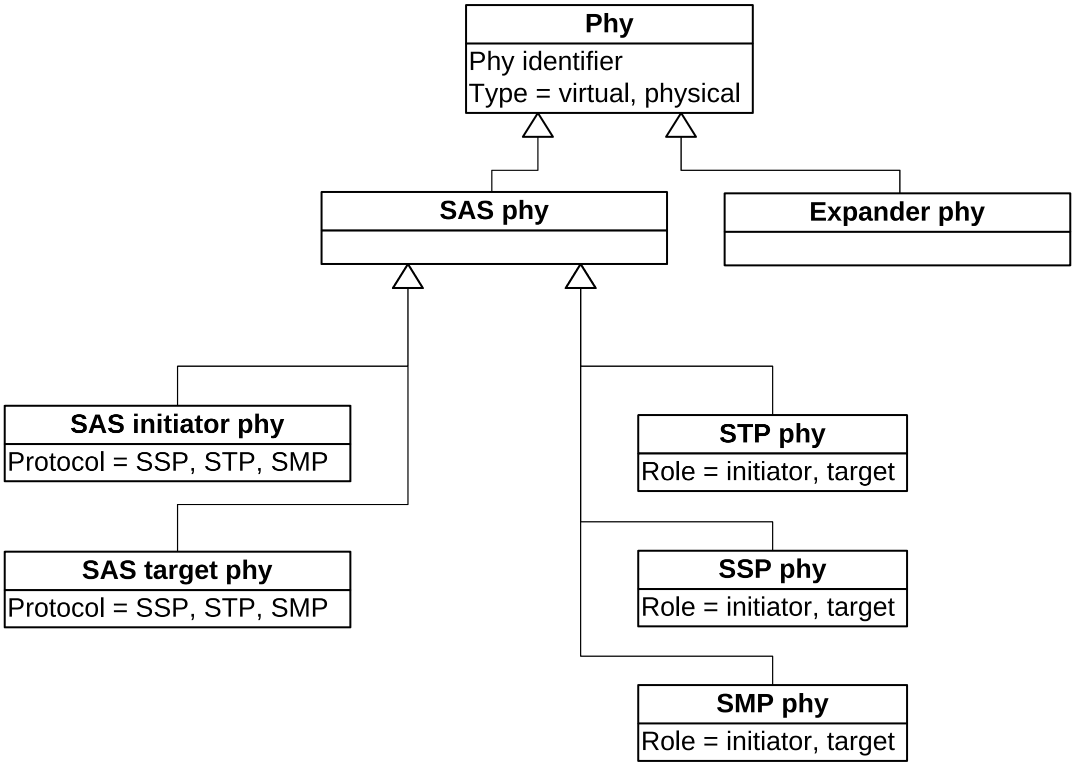


Figure 3 — Phy class diagram

Editor’s Note 1: Logical phys (SAS logical phys and expander logical phys) need to be included in the UML model.

Figure 4 shows the objects instantiated from the phy classes, including:

a) from the SAS phy class:
   A) SSP initiator phy;
   B) SSP target phy,
   C) virtual SSP initiator phy;
   D) virtual SSP target phy;
   E) STP initiator phy;
   F) STP target phy;
   G) virtual STP initiator phy;
   H) virtual STP target phy;
   I) SMP initiator phy;
   J) SMP target phy;
   K) virtual SMP initiator phy; and
   L) virtual SMP target phy;

   and

b) from the expander phy class:
   A) expander phy; and
   B) virtual expander phy.

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

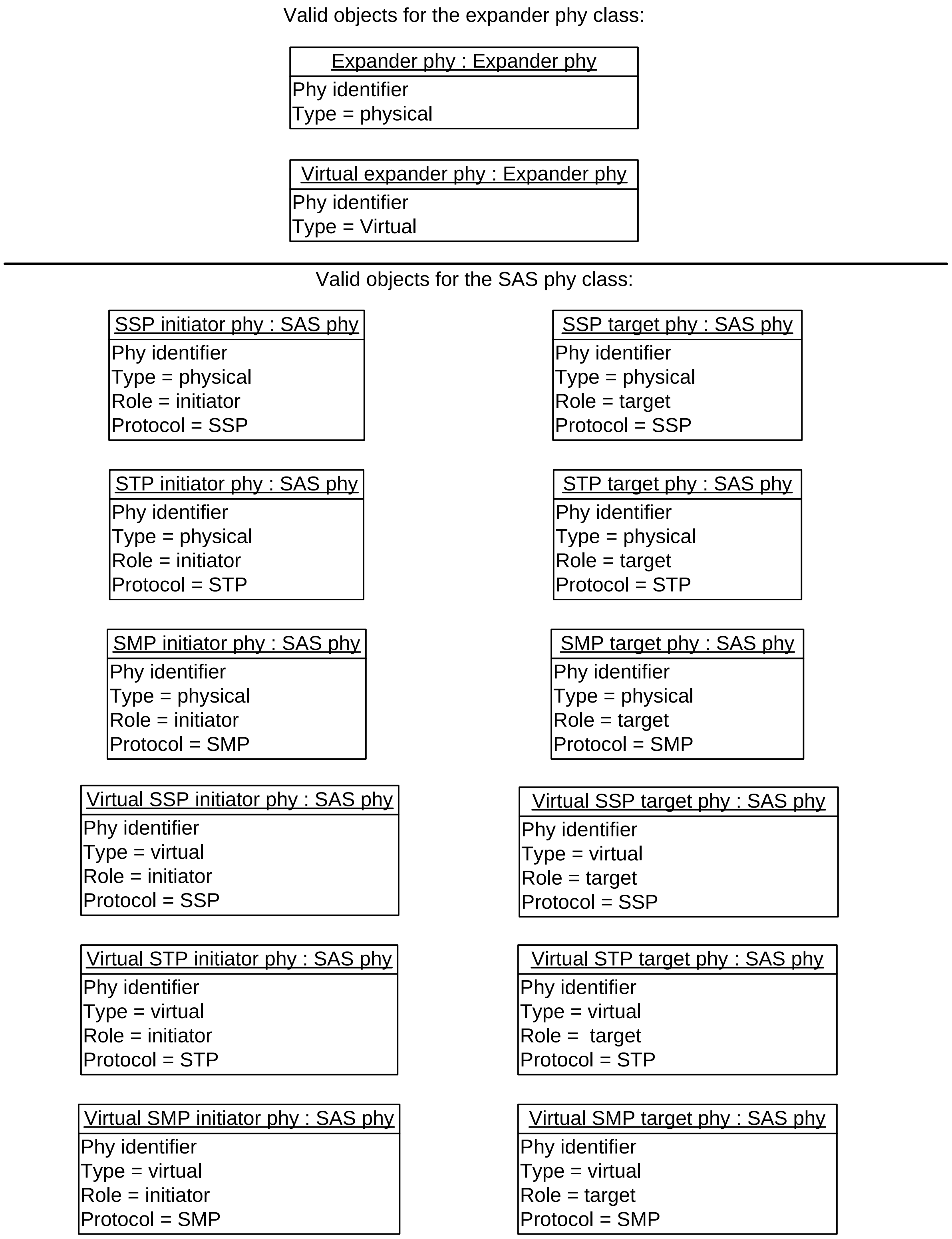


Figure 4 — Phy object diagram

4.1.9 Pathways

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

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

Figure 5 shows examples of potential pathways.

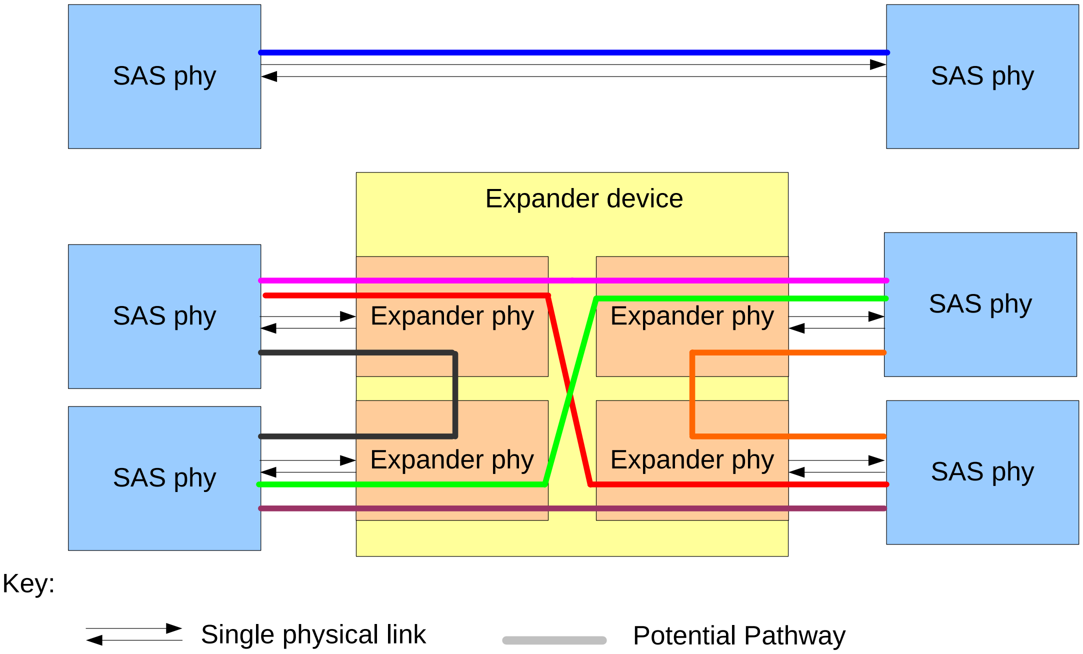


Figure 5 — Potential pathways [no changes]

A partial pathway is the set of ~~physical~~logical links participating in a connection request that have not yet conveyed a connection response (see 7.12).

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

4.1.10 Connections

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

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 received by 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~~logical link rate.

No more than one connection is active on a ~~physical~~logical link at a time. If the connection is an SSP or SMP connection and there are no dwords to transmit associated with that connection, idle dwords are transmitted. If the connection is an STP connection and there are no dwords to transmit associated with that connection, SATA_SYNCs, SATA_CONTs, or vendor-specific scrambled data dwords (after a SATA_CONT) are transmitted. If there is no connection on a ~~physical~~logical link then idle dwords are transmitted.

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

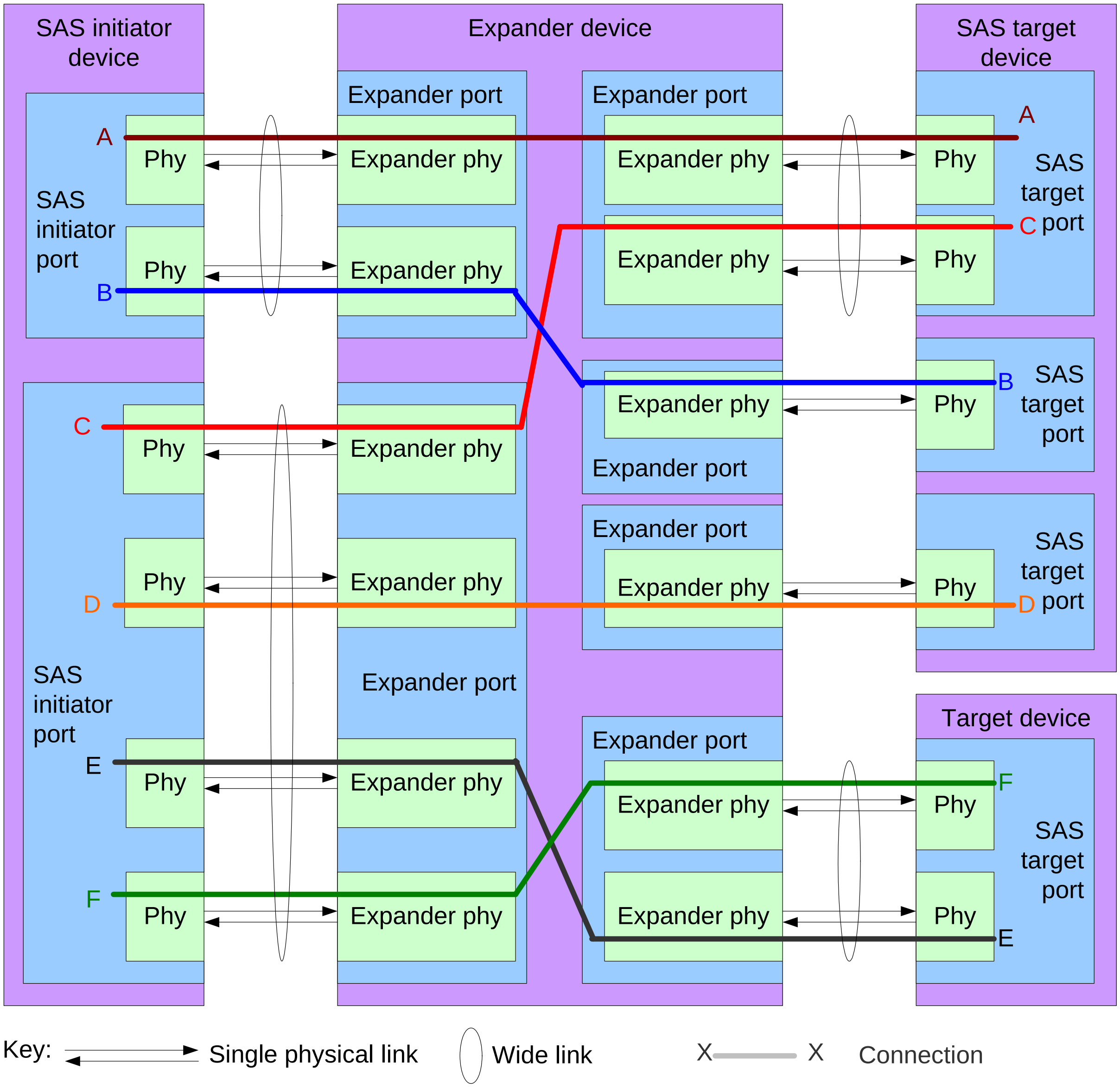
If multiple potential pathways exist between the SAS initiator port(s) and the SAS target port(s), multiple connections may be established by a SAS port between the following:

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

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

Figure 6 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.



Note: 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 6 — Multiple connections on wide ports [no changes]

4.1.x Logical links

A physical link with a physical link rate greater than 1,5 Gbps may be multiplexed into two logical links as defined in table 1.

Table 1 — Logical links

Physical link rate	Logical link(s)
6 Gbps	One 6 Gbps logical link
	Two 3 Gbps logical links
3 Gbps	One 3 Gbps logical link
	Two 1,5 Gbps logical links
1,5 Gbps	One 1,5 Gbps logical link

Multiplexing is defined in 7.xx.

4.3 State machines

4.3.1 State machine overview

Figure 7 shows the state machines for SAS devices, their relationships to each other and to the SAS device, SAS port, and SAS phy classes.

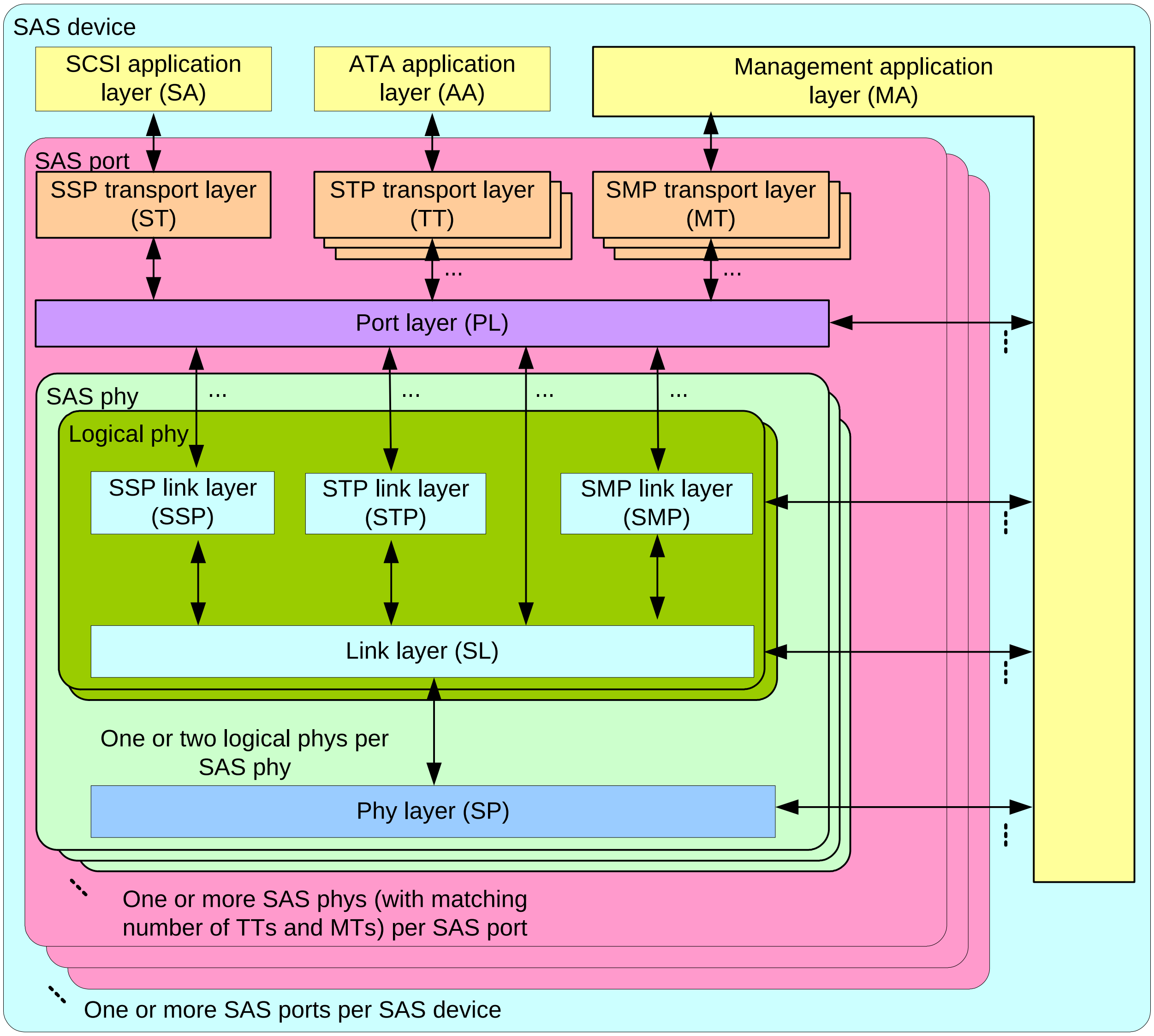


Figure 7 — State machines for SAS devices [updated to include logical phys]

Figure 8 shows the state machines for expander devices, their relationships to each other and to the expander device, expander port, and expander phy classes. Expander function state machines are not defined in this standard, but the interface to the expander function is defined in 4.6.6.

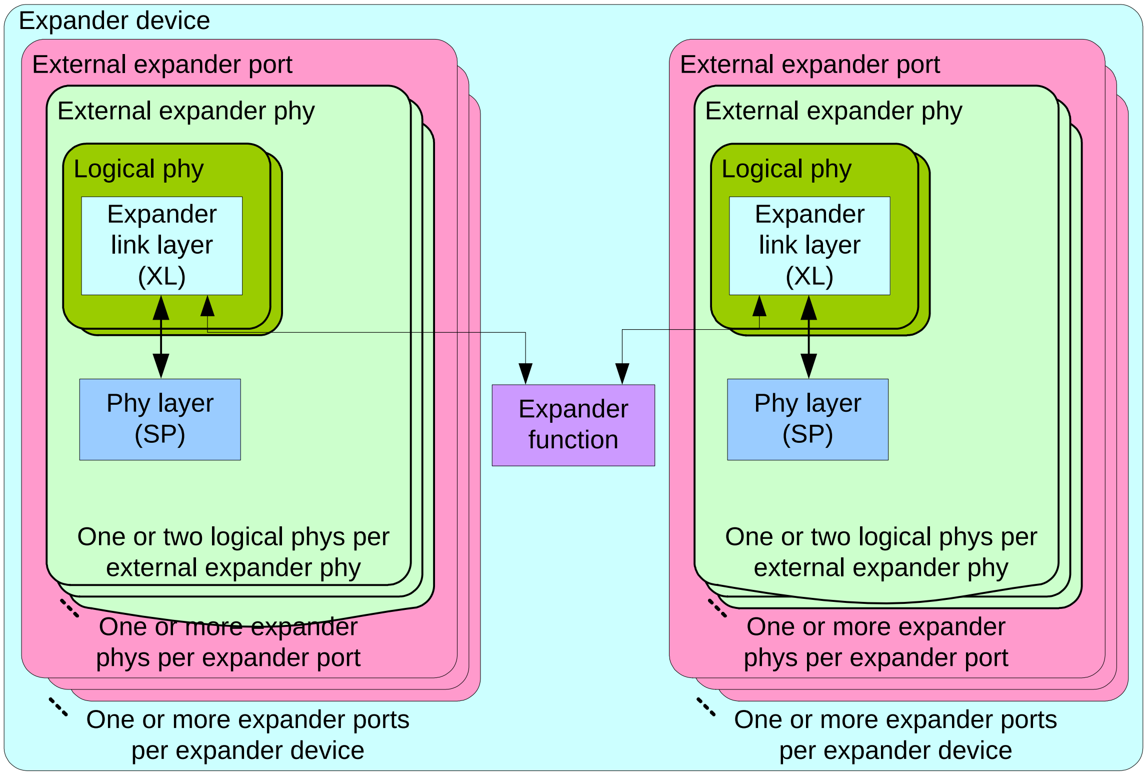

Expander device
External expander port
External expander phy
Logical phy
Expander link layer (XL)
Phy layer (SP)
One or two logical phys per external expander phy
One or more expander phys per expander port
One or more expander ports per expander device
Expander function
External expander port
External expander phy
Logical phy
Expander link layer (XL)
Phy layer (SP)
One or two logical phys per external expander phy
One or more expander phys per expander port
One or more expander ports per expander device

Figure 8 — State machines for expander devices [updated to include logical phys]

Editor's Note 2: Consider including SL_IR in the previous figure alongside XL. It goes in the logical phy. It should probably be renamed SXL_IR or L_IR so the name doesn't imply it is only used by SAS phys.

Annex K contains a list of messages between state machines.

4.3.2 Transmit data path

Figure 9 shows the transmit data path in a SAS phy.

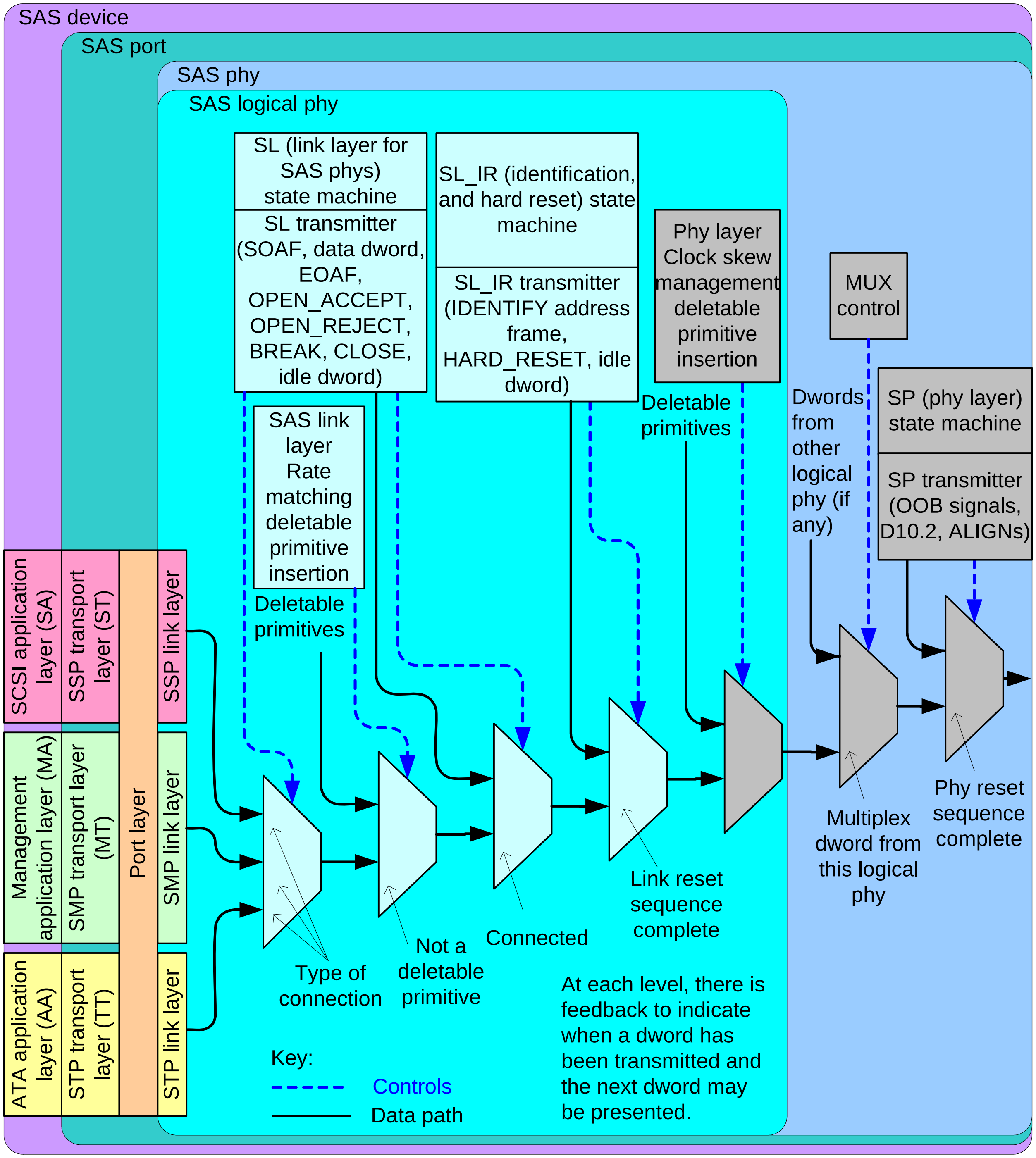

SAS device
SAS port
SAS phy
SAS logical phy
SL (link layer for SAS phys) state machine
SL transmitter (SOAF, data dword, EOAF, OPEN_ACCEPT, OPEN_REJECT, BREAK, CLOSE, idle dword)
SL_IR (identification, and hard reset) state machine
SL_IR transmitter (IDENTIFY address frame, HARD_RESET, idle dword)
Phy layer Clock skew management deletable primitive insertion
MUX control
SP (phy layer) state machine
SP transmitter (OOB signals, D10.2, ALIGNs)
Dwords from other logical phy (if any)
Deletable primitives
SAS link layer Rate matching deletable primitive insertion
Deletable primitives
SCSI application layer (SA)
SSP transport layer (ST)
Port layer
SSP link layer
Management application layer (MA)
SMP transport layer (MT)
SMP link layer
ATA application layer (AA)
STP transport layer (TT)
STP link layer
Type of connection
Not a deletable primitive
Connected
Link reset sequence complete
Multiplex dword from this logical phy
Phy reset sequence complete
At each level, there is feedback to indicate when a dword has been transmitted and the next dword may be presented.
Key:
Controls
Data path

Figure 9 — Transmit data path in a SAS phy [updated to include SAS logical phy]

...

Figure 10 shows the transmit data path in an expander phy.

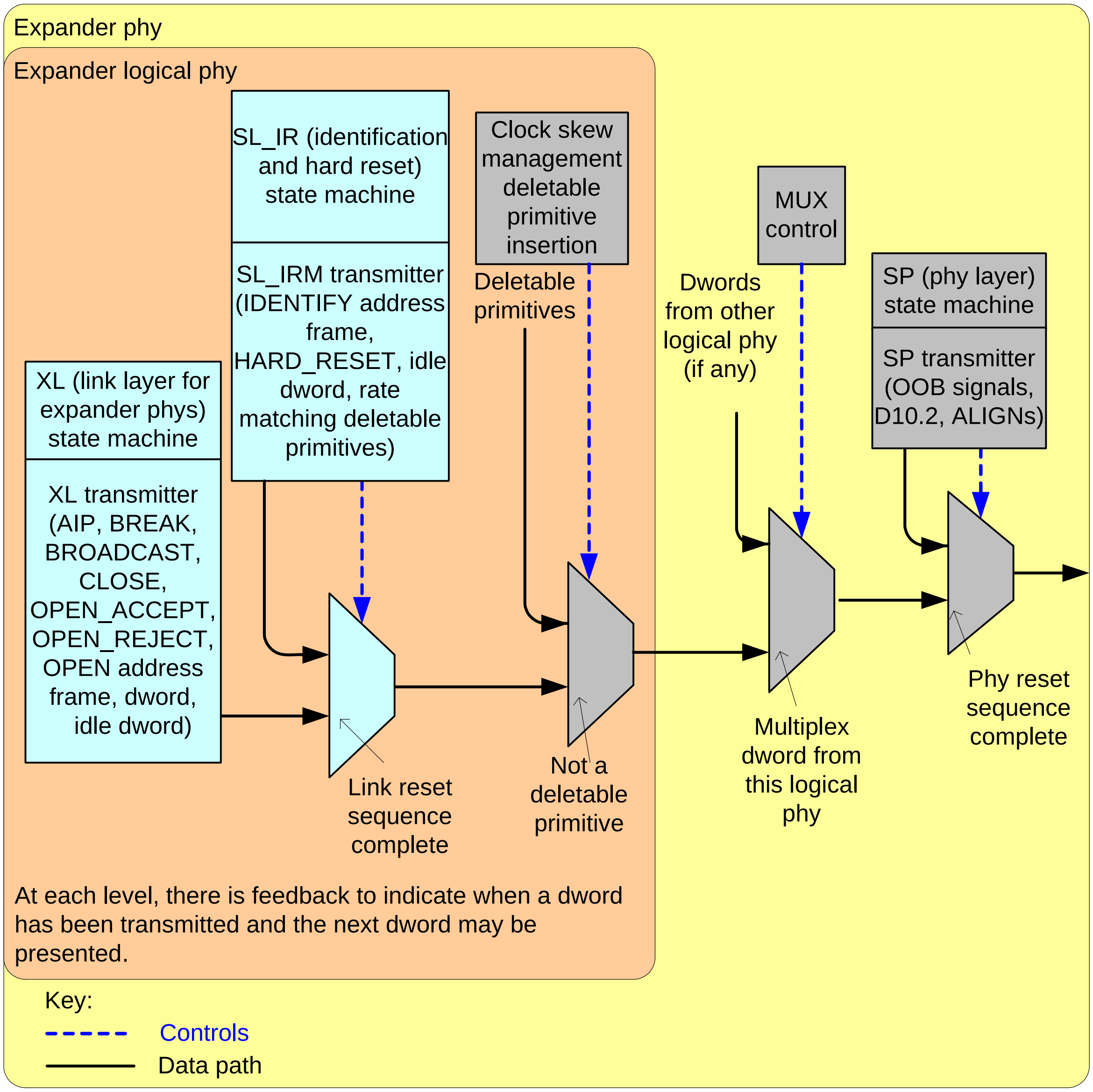

Expander phy
Expander logical phy
SL_IR (identification and hard reset) state machine
SL_IRM transmitter (IDENTIFY address frame, HARD_RESET, idle dword, rate matching deletable primitives)
Clock skew management deletable primitive insertion
Deletable primitives
MUX control
Dwords from other logical phy (if any)
SP (phy layer) state machine
SP transmitter (OOB signals, D10.2, ALIGNs)
XL (link layer for expander phys) state machine
XL transmitter (AIP, BREAK, BROADCAST, CLOSE, OPEN_ACCEPT, OPEN_REJECT, OPEN address frame, dword, idle dword)
Link reset sequence complete
Not a deletable primitive
Multiplex dword from this logical phy
Phy reset sequence complete
At each level, there is feedback to indicate when a dword has been transmitted and the next dword may be presented.
Key:
Controls
Data path

Figure 10 — Transmit data path ~~and state machines~~ in an expander phy [updated to include expander logical phy]

4.3.3 Receive data path

The SP_DWS receiver (see 6.9.2) establishes dword synchronization and sends dwords to the SP_DWS state machine (see 6.9) and to the link layer state machine receivers.

If multiplexing is enabled (see table 14 in x.y), the SP_DWS receiver uses incoming MUX primitives to determine the logical link numbers and route the dwords to the appropriate link layer receivers.

Figure 11 shows the receive data path in a SAS phy.

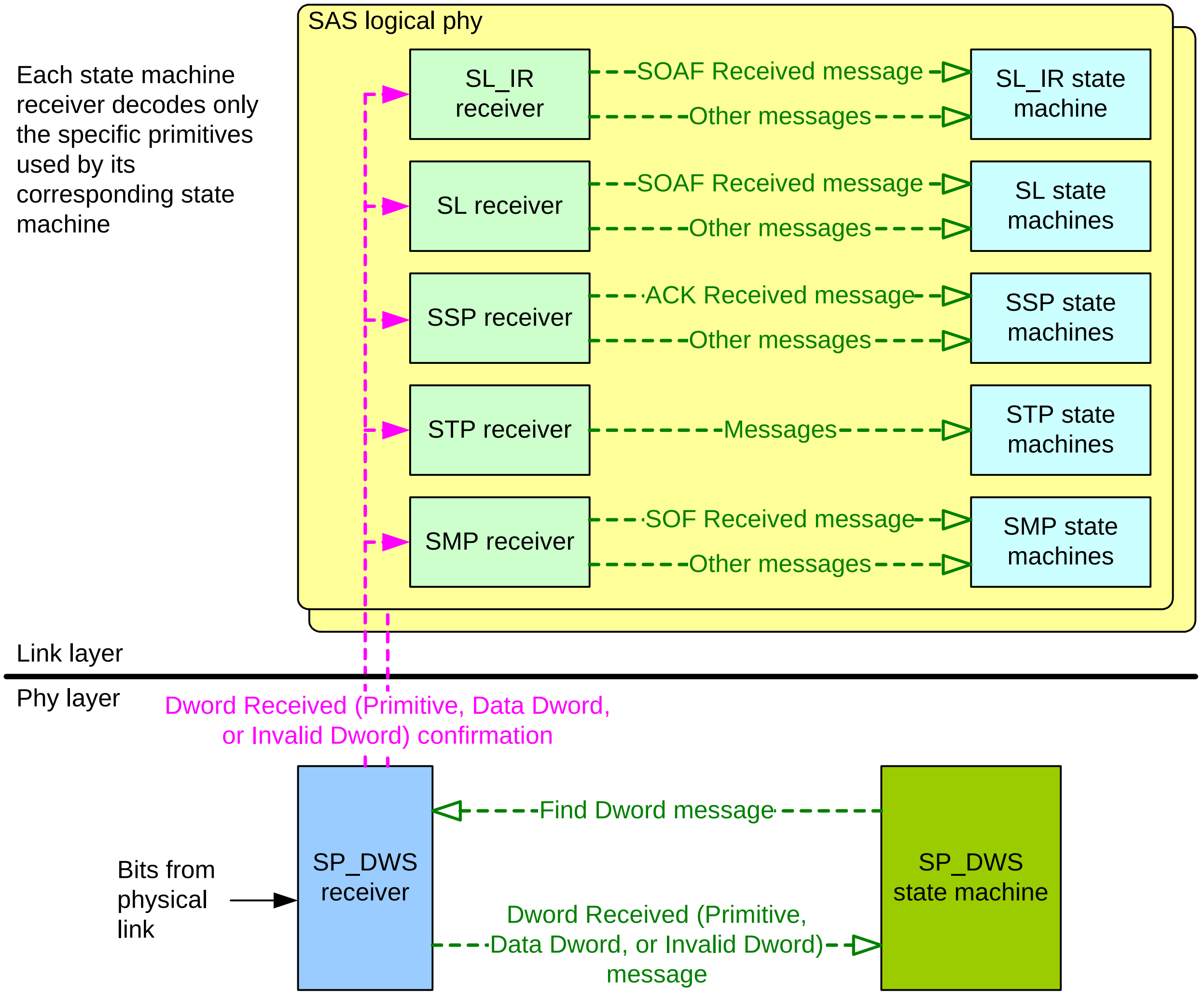


Figure 11 — Receive data path in a SAS phy [updated]

Figure 12 shows the receive data path in an expander phy.

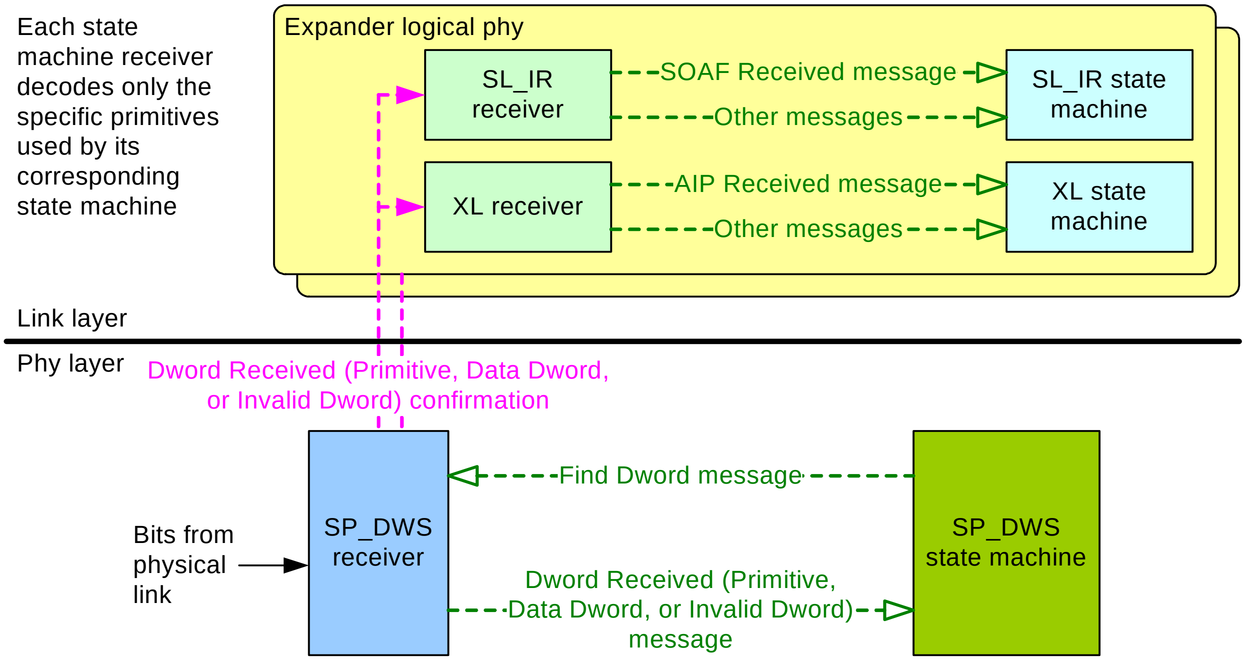


Figure 12 — Receive data path in an expander phy [updated]

4.4 Resets

4.4.1 Reset overview

Figure 13 illustrates the reset terminology used in this standard:

a) link reset sequence;
b) phy reset sequence (see 6.7);
c) SATA OOB sequence (see 6.7.2.1);
d) SATA speed negotiation sequence (see 6.7.2.2);
e) SAS OOB sequence (see 6.7.4.1);
f) SAS speed negotiation sequence (see 6.7.4.2);
g) hard reset sequence (see 7.9); and
h) identification sequence (see 7.9).

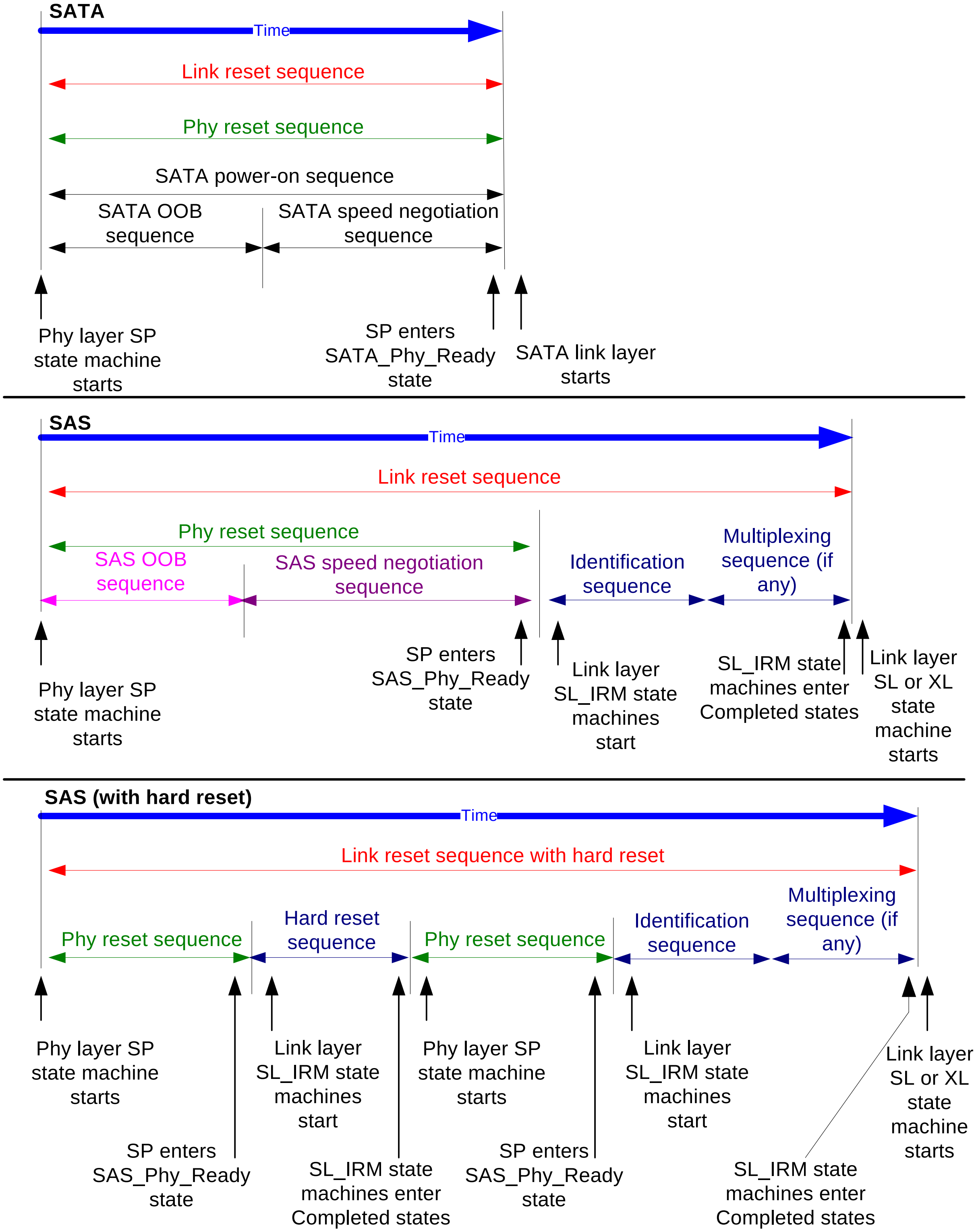

SATA
Time
Link reset sequence
Phy reset sequence
SATA power-on sequence
SATA OOB sequence
SATA speed negotiation sequence
Phy layer SP state machine starts
SP enters SATA_Phy_Ready state
SATA link layer starts
SAS
Time
Link reset sequence
Phy reset sequence
SAS OOB sequence
SAS speed negotiation sequence
Identification sequence
Multiplexing sequence (if any)
Phy layer SP state machine starts
SP enters SAS_Phy_Ready state
Link layer SL_IRM state machines start
SL_IRM state machines enter Completed states
Link layer SL or XL state machine starts
SAS (with hard reset)
Time
Link reset sequence with hard reset
Phy reset sequence
Hard reset sequence
Phy reset sequence
Identification sequence
Multiplexing sequence (if any)
Phy layer SP state machine starts
SP enters SAS_Phy_Ready state
Link layer SL_IRM state machines start
SL_IRM state machines enter Completed states
Phy layer SP state machine starts
SP enters SAS_Phy_Ready state
Link layer SL_IRM state machines start
SL_IRM state machines enter Completed states
Link layer SL or XL state machine starts

Figure 13 — Reset terminology [no change]

The phy reset sequences, including the OOB sequence and speed negotiation sequences, are implemented by the SP state machine and are described in 6.7 and 6.8. The hard reset sequence and identification sequence are implemented by the SL_IR state machine and are described in 7.9.

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

4.6 Expander device model

4.6.1 Expander device model overview

...

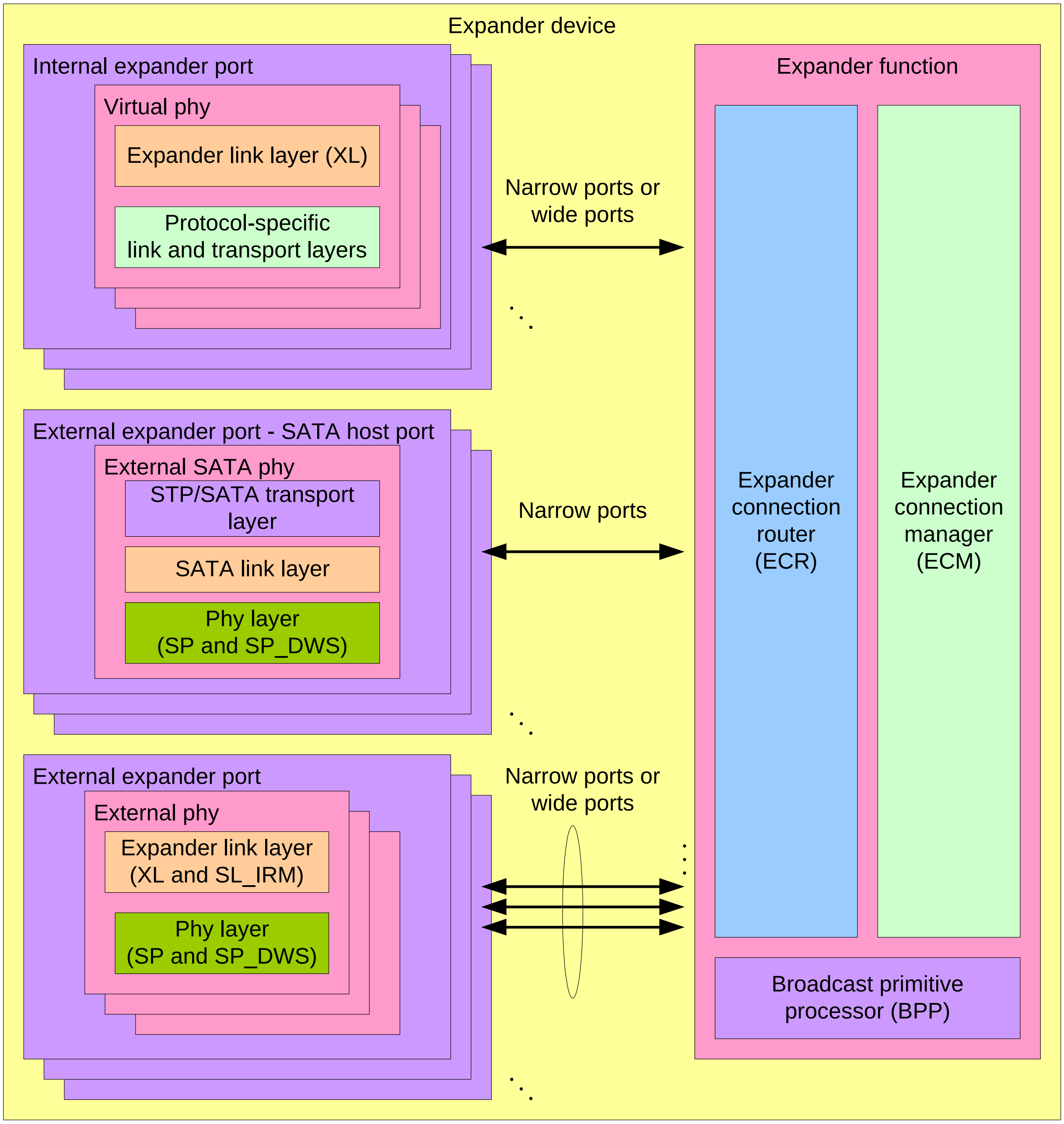

Expander device
Internal expander port
Virtual phy
Expander link layer (XL)
Protocol-specific link and transport layers
Narrow ports or wide ports
External expander port - SATA host port
External SATA phy
STP/SATA transport layer
SATA link layer
Phy layer (SP and SP_DWS)
Narrow ports
External expander port
External phy
Expander link layer (XL and SL_IRM)
Phy layer (SP and SP_DWS)
Narrow ports or wide ports
Expander function
Expander connection router (ECR)
Expander connection manager (ECM)
Broadcast primitive processor (BPP)

Figure 14 — Expander device model [no change]

4.6.2 Expander ports

An external expander port contains one or more physical phys (see 4.1.2). Since each phy in the expander device has the same SAS address, expander ports are created based on the attached SAS addresses (see 4.1.3).

Each phy in an expander port shall have the same routing attribute (see 4.6.7.1), and the DISCOVER function (see 10.4.3.5) shall return the same value in the ROUTING ATTRIBUTE field for each phy in an expander port.

Each phy in an expander port containing phys with table routing attributes shall have the same number of routing table entries (see 4.6.7.3)

A set of expander phys with table routing attributes using the same external connector (see 5.2.3.3) is called an enclosure out port. A set of expander phys with subtractive routing attributes using the same external connector is called an enclosure in port.

Each phy in an expander port shall have the same zone phy information (see 4.8.3.1). The zone phy information associated with each of the phys in an expander port is treated as the zoning properties of the expander port.

Each expander phy contains an expander link layer with an XL state machine (see 7.15) and an SL_IR state machine (see 7.9.5). The XL state machine in each expander phy within an expander port processes connection requests independently of the XL state machines in other expander phys.

An internal expander port contains a virtual phy with an expander link layer and a protocol-specific transport layer (e.g., to provide access as an SSP target port to a logical unit with a peripheral device type set to 0Dh (i.e., enclosure services device) (see SPC-3 and SES-2)).

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

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

Each STP/SATA bridge shall have a unique SAS address. This SAS address is reported in the ATTACHED SAS ADDRESS field in the DISCOVER function (see 10.4.3.5) for the expander phy containing the STP/SATA bridge (i.e., the expander phy attached to the SATA device or SATA port selector).

4.6.4 Expander connection router (ECR)

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

When forwarding dwords during a connection from a source phy with a higher physical link rate to a destination phy with a lower physical link rate, rate matching (see 7.13) ensures the dwords are at a connection rate equal to or less than the lower physical link rate. However, the ECR may be requested to forward more dwords than the destination phy is able to accept if:

a) an invalid dword occurs during a~~n ALIGN or NOTIFY~~deletable primitive;
b) an invalid dword occurs during a CLOSE; or
c) multiple invalid dwords occur during a BREAK.

The ECR may discard dwords if needed and count them as receive elasticity buffer overflows (see 4.9).

4.6.5 Broadcast primitive processor (BPP)

The BPP receives BROADCASTs from each expander logical phy and requests transmission of those BROADCASTs on all expander ports except the expander port from which the BROADCAST was received.

In a self-configuring expander device (see 4.1.5), the BPP requests transmission of a BROADCAST (CHANGE) when it completes configuration (see 10.4.3.3).

In a zoning expander device (see 4.8.2), BROADCASTs and SMP ZONED BROADCAST requests are forwarded as BROADCASTs and/or SMP ZONED BROADCAST request (see 10.4.3.11).

4.6.6 Expander device interfaces

4.6.6.1 Expander device interface overview

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

Figure 43 describes the interfaces present within an expander device.

Editor’s Note 3: Change “Expander phy” to “Expander logical phy” in the “Expander device interfaces” figure

4.6.6.2 Expander device interface detail

Figure 44 shows the interface requests, confirmations, indications, and responses used by an expander device to manage connections.

Editor’s Note 4: Change “Source expander phy” to “Source expander logical phy” and “Destination expander phy” to “Destination expander logical phy” in the “Expander device interface detail” figure

4.6.6.3 ECM interface

Table 2 describes the requests from an expander logical phy to the ECM.

Table 2 — Expander logical phy to ECM requests

Message	Description
Request Path (arguments)	Request for a connection.
Partial Pathway Timeout Timer Expired	The Partial Pathway Timeout Timer expired.

Table 3 describes the responses from an expander logical phy to the ECM.

Table 3 — Expander logical phy to ECM responses

Message	Description
Phy Status (Partial Pathway)	Response meaning that an expander logical phy: a) is being used for an unblocked partial pathway (i.e., the expander logical phy is in the XL3:Open_Confirm_Wait state and the last AIP transmitted was not AIP (WAITING ON PARTIAL), or the expander logical phy is in the XL6:Open_Response_Wait state and the last AIP received was not AIP (WAITING ON PARTIAL)); or b) has sent a Request Path request to the ECM and is receiving Arbitrating (Waiting On Partial) from the ECM.
Phy Status (Blocked Partial Pathway)	Response meaning that an expander logical phy: a) is being used for a blocked partial pathway (i.e., the expander logical phy is in the XL3:Open_Confirm_Wait state and the last AIP transmitted was AIP (WAITING ON PARTIAL), or the expander logical phy is in the XL6:Open_Response_Wait state and the last AIP received was AIP (WAITING ON PARTIAL)); or b) has sent a Request Path request to the ECM and is receiving Arbitrating (Blocked On Partial) from the ECM.
Phy Status (Connection)	Response meaning that an expander logical phy: a) is being used for a connection (i.e., the expander logical phy is in the XL7:Connected or XL8:Close_Wait state); or b) has sent a Request Path request to the ECM and is receiving Arbitrating (Waiting On Connection) from the ECM.

Table 4 describes the confirmations from the ECM to an expander logical phy. These confirmations are sent in confirmation of a Request Path request.

Table 4 — ECM to expander logical phy confirmations (part 1 of 2)

Message	**Description**
Arbitrating (Normal)	Confirmation that the ECM has received the Request Path request.
Arbitrating (Waiting On Partial)	Confirmation that the ECM has determined that: a) there is a destination port capable of routing to the requested destination SAS address; b) at least one phy within the destination port supports the requested connection rate; c) each of the phys within the destination port is returning a Phy Status (Partial Pathway) or Phy Status (Blocked Partial Pathway) response; and d) at least one of the phys within the destination port is returning a Phy Status (Partial Pathway) response.
Arbitrating (Blocked On Partial)	Confirmation that the ECM has determined that: a) there is a destination port capable of routing to the requested destination SAS address; b) at least one phy within the destination port supports the requested connection rate; and c) each of the phys within the destination port is returning a Phy Status (Blocked Partial Pathway) response.
Arbitrating (Waiting On Connection)	Confirmation that the ECM has determined that the connection request is blocked due to one of the following reasons: a) the connection request is blocked by an active connection; or b) there are insufficient routing resources within the expander to complete the connection request. A connection request shall be considered blocked by an active connection when: a) there is a destination port capable of routing to the requested destination SAS address; b) at least one phy within the destination port supports the requested connection rate; c) each of the phys within the destination port is returning a Phy Status (Partial Pathway), Phy Status (Blocked Partial Pathway), or Phy Status (Connection) response; and d) at least one of the phys within the destination port is returning a Phy Status (Connection) response.

Table 4 — ECM to expander logical phy confirmations (part 2 of 2)

Message	Description
Arb Won	Confirmation that an expander logical phy has won path arbitration.
Arb Lost	Confirmation that an expander logical phy has lost path arbitration.
Arb Reject (No Destination)	Confirmation that: a) there is no operational expander logical phy capable of routing to the requested destination SAS address; or b) the requested destination SAS address maps back to the requesting port (see 7.12.4.3 and 7.12.4.4).
Arb Reject (Bad Destination)	Confirmation that: a) the requested destination SAS address maps back to the requesting port; b) the requesting port is using the direct routing method or the table routing method; and c) the ECM has not chosen to return Arb Reject (No Destination) (see 7.12.4.3 and 7.12.4.4).
Arb Reject (Bad Connection Rate)	Confirmation that the ECM has determined that there is a destination port capable of routing to the requested destination SAS address but no phys within the destination port are configured to support the requested connection rate.
Arb Reject (Pathway Blocked)	Confirmation that the ECM has determined that the requesting expander logical phy shall back off according to SAS pathway recovery rules.

4.6.6.4 ECR interface

Table 5 describes the requests from an expander logical phy to the ECR and the corresponding indications from the ECR to another expander logical phy.

Table 5 — Expander logical phy to ECR to expander logical phy requests and indications

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

Table 6 describes the responses from an expander logical phy to the ECR and the corresponding confirmations from the ECR to another expander logical phy. These responses are sent in response to a Forward Open indication.

Table 6 — Expander logical phy to ECR to expander logical 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: a) AIP (WAITING ON DEVICE) has been received; or b) the expander logical phy has completed the forwarding of an OPEN address frame and has entered the XL6:Open_Response_Wait state.
Open Accept	Confirmation/response that OPEN_ACCEPT has been received.
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.

4.6.6.5 BPP interface

Table 7 describes the requests from an expander logical phy to the BPP. See 7.11 for more information on broadcasts. See 4.8.5 for more information on how zoning expander devices handle broadcasts.

Table 7 — Expander logical phy to BPP requests

Message	Description
Broadcast Event Notify (Phy Not Ready)	Request to transmit a BROADCAST (CHANGE) on all other ports because ~~an~~the expander phy's SP state machine transitioned from the SP15:SAS_PHY_Ready or SP22:SATA_PHY_Ready state to the SP0:OOB_COMINIT state (see 6.8).
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 6.8 and 6.10) by the expander phy.
Broadcast Event Notify (Identification Sequence Complete)	Request to transmit a BROADCAST (CHANGE) on all other ports because ~~an~~the expander phy has completed the identification sequence (see 7.9).
Broadcast Event Notify (SATA Port Selector Change)	Request to transmit a BROADCAST (CHANGE) on all other ports because the expander phy detected that a SATA port selector appeared or disappeared.
Broadcast Event Notify (CHANGE Received)	Request to transmit a BROADCAST (CHANGE) on all other ports because a BROADCAST (CHANGE) was received by the expander logical phy.
Broadcast Event Notify (RESERVED CHANGE Received)	Request to transmit a BROADCAST (RESERVED CHANGE) on all other ports because a BROADCAST (RESERVED CHANGE) was received by the expander logical phy.
Broadcast Event Notify (SES Received)	Request to transmit a BROADCAST (SES) on all other ports because a BROADCAST (SES) was received by the expander logical phy.
Broadcast Event Notify (EXPANDER Received)	Request to transmit a BROADCAST (EXPANDER) on all other ports because a BROADCAST (EXPANDER) was received by the expander logical phy.

Editor's Note 5: 06-177 Make similar changes to the Reserved types which appear in later working drafts than sas2r02 on which this was based

Table 8 describes the indications from the BPP to an expander logical phy.

Table 8 — BPP to expander logical 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 logical phys in 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, the SMP port of an attached expander device, or SAS devices contained in the expander device.

Table 9 describes the routing methods that the ECM uses based on the routing attributes of an expander phy.

Table 9 — Routing attributes and routing methods

Routing attribute of an expander phy	Routing method used by ECM for the expander phy
Direct	Direct [a]
Table	Direct, if attached to an end device
	Direct, if attached to an expander device, for the SAS address of the expander device
	Table, if attached to an expander device, for SAS addresses beyond the expander device
Subtractive	Direct, if attached to an end device
	Subtractive, if attached to an expander device

[a] If attached to an expander device, the ECM is only able to route to the expander device itself through a phy with the direct routing attribute

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

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 4.7) 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 logical phy to a destination expander logical phy in a different expander port if the destination expander logical phy is enabled and operating at a valid ~~physical~~ logical link rate (e.g., the DISCOVER ~~function~~ response reports a NEGOTIATED ~~PHYSICAL~~LOGICAL LINK RATE field set to G1 (i.e., 8h) ~~or~~, G2 (i.e., 9h), or G3 (i.e., Ah)) using the following precedence:

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

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

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

4.7 Discover process

4.7.x Enabling multiplexing

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

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

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

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

If the SAS domain contains all 1,5 Gbps target phys, then the management application client should:

a) multiplex each 6 Gbps physical link into two 3 Gbps logical links; and
b) multiplex each 3 Gbps physical link into two 1,5 Gbps logical links.

NOTE 1 - Rate matching is used for 1,5 Gbps connections carried on 3 Gbps logical links.

4.9 Phy event information

...

Table 10 — PHY EVENT INFORMATION SOURCE field

Code	Name	Type [a]	Description
...			
05h	Elasticity buffer overflow count	WC	Number of times the phy’s receive elasticity buffer (see 7.3) has overflowed (e.g., because it did receive a sufficient number of ~~ALIGNs and/or NOTIFYs~~deletable primitives)
...			

[a] The Type column indicates the source type:
a) WC = wrapping counter
b) PVD = peak value detector
c) N/A = not applicable

Changes to chapter 6 (phy layer)

Add MUX transmission to the SP state machine and make SP_DWS cognizant of multiplexing. If multiplexing is enabled, the phy must give up immediately upon losing dword synchronization and not try to reacquire it, since it won’t know which logical links are which.

6.6.3 Receiving OOB signals

...

A SAS receiver device shall detect OOB bursts comprised of ALIGN (0) primitives 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 device supporting only 3~~,0~~ Gbps detects 1,5 Gbps based ALIGN (0) primitives, providing interoperability with a SAS transmitter device supporting both 1,5 Gbps and 3~~,0~~ Gbps).

6.7.4.2.1 SAS speed negotiation sequence overview

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 (see 6.7.4.2.2) for each physical link rate, starting with 1,5 Gbps, then 3~~,0~~ Gbps, then the next physical link rate. The length of the speed negotiation sequence (i.e., the number of speed negotiation windows) is determined by the number of physical link rates supported by the phys.

...

6.7.4.2.3 SAS speed negotiation sequence

The SAS speed negotiation sequence consists of a set of speed negotiation windows (see 6.7.4.2.2) for each physical link rate in this order:

1) G1 (i.e., 1,5 Gbps);
2) G2 (i.e., 3~~,0~~ Gbps);
3) G3 (i.e., 6 Gbps), if needed;
4) G4, if needed; and
5) etc.

Editor's Note 6: The above is being reworded by the speed negotiation window proposals. See 06-324.

[text from 06-363r2 is in black. Location of this text is TBD.]

4.9.1 Speed Negotiation Window - 3 information supported changes

The phy shall negotiate to the highest commonly supported settings based on the outgoing and incoming SNW-3 information supported settings bits. For bits defined as reserved, the phy shall transmit zeros in the outgoing SNW-3 information and shall ignore the bits in the incoming SNW-3 information. Table 11 defines the priority of the SNW-3 supported settings.

Table 11 — Negotiation priority of SNW-3 information supported settings

Priority	Supported setting
Highest	G3 with SSC
...	G3 without SSC
...	G2 with SSC
...	G2 without SSC
...	G1 with SSC
Lowest	G1 without SSC

If the phy does not successfully negotiate the highest commonly supported setting in the final SNW, it shall perform another final SNW with the next highest commonly supported setting (e.g., if the phys support both G3 without SSC and G3 with SSC but fail to negotiate the final SNW successfully using G3 with SSC, they shall perform another final SNW using G3 without SSC).

Table 12 defines the SNW-3 information.

Table 12 — SNW-3 information

Bit(s)	Description
Header	
0 (first bit)	START bit
1	TX SSC TYPE bit
~~2 to 7~~ 2 to 3	Reserved
4 to 7	REQUESTED LOGICAL LINK RATE field (bit 4 is the MSB, bit 7 is the LSB)
Supported settings	
8	G1 WITH SSC SUPPORTED bit
9	G1 WITHOUT SSC SUPPORTED bit
10	G2 WITH SSC SUPPORTED bit
11	G2 WITHOUT SSC SUPPORTED bit
12	G3 WITH SSC SUPPORTED bit
13	G3 WITHOUT SSC SUPPORTED bit
14 to 30	Reserved
Trailer	
31 (last bit)	PARITY field

The START bit shall be set to one. The phy's receiver shall use this bit to establish the timing for the subsequent bits.

A TX SSC TYPE bit set to one indicates that the phy's transmitter uses center-spreading SSC when SSC is enabled. A TX SSC TYPE bit set to zero indicates that the phy's transmitter uses down-spreading SSC when SSC is enabled, or that the phy does not support SSC.

NOTE 2 - The phy receiver may use the TX SSC TYPE bit to optimize its CDR circuitry. This bit indicates the type of SSC used when attached to a SAS phy or an expander phy; if a phy supports center-spreading when attached to a SAS phy or an expander phy and down-spreading when attached to a SATA phy, it sets the TX SSC TYPE bit to one.

The REQUESTED LOGICAL LINK RATE field indicates the logical link rate that the phy is requesting. If the phy is managed by an SMP target port, the field is based on the REQUESTED LOGICAL LINK RATE field in the SMP PHY CONTROL and DISCOVER functions. This field is independent of the supported settings bits (e.g., G3, G2, and G1 with or without SSC).

Table 13 defines the requested logical link rate based on the transmitted and received REQUESTED LOGICAL LINK RATE fields.

Table 13 — Requested logical link rate

Transmitted REQUESTED LOGICAL LINK RATE field	**Received REQUESTED LOGICAL LINK RATE field**	**Requested logical link rate**
0h (i.e., no multiplexing)	Any	Negotiated physical link rate
8h (i.e., 1,5 Gbps)	8h (i.e., 1,5 Gbps)	1,5 Gbps
	9h (i.e., 3 Gbps)	
	Ah (i.e., 6 Gbps)	
9h (i.e., 3 Gbps)	8h (i.e., 1,5 Gbps)	1,5 Gbps
	9h (i.e., 3 Gbps)	3 Gbps
	Ah (i.e., 6 Gbps)	
Ah (i.e., 6 Gbps)	8h (i.e., 1,5 Gbps)	1,5 Gbps
	9h (i.e., 3 Gbps)	3 Gbps
	Ah (i.e., 6 Gbps)	6 Gbps
All others	All others	1,5 Gbps

Table 14 defines whether or not multiplexing is enabled and the negotiated logical link rate based on the requested logical link rate (see table 13) and the negotiated physical link rate.

Table 14 — Multiplexing negotiation

Requested logical link rate (see table 13)	**Negotiated physical link rate**	**Multiplexing**	**Negotiated logical link rate**
1,5 Gbps	1,5 Gbps	Disabled	1,5 Gbps
	3 Gbps	Enabled	1,5 Gbps
	6 Gbps		3 Gbps
3 Gbps	1,5 Gbps	Disabled	1,5 Gbps
	3 Gbps		3 Gbps
	6 Gbps	Enabled	3 Gbps
6 Gbps	1,5 Gbps	Disabled	1,5 Gbps
	3 Gbps		3 Gbps
	6 Gbps		6 Gbps
Negotiated physical link rate	1,5 Gbps	Disabled	1,5 Gbps
	3 Gbps		3 Gbps
	6 Gbps		6 Gbps

The G<GENERATION NUMBER> WITH SSC SUPPORTED bits and G<GENERATION NUMBER> WITHOUT SSC SUPPORTED bits indicate the physical link rates and SSC options the phy is attempting to negotiate.

A G1 WITH SSC SUPPORTED bit set to one indicates that the phy supports G1 (i.e., 1,5 Gbps) with SSC. A G1 WITH SSC SUPPORTED bit set to zero indicates that the phy does not support G1 with SSC.

A G1 WITHOUT SSC SUPPORTED bit set to one indicates that the phy supports G1 without SSC. A G1 WITHOUT SSC SUPPORTED bit set to zero indicates that the phy does not support G1 without SSC.

A G2 WITH SSC SUPPORTED bit set to one indicates that the phy supports G2 (i.e., 3 Gbps) with SSC. A G2 WITH SSC SUPPORTED bit set to zero indicates that the phy does not support G2 with SSC.

A G2 WITHOUT SSC SUPPORTED bit set to one indicates that the phy supports G2 without SSC. A G2 WITHOUT SSC SUPPORTED bit set to zero indicates that the phy does not support G2 without SSC.

A G3 WITH SSC SUPPORTED bit set to one indicates that the phy supports G3 (i.e., 6 Gbps) with SSC. A G3 WITH SSC SUPPORTED bit set to zero indicates that the phy does not support G3 with SSC.

A G3 WITHOUT SSC SUPPORTED bit set to one indicates that the phy supports G3 without SSC. A G3 WITHOUT SSC SUPPORTED bit set to zero indicates that the phy does not support G3 without SSC.

The PARITY bit provides for error detection of all the SNW-3 information bits. The PARITY bit shall be set to one or zero such that the total number of SNW-3 information bits that are set to one is even, including the START bit and the PARITY bit. If the PARITY bit received is incorrect, the phy shall consider it a phy reset problem.

Table 15 lists some example SNW-3 information values.

Table 15 — Example SNW-3 information values

Code [a]	Description
80A80000h	Down-spreading SSC G1, G2, and G3 with SSC supported
80FC0001h	Down-spreading SSC G1, G2, and G3 with and without SSC supported
C0540001h	Center-spreading SSC G1, G2, and G3 without SSC supported
C0FC0000h	Center-spreading SSC G1, G2, and G3 with and without SSC supported

[a] Expressed as a 32-bit value with bit 0 (i.e., the START bit) as the MSB and bit 31 as the LSB (i.e., the PARITY bit).

6.8 SP (phy layer) state machine

6.8.1 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 6.8.3.2)(initial state);
b) SP1:OOB_AwaitCOMX (see 6.8.3.3);
c) SP2:OOB_NoCOMSASTimeout;
d) SP3:OOB_AwaitCOMINIT_Sent (see 6.8.3.5);
e) SP4:OOB_COMSAS (see 6.8.3.6);
f) SP5:OOB_AwaitCOMSAS_Sent (see 6.8.3.7);
g) SP6:OOB_AwaitNoCOMSAS (see 6.8.3.8);
h) SP7:OOB_AwaitCOMSAS (see 6.8.3.9);
i) SP8:SAS_Start (see 6.8.4.2);
j) SP9:SAS_RateNotSupported (see 6.8.4.3);
k) SP10:SAS_AwaitALIGN (see 6.8.4.4);
l) SP11:SAS_AwaitALIGN1 (see 6.8.4.5);

m) SP12:SAS_AwaitSNW (see 6.8.4.6);
n) SP13:SAS_Pass (see 6.8.4.7);
o) SP14 SAS_Fail (see 6.8.4.8);
p) SP15:SAS_PHY_Ready (see 6.8.4.9);
q) SP16:SATA_COMWAKE (see 6.8.5.2);
r) SP17:SATA_AwaitCOMWAKE (see 6.8.5.3);
s) SP18:SATA_AwaitNoCOMWAKE (see 6.8.5.4);
t) SP19:SATA_AwaitALIGN (see 6.8.5.5);
u) SP20:SATA_AdjustSpeed (see 6.8.5.6);
v) SP21:SATA_Transmit_ALIGN (see 6.8.5.7);
w) SP22:SATA_PHY_Ready (see 6.8.5.8);
x) SP23:SATA_PM_Partial (see 6.8.5.9);
y) SP24:SATA_PM_Slumber (see 6.8.5.10);
z) SP25:SATA_PortSel (see 6.8.6.2); and
aa) SP26:SATA_SpinupHold (see 6.8.7.2).

The SP state machine shall start in the SP0:OOB_COMINIT state after:

a) a power on;
b) a hard reset;
c) receiving a Management Reset request from the management layer (e.g., from the SMP PHY CONTROL function requesting a phy operation of LINK RESET or HARD RESET in an expander device); or
d) receiving a Disable Phy request from the management layer (e.g., from the SMP PHY CONTROL function requesting a phy operation of DISABLE in an expander device).

The SP state machine shall maintain a MgmtReset state machine variable to determine whether a Management Reset request has been received. Any SP state that receives a Management Reset request shall set the MgmtReset state machine variable to one before making a transition to the SP0:OOB_COMINIT state (see 6.8.3.2). Any SP state that receives a power on, or a hard reset shall set the MgmtReset state machine variable to zero before making a transition to the SP0:OOB_COMINIT state.

If the phy supports SATA port selectors, the SP state machine shall maintain a COMWAKE_Received state machine variable to indicate whether a COMWAKE detected message was received in the SP0:OOB_COMINIT state or the SP1:OOB_AwaitCOMX state since the last time the SP0:OOB_COMINIT state was entered, and the SP state machine shall transition to the SP25:SATA_PortSel state whenever it receives a Transmit SATA Port Selection Signal request.

The SP state machine sends the following messages to the SP_DWS state machine (see 6.9):

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 16.

Table 16 — SP state machine timers

Timer	Initial value
COMSAS Detect Timeout timer	COMSAS detect timeout (see table 72 in 6.6.1)
Await ALIGN Timeout timer	Await ALIGN timeout (see table 79 in 6.7.2.2)
Hot-Plug Timeout timer	Hot plug timeout (see table 78 in 6.7.1)
RCDT timer	RCDT (see table 80 in 6.7.4.2)
SNLT timer	SNLT (see table 80 in 6.7.4.2)
SNTT timer	SNTT (see table 80 in 6.7.4.2)

6.8.2 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 6.8).

The SP transmitter receives the following messages from the SP state machine:

a) Transmit COMINIT;
b) Transmit COMSAS;
c) Transmit COMWAKE;
d) Transmit SATA Port Selection Signal;
e) Transmit D10.2;
f) Set Rate (Physical Link Rate); ~~and~~
g) Transmit ALIGN with an argument indicating the specific type (e.g., Transmit ALIGN (0)); and
h) Transmit MUX Sequence.

Upon receiving a Transmit MUX Sequence message, the SP transmitter transmits:

1) MUX (LOGICAL LINK 0);
2) MUX (LOGICAL LINK 1);
3) MUX (LOGICAL LINK 0);
4) MUX (LOGICAL LINK 1);
5) MUX (LOGICAL LINK 0); and
6) MUX (LOGICAL LINK 1).

When not otherwise instructed, the SP transmitter transmits D.C. idle.

The SP transmitter shall complete any physical link rate change requested with the Set Rate message within RCDT (see table 80 in 6.7.4.2).

.The SP transmitter sends the following messages to the SP state machine:

a) COMINIT Transmitted;
b) COMSAS Transmitted;
c) COMWAKE Transmitted; and
d) SATA Port Selection Signal 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) primitives 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.

6.8.3 OOB sequence states

6.8.3.1 OOB sequence states overview

Figure 15 shows the OOB sequence states. These states are indicated by state names with a prefix of OOB.

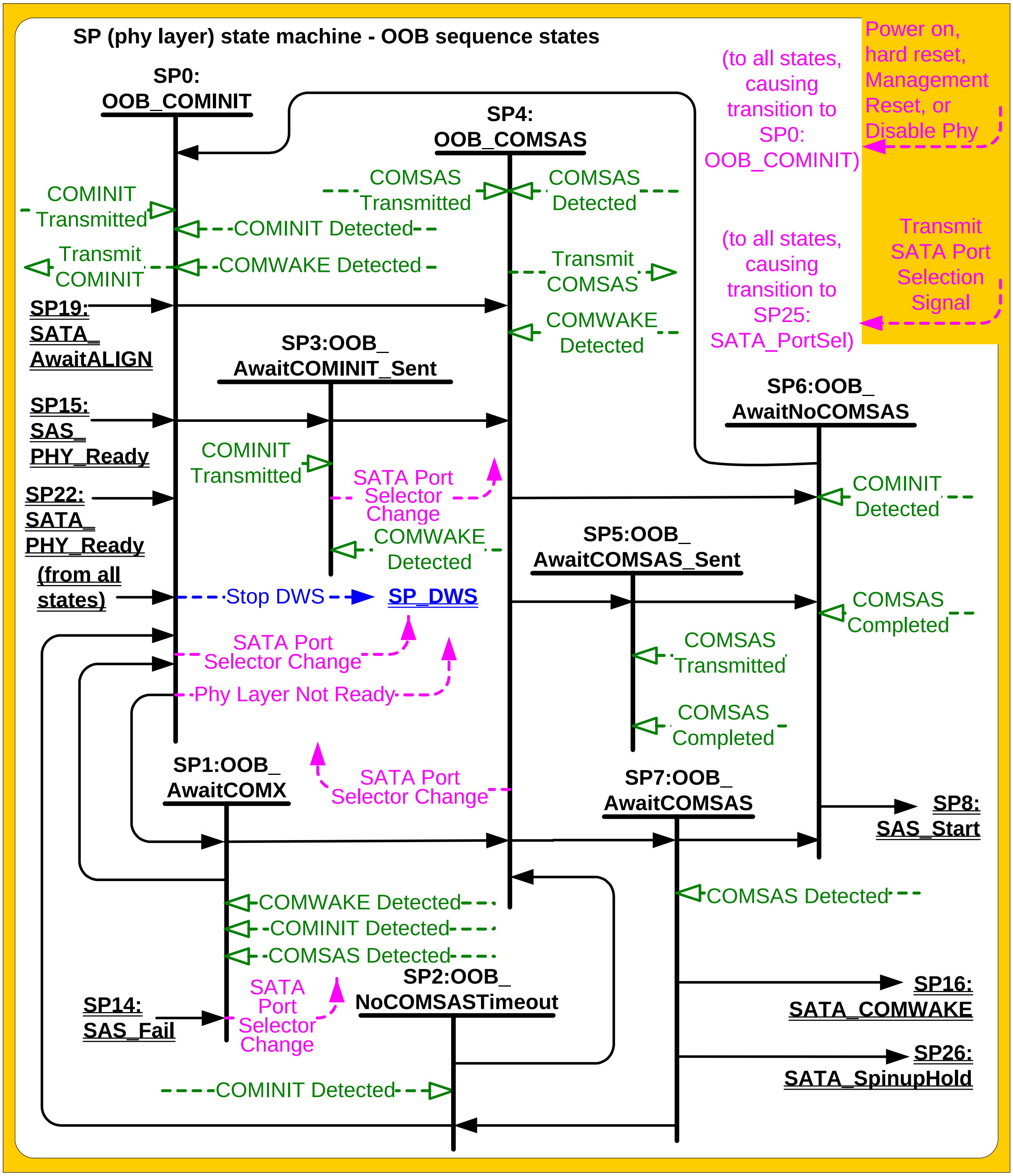


Figure 15 — SP (phy layer) state machine - OOB sequence states

6.8.3.2 SP0:OOB_COMINIT state

6.8.3.2.1 State description

This state is the initial state for this state machine.

This state machine waits for receipt of a COMINIT Transmitted message and/or a COMINIT Detected message.

Upon entry into this state, this state shall:

a) set the COMWAKE_Received state machine variable to zero;
b) send a Stop DWS message to the SP_DWS state machine;
c) send a Phy Layer Not Ready confirmation to the link layer; and
d) set the ATTACHED SATA DEVICE bit to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3).

If this state was entered due to power on, the phy shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3).

If this state was not entered because of a Disable Phy request, this state shall send a Transmit COMINIT message to the SP transmitter upon entry into this state.

If this state was entered because of a Disable Phy request, this state shall ignore COMINIT Detected messages until this state is re-entered due to a power on, hard reset, or Management Reset request.

If the phy supports SATA port selectors and this state receives a COMWAKE Detected message, this state shall:

a) set the COMWAKE_Received state machine variable to one; and
b) if the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3):
    A) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response and the SMP DISCOVER LIST response; and
    B) send a SATA Port Selector Change confirmation to the link layer.

6.8.3.2.2 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.

6.8.3.2.3 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.

6.8.3.2.4 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.

6.8.3.3 SP1:OOB_AwaitCOMX state

6.8.3.3.1 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.

If the phy supports SATA port selectors and this state receives a COMWAKE Detected message, this state shall:

a) set the COMWAKE_Received state machine variable to one; and

- b) if the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3):
  - A) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response and the SMP DISCOVER LIST response; and
  - B) send a SATA Port Selector Change confirmation to the link layer.

6.8.3.3.2 Transition SP1:OOB_AwaitCOMX to SP0:OOB_COMINIT

This transition shall occur if the Hot-Plug Timeout timer expires.

If the COMWAKE_Received state machine variable is set to zero and the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3), the state machine shall, before the transition:

- a) set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response and the SMP DISCOVER LIST response; and
- b) send a SATA Port Selector Change confirmation to the link layer.

6.8.3.3.3 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.

If the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3), the state machine shall, before the transition:

- a) set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response and the SMP DISCOVER LIST response; and
- b) send a SATA Port Selector Change confirmation to the link layer.

6.8.3.4 SP2:OOB_NoCOMSASTimeout state

6.8.3.4.1 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.

6.8.3.4.2 Transition SP2:OOB_NoCOMSASTimeout to SP0:OOB_COMINIT

This transition shall occur if the Hot-Plug Timeout timer expires.

6.8.3.4.3 Transition SP2:OOB_NoCOMSASTimeout to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Detected message.

6.8.3.5 SP3:OOB_AwaitCOMINIT_Sent state

6.8.3.5.1 State description

This state waits for a COMINIT Transmitted message.

If the phy supports SATA port selectors, the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3), and this state receives a COMWAKE Detected message, this state shall:

- a) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response and the SMP DISCOVER LIST response; and
- b) send a SATA Port Selector Change confirmation to the link layer.

6.8.3.5.2 Transition SP3:OOB_AwaitCOMINIT_Sent to SP4:OOB_COMSAS

This transition shall occur after receiving a COMINIT Transmitted message.

6.8.3.6 SP4:OOB_COMSAS state

6.8.3.6.1 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.

If the phy supports SATA port selectors, the ATTACHED SATA PORT SELECTOR bit is set to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3), and this state receives a COMWAKE Detected message, this state shall:

a) set the ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response and the SMP DISCOVER LIST response; and
b) send a SATA Port Selector Change confirmation to the link layer.

6.8.3.6.2 Transition SP4:OOB_COMSAS to SP5:OOB_AwaitCOMSAS_Sent

This transition shall occur if this state receives a COMSAS Detected message or this state was entered with a COMSAS Detected argument, and this state has not received a COMSAS Transmitted message.

If the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3), the state machine shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response and the SMP DISCOVER LIST response and send a SATA Port Selector Change confirmation to the link layer before the transition.

6.8.3.6.3 Transition SP4:OOB_COMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur if this state receives both a COMSAS Transmitted message and a COMSAS Detected message.

If the ATTACHED SATA PORT SELECTOR bit is set to one in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3), the state machine shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response and the SMP DISCOVER LIST response and send a SATA Port Selector Change confirmation to the link layer before the transition.

6.8.3.6.4 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.

6.8.3.7 SP5:OOB_AwaitCOMSAS_Sent state

6.8.3.7.1 State description

This state waits for receipt of a COMSAS Transmitted message.

6.8.3.7.2 Transition SP5:OOB_AwaitCOMSAS_Sent to SP6:OOB_AwaitNoCOMSAS

This transition shall occur after receiving a COMSAS Transmitted message.

If this state received a COMSAS Completed message, it shall include a COMSAS Completed argument with the transition.

6.8.3.8 SP6:OOB_AwaitNoCOMSAS state

6.8.3.8.1 State description

This state machine waits for a COMSAS Completed message, which indicates that COMSAS has been completely received.

6.8.3.8.2 Transition SP6:OOB_AwaitNoCOMSAS to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

6.8.3.8.3 Transition SP6:OOB_AwaitNoCOMSAS to SP8:SAS_Start

This transition shall occur after receiving a COMSAS Completed message, or shall occur if a COMSAS Completed argument was received in the transition.

6.8.3.9 SP7:OOB_AwaitCOMSAS state

6.8.3.9.1 State description

Upon entry into this state the COMSAS Detect Timeout timer shall be initialized and started.

6.8.3.9.2 Transition SP7:OOB_AwaitCOMSAS to SP2:OOB_NoCOMSASTimeout

This transition shall occur if the phy does not support SATA and the COMSAS Detect Timeout timer expires.

The state machine shall set the MgmtReset state machine variable to zero before the transition.

6.8.3.9.3 Transition SP7:OOB_AwaitCOMSAS to SP6:OOB_AwaitNoCOMSAS

This transition shall occur after receiving a COMSAS Detected message.

The state machine shall set the MgmtReset state machine variable to zero before the transition.

The state machine shall set the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3). If the ATTACHED SATA PORT SELECTOR bit in the SMP DISCOVER response and the SMP DISCOVER LIST response was set to one prior to this transition, the state machine shall send a SATA Port Selector Change confirmation to the link layer before the transition.

6.8.3.9.4 Transition SP7:OOB_AwaitCOMSAS to SP16:SATA_COMWAKE

This transition shall occur if:

a) the phy supports SATA;
b) the COMSAS Detect Timeout timer expires; and
    A) the MgmtReset state machine variable is set to one; or
    B) the phy does not implement SATA spinup hold.

The state machine shall set the MgmtReset state machine variable to zero before the transition.

The state machine shall set the ATTACHED SATA DEVICE bit to one in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3) before the transition.

6.8.3.9.5 Transition SP7:OOB_AwaitCOMSAS to SP26:SATA _SpinupHold

This transition shall occur if:

a) the phy supports SATA;
b) the COMSAS Detect Timeout timer expires;
c) the phy implements SATA spinup hold; and
d) the MgmtReset state machine variable is set to zero.

The state machine shall set the ATTACHED SATA DEVICE bit to one in the SMP DISCOVER response (see 10.4.3.7) and the SMP DISCOVER LIST response (see 10.4.3.13.3) before the transition.

6.8.4 SAS speed negotiation states

6.8.4.1 SAS speed negotiation states overview

Figure 16 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.

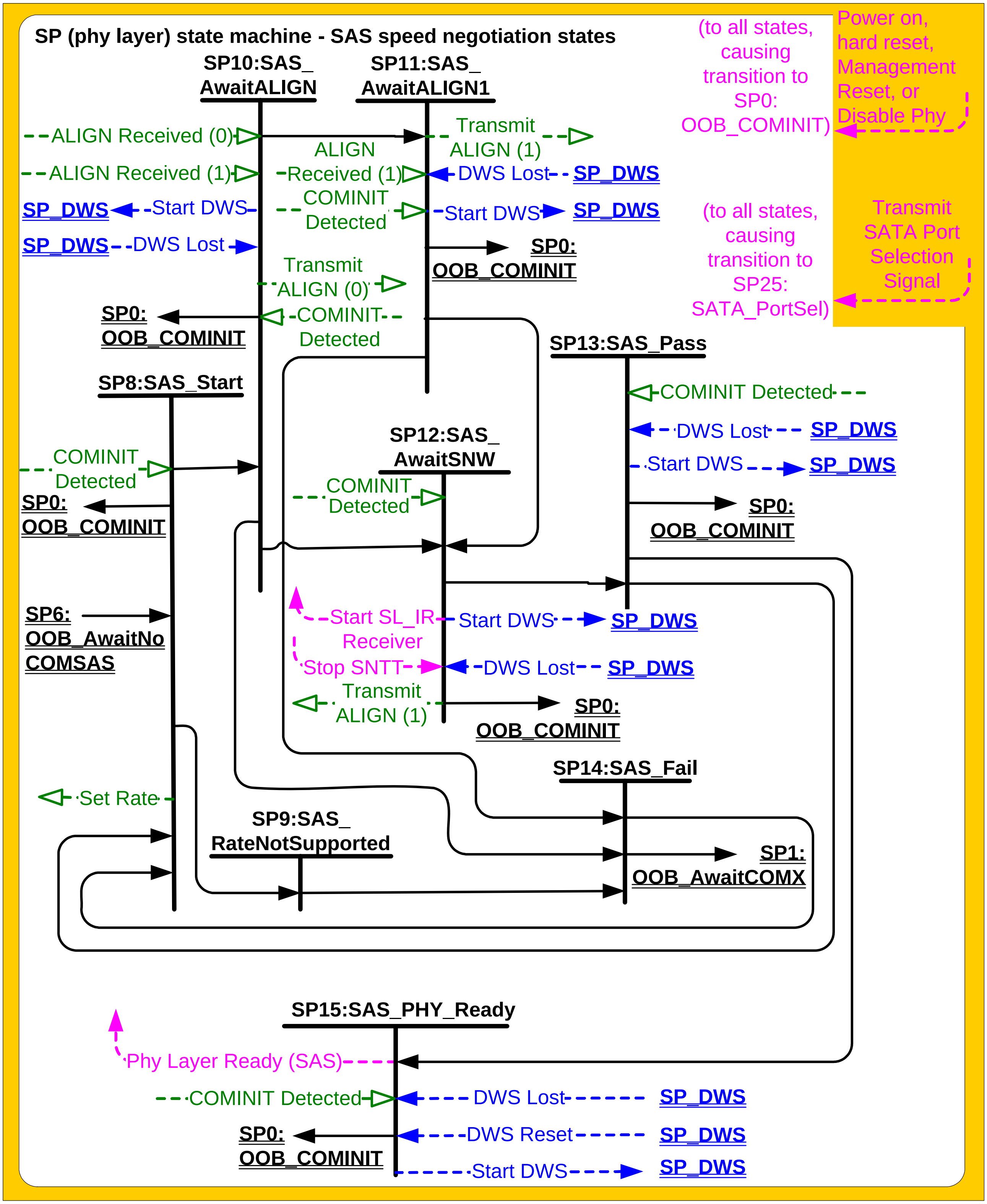


Figure 16 — SP (phy layer) state machine - SAS speed negotiation states

6.8.4.2 SP8:SAS_Start state

6.8.4.2.1 State description

This is the initial state for the SAS speed negotiation sequence.

Upon entry into this state, this state shall:

a) initialize and start the RCDT timer;
b) send a Set Rate message to the SP transmitter with the argument set to:
   A) 1.5 Gbps, if the transition into this state was from the SP6:OOB_AwaitNoCOMSAS state (i.e., if this is the first speed negotiation window); or
   B) the value of the SAS Speed Negotiation Window Rate argument.

During this state D.C. idle shall be transmitted.

6.8.4.2.2 Transition SP8:SAS_Start to SP0:OOB_COMINIT

This transition shall occur after receiving a COMINIT Detected message.

6.8.4.2.3 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.

6.8.4.2.4 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.

6.8.4.3 SP9:SAS_RateNotSupported state

6.8.4.3.1 State description

Upon entry into this state the SNTT timer shall be initialized and started.

During this state D.C. idle shall be transmitted.

6.8.4.3.2 Transition SP9:SAS_RateNotSupported to SP14:SAS_Fail

This transition shall occur after the SNTT timer expires.

6.8.4.4 SP10:SAS_AwaitALIGN state

6.8.4.4.1 State description

Upon entry into 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 entry into this state, this state shall send a Start DWS message to the SP_DWS state machine.

6.8.4.4.2 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, or after receiving a COMINIT Detected message.

6.8.4.4.3 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.

6.8.4.4.4 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.

6.8.4.4.5 Transition SP10:SAS_AwaitALIGN to SP14:SAS_Fail

This transition shall occur if the SNTT timer expires.

6.8.4.5 SP11:SAS_AwaitALIGN1 state

6.8.4.5.1 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.

6.8.4.5.2 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, or after receiving a COMINIT Detected message.

6.8.4.5.3 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 achieve dword synchronization in the current speed negotiation window.

6.8.4.5.4 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 achieve dword synchronization in the current speed negotiation window.

6.8.4.6 SP12:SAS_AwaitSNW state

6.8.4.6.1 State description

This state shall repeatedly send Transmit ALIGN (1) messages to the SP transmitter.

If this is the final 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.

6.8.4.6.2 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, or after receiving a COMINIT Detected message.

6.8.4.6.3 Transition SP12:SAS_AwaitSNW to SP13:SAS_Pass

This transition shall occur after the SNTT timer expires or after receiving a Stop SNTT request.

6.8.4.7 SP13:SAS_Pass state

6.8.4.7.1 State description

This state determines if:

a) another SAS speed negotiation window is required; or

b) the SAS speed negotiation sequence is complete.

Each time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

6.8.4.7.2 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, or after receiving a COMINIT Detected message.

6.8.4.7.3 Transition SP13:SAS_Pass to SP8:SAS_Start

This transition shall occur if this is not the final speed negotiation window.

This transition shall include a SAS Speed Negotiation Window Rate argument with the transition set to the next higher speed negotiation window rate.

6.8.4.7.4 Transition SP13:SAS_Pass to SP15:SAS_PHY_Ready

This transition shall occur if this is the final speed negotiation window.

6.8.4.8 SP14:SAS_Fail state

6.8.4.8.1 State description

This state determines if:

a) another SAS speed negotiation window is required; or
b) the SAS speed negotiation sequence is complete.

6.8.4.8.2 Transition SP14:SAS_Fail to SP1:OOB_AwaitCOMX

This transition shall occur if:

a) the current speed negotiation window is the maximum SAS speed negotiation window and the previous speed negotiation window was not successful; or
b) the current speed negotiation window is the final SAS speed negotiation window.

6.8.4.8.3 Transition SP14:SAS_Fail to SP8:SAS_Start

If the previous SAS speed negotiation window was successful, this transition shall occur and shall include:

a) a SAS Speed Negotiation Window Rate argument set to the previous speed negotiation window rate; and
b) a Final SAS Speed Negotiation Window argument.

If the previous SAS speed negotiation window failed and the current speed negotiation window is not the maximum SAS speed negotiation window, this transition shall occur and shall include a SAS Speed Negotiation Window Rate argument set to the next higher speed negotiation window rate.

6.8.4.9 .SP15:SAS_PHY_Ready state

6.8.4.9.1 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 entry into this state, this state shall:

1) if multiplexing is enabled (see table 14 in x.y), send a Transmit MUX Sequence message to the SP transmitter; and
2) 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.

If the multiplexing is disabled (see table 14 in x.y), each ~~Each~~ time this state receives a DWS Lost message, this state may send a Start DWS message to the SP_DWS state machine to re-acquire dword synchronization without running a new link reset sequence.

> NOTE 3 - If multiplexing is enabled and this state receives a DWS Lost message, this state does not send a Start DWS message and the state machine transitions to SP0:OOB_COMINIT.

6.8.4.9.2 Transition SP15:SAS_PHY_Ready to SP0:OOB_COMINIT

This transition shall occur after:

a) receiving a DWS Lost message, if this state does not send a Start DWS message (e.g., if multiplexing is enabled);
b) receiving a DWS Lost message followed by a COMINIT Detected message, if this state does not send a Start DWS message; or
c) receiving a DWS Reset message.

This transition may but should not occur after receiving a COMINIT Detected message before receiving a DWS Lost message, or after receiving a COMINIT Detected message after sending a Start DWS message (i.e., the SP state machine should ignore COMINIT Detected messages unless the SP_DWS state machine has indicated loss of dword synchronization).

...

6.8.5.8 SP22:SATA_PHY_Ready state

6.8.5.8.1 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.

6.8.5.8.2 Transition SP22:SATA_PHY_Ready to SP0:OOB_COMINIT

This transition shall occur after:

a) receiving a DWS Lost message, if this state does not send a Start DWS message;
a) receiving a DWS Lost message followed by a COMINIT Detected message, if this state does not send a Start DWS message; or
b) receiving a DWS Reset message.

This transition may but should not occur after receiving a COMINIT Detected message before receiving a DWS Lost message, or after receiving a COMINIT Detected message after sending a Start DWS message (i.e., the SP state machine should ignore COMINIT Detected messages unless the SP_DWS state machine has indicated loss of dword synchronization).

6.8.5.8.3 Transition SP22:SATA_PHY_Ready to SP23:SATA_PM_Partial

This transition shall occur after receiving an Enter Partial request.

6.8.5.8.4 Transition SP22:SATA_PHY_Ready to SP24:SATA_PM_Slumber

This transition shall occur after receiving an Enter Slumber request.

6.9 SP_DWS (phy layer dword synchronization) state machine

6.9.1 SP_DWS state machine overview

Each phy includes an SP_DWS state machine and an SP_DWS receiver.

The SP_DWS state machine establishes the same dword boundaries at the receiver as at the attached transmitter by searching for control characters. The SP_DWS receiver monitors and decodes the incoming data stream and forces K28.5 characters into the first character position to effectively perform dword alignment when requested by the SP_DWS state machine. K28.5 characters with either positive or negative disparity shall be accepted. The SP_DWS receiver continues to reestablish dword alignment by forcing received K28.5 characters into the first character position until a K28.5-based primitive (i.e., K28.5, Dxx.y, Dxx.y, Dxx.y) with correct disparity on each data character is detected. The resultant primitives, dwords and valid dword indicators (e.g., encoding error indicators) are sent to this state machine to enable it to determine the dword synchronization policy.

After dword synchronization has been achieved, this state machine 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 6.9.3)(initial state);
b) SP_DWS1:Valid1 (see 6.9.4);
c) SP_DWS2:Valid2 (see 6.9.5);
d) SP_DWS3:SyncAcquired (see 6.9.6);
e) SP_DWS4:Lost1 (see 6.9.7);
f) SP_DWS5:Lost1Recovered (see 6.9.8);
g) SP_DWS6:Lost2 (see 6.9.9);
h) SP_DWS7:Lost2Recovered (see 6.9.10);
i) SP_DWS8:Lost3 (see 6.9.11); and
j) SP_DWS9:Lost3Recovered (see 6.9.12).

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 requesting a phy operation of LINK RESET or HARD RESET in an expander device);
d) receiving a Disable Phy request from the management layer (e.g., from the SMP PHY CONTROL function requesting a phy operation of DISABLE in an expander device); or
e) receiving a Stop DWS message from the SP state machine.

This state machine receives the following messages from the SP state machine (see 6.8):

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.

The SP_DWS state machine shall maintain the timers listed in table 87.

Table 17 — SP_DWS timers

Timer	Initial value
DWS Reset Timeout timer	1 ms

Figure 17 shows the SP_DWS state machine.

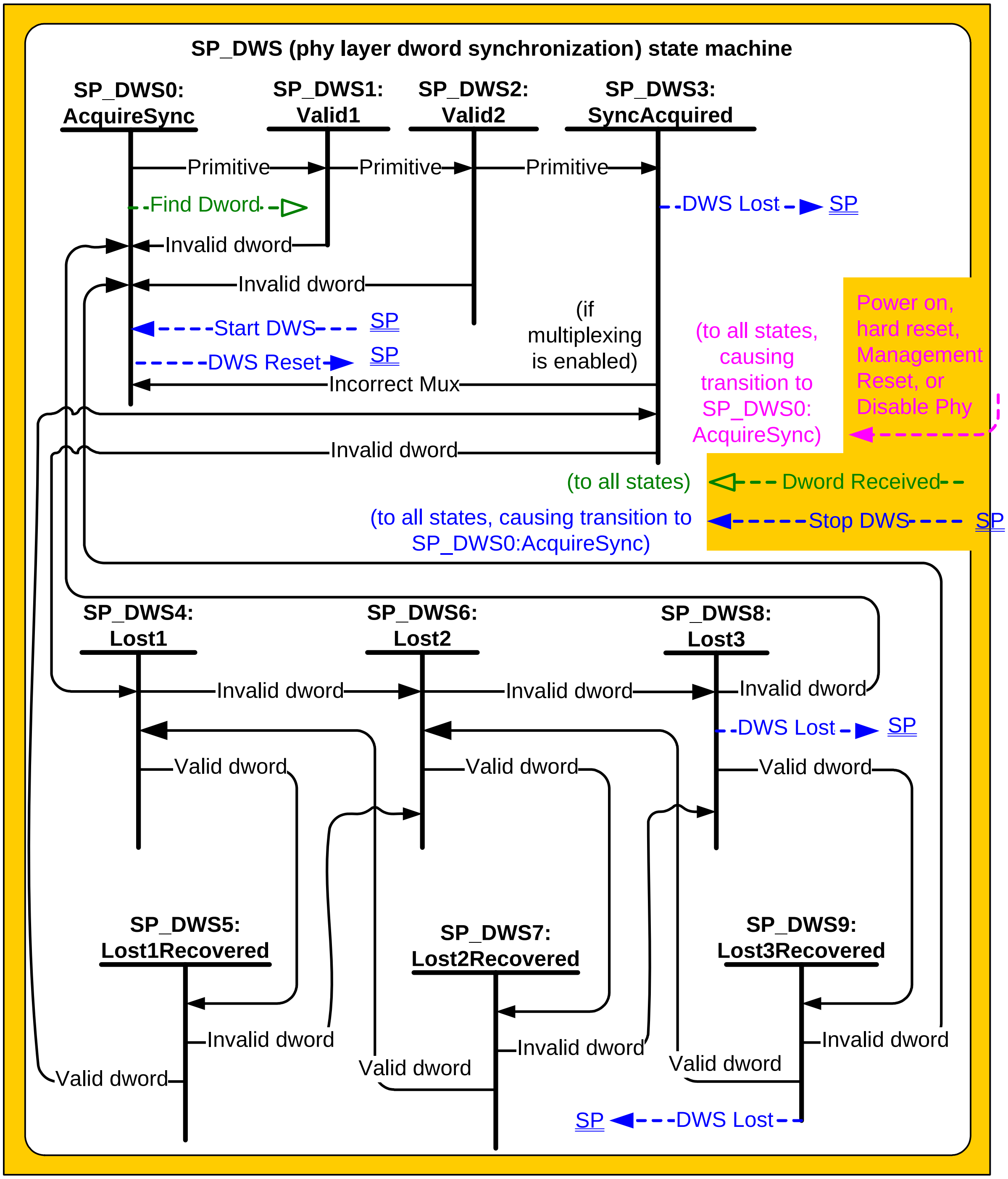


Figure 17 — SP_DWS (phy layer dword synchronization) state machine [updated]

6.9.2 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 (Primitive);
b) Dword Received (Data Dword); ~~and~~
c) Dword Received (Invalid); and
d) Incorrect Mux Received.

The SP_DWS receiver also sends Dword Received confirmations to the link layer state machine receivers (e.g., SL_IR, SL, SSP, SMP, and XL). If multiplexing is enabled (see table 14 in x.y), the SP_DWS receiver shall use the first incoming MUX primitive to determine the logical phy to which it sends each Dword Received confirmation and shall not send an Dword Received confirmations until it receives the first incoming MUX primitive.

Upon receiving a Find Dword message, the SP_DWS receiver shall monitor the input data stream and force each K28.5 character detected into the first character position of a possible dword. If the next three characters are data characters with correct disparity, it shall send the dword as a Dword Received (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 3.1.104);
b) send a Dword Received (Primitive) message to the SP_DWS state machine if the dword is a primitive (i.e., the dword contains a K28.5 character in the first character position followed by three data characters); or
c) send a Dword Received (Data Dword) message to the SP_DWS state machine if the dword is a data dword (i.e., it is not an invalid dword or a primitive).

6.9.3 SP_DWS0:AcquireSync state

6.9.3.1 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.

6.9.3.2 Transition SP_DWS0:AcquireSync to SP_DWS1:Valid1

This transition shall occur after sending a Find Dword message and receiving a Dword Received (Primitive) message.

6.9.4 SP_DWS1:Valid1 state

6.9.4.1 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.

6.9.4.2 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.

6.9.4.3 Transition SP_DWS1:Valid1 to SP_DWS2:Valid2

This transition shall occur after receiving a Dword Received (Primitive) message.

6.9.5 SP_DWS2:Valid2 state

6.9.5.1 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.

6.9.5.2 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.

6.9.5.3 Transition SP_DWS2:Valid2 to SP_DWS3:SyncAcquired

This transition shall occur after receiving a Dword Received (Primitive) message.

6.9.6 SP_DWS3:SyncAcquired state

6.9.6.1 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, or an Incorrect Mux Received message, which indicates that dword synchronization is lost.

If an Incorrect Mux Received message is received, this state shall send a DWS Lost message to the SP state machine.

6.9.6.2 Transition SP_DWS3:SyncAcquired to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

6.9.6.3 Transition SP_DWS3:SyncAcquired to SP_DWS4:Lost1

This transition shall occur after receiving a Dword Received (Invalid) message.

6.9.7 SP_DWS4:Lost1 state

6.9.7.1 State description

This state is reached when one invalid dword has been received and not nullified. This state waits for a Dword Received message.

6.9.7.2 Transition SP_DWS4:Lost1 to SP_DWS5:Lost1Recovered

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

6.9.7.3 Transition SP_DWS4:Lost1 to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Invalid) message.

6.9.8 SP_DWS5:Lost1Recovered state

6.9.8.1 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.

6.9.8.2 Transition SP_DWS5:Lost1Recovered to SP_DWS3:SyncAcquired

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

6.9.8.3 Transition SP_DWS5:Lost1Recovered to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Invalid) message.

6.9.9 SP_DWS6:Lost2 state

6.9.9.1 State description

This state is reached when two invalid dwords have been received and not nullified. This state waits for a Dword Received message.

6.9.9.2 Transition SP_DWS6:Lost2 to SP_DWS7:Lost2Recovered

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

6.9.9.3 Transition SP_DWS6:Lost2 to SP_DWS8:Lost3

This transition shall occur after receiving a Dword Received (Invalid) message.

6.9.10 SP_DWS7:Lost2Recovered state

6.9.10.1 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.

6.9.10.2 Transition SP_DWS7:Lost2Recovered to SP_DWS4:Lost1

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

6.9.10.3 Transition SP_DWS7:Lost2Recovered to SP_DWS8:Lost3

This transition shall occur after receiving a Dword Received (Invalid) message.

6.9.11 SP_DWS8:Lost3 state

6.9.11.1 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.

6.9.11.2 Transition SP_DWS8:Lost3 to SP_DWS9:Lost3Recovered

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

6.9.11.3 Transition SP_DWS8:Lost3 to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

6.9.12 SP_DWS9:Lost3Recovered state

6.9.12.1 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.

6.9.12.2 Transition SP_DWS9:Lost3Recovered to SP_DWS6:Lost2

This transition shall occur after receiving a Dword Received (Data Dword) message or a Dword Received (Primitive) message.

6.9.12.3 Transition SP_DWS9:Lost3Recovered to SP_DWS0:AcquireSync

This transition shall occur after sending a DWS Lost message.

Changes to chapter 7 (link layer)

Define multiplexing and the MUX primitives.

7.2 Primitives

7.2.2 Primitive summary

Table 76 defines the deletable primitives.

Table 76 — Deletable primitives

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [c]
		I	E	T	I	E	T	
ALIGN (0)	All	I	E	T	I	E	T	Single
ALIGN (1)		I	E	T	I	E	T	
ALIGN (2)		I	E	T	I	E	T	
ALIGN (3)		I	E	T	I	E	T	
MUX (LOGICAL LINK 0)	NoConn	I	E	T	I	E	T	Single
MUX (LOGICAL LINK 1)		I	E	T	I	E	T	
NOTIFY (ENABLE SPINUP)	All	I	E				T	Single
NOTIFY (POWER FAILURE EXPECTED)		I	E				T	
NOTIFY (RESERVED 1)					I	E	T	
NOTIFY (RESERVED 2)					I	E	T	

[a] The Use column indicates when the primitive is used:
a) NoConn: SAS ~~physical~~logical links, outside connections;
b) Conn: SAS ~~physical~~logical links, inside connections;
c) All: SAS ~~physical~~logical links, both outside connections or inside any type of connection; or
d) STP: SAS ~~physical~~logical 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 continued primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

Table 76 defines the primitives not specific to the type of connection.

Table 76 — Primitives not specific to type of connection (part 1 of 2)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [c]
		I	E	T	I	E	T	
AIP (NORMAL)	NoConn		E		I	E	T	Single
AIP (RESERVED 0)					I	E	T	
AIP (RESERVED 1)					I	E	T	
AIP (RESERVED 2)					I	E	T	
AIP (RESERVED WAITING ON PARTIAL)					I	E	T	
AIP (WAITING ON CONNECTION)			E		I	E	T	
AIP (WAITING ON DEVICE)			E		I	E	T	
AIP (WAITING ON PARTIAL)			E		I	E	T	
~~ALIGN (0)~~	~~All~~	~~I~~	~~E~~	~~T~~	~~I~~	~~E~~	~~T~~	~~Single~~
~~ALIGN (1)~~		~~I~~	~~E~~	~~T~~	~~I~~	~~E~~	~~T~~	
~~ALIGN (2)~~		~~I~~	~~E~~	~~T~~	~~I~~	~~E~~	~~T~~	
~~ALIGN (3)~~		~~I~~	~~E~~	~~T~~	~~I~~	~~E~~	~~T~~	
BREAK	All	I	E	T	I	E	T	Redundant
BROADCAST (CHANGE)	NoConn	I	E		I	E	T	
BROADCAST (SES)				T	I	E	T	
BROADCAST (RESERVED 1)					I	E	T	
BROADCAST (RESERVED 2)					I	E	T	
BROADCAST (RESERVED 3)					I	E	T	
BROADCAST (RESERVED 4)					I	E	T	
BROADCAST (RESERVED CHANGE 0)					I	E	T	
BROADCAST (RESERVED CHANGE 1)					I	E	T	
CLOSE (CLEAR AFFILIATION)	STP	I					T	Triple
CLOSE (NORMAL)	Conn	I		T	I		T	
CLOSE (RESERVED 0)	Conn				I		T	
CLOSE (RESERVED 1)	Conn				I		T	
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 (POWER FAILURE EXPECTED)~~		~~I~~	~~E~~				~~T~~	
~~NOTIFY (RESERVED 1)~~					~~I~~	~~E~~	~~T~~	
~~NOTIFY (RESERVED 2)~~					~~I~~	~~E~~	~~T~~	
OPEN_ACCEPT	NoConn	I		T	I		T	Single

Table 76 — Primitives not specific to type of connection (part 2 of 2)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [c]
		I	E	T	I	E	T	
OPEN_REJECT (BAD DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)		I	E	T	I		T	
OPEN_REJECT (NO DESTINATION)			E		I		T	
OPEN_REJECT (PATHWAY BLOCKED)			E		I		T	
OPEN_REJECT (PROTOCOL NOT SUPPORTED)		I		T	I		T	
OPEN_REJECT (RESERVED ABANDON 0)					I		T	
OPEN_REJECT (RESERVED ABANDON 1)					I		T	
OPEN_REJECT (RESERVED ABANDON 2)					I		T	
OPEN_REJECT (RESERVED ABANDON 3)					I		T	
OPEN_REJECT (RESERVED CONTINUE 0)					I		T	
OPEN_REJECT (RESERVED CONTINUE 1)					I		T	
OPEN_REJECT (RESERVED INITIALIZE 0)					I		T	
OPEN_REJECT (RESERVED INITIALIZE 1)					I		T	
OPEN_REJECT (RESERVED STOP 0)					I		T	
OPEN_REJECT (RESERVED STOP 1)					I		T	
OPEN_REJECT (RETRY)		I		T	I		T	
OPEN_REJECT (STP RESOURCES BUSY)			E	T	I			
OPEN_REJECT (WRONG DESTINATION)		I		T	I		T	
SOAF	NoConn	I	E	T	I	E	T	Single

[a] The Use column indicates when the primitive is used:

a) NoConn: SAS ~~physical~~logical links, outside connections;
b) Conn: SAS ~~physical~~logical links, inside connections;
c) All: SAS ~~physical~~logical links, both outside connections or inside any type of connection; or
d) STP: SAS ~~physical~~logical 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 continued primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

Editor's Note 7: Change “SAS physical links” to “SAS logical links” in table 77

Editor's Note 8: Change “SAS physical links” to “SAS logical links” in table 78

7.2.3 Primitive encodings

Table 77 defines the primitive encoding for deletable primitives.

Table 77 — Primitive encoding for deletable primitives

Primitive	Character			
	1st	2nd	3rd	4th (last)
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
MUX (LOGICAL LINK 0)	K28.5	D02.0	D16.7	D31.4
MUX (LOGICAL LINK 1)	K28.5	D02.0	D29.7	D16.7
NOTIFY (ENABLE SPINUP)	K28.5	D31.3	D31.3	D31.3
NOTIFY (POWER FAILURE EXPECTED)	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

Table 78 defines the primitive encoding for primitives not specific to type of connection.

Table 78 — Primitive encoding for primitives not specific to type of connection (part 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 (SES)	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

Table 78 — Primitive encoding for primitives not specific to type of connection (part 2 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
~~NOTIFY (ENABLE SPINUP)~~	~~K28.5~~	~~D31.3~~	~~D31.3~~	~~D31.3~~
~~NOTIFY (POWER FAILURE EXPECTED)~~	~~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

7.2.4 Primitive sequences

7.2.4.1 Primitive sequences overview

...

Any number of ~~ALIGNs and NOTIFYs~~deletable primitives may be sent inside primitive sequences without affecting the count or breaking the consecutiveness requirements. Rate matching ~~ALIGNs and NOTIFYs~~deletable primitives 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 (e.g., RRDY, SATA_SOF) shall be transmitted one time to form a single primitive sequence.

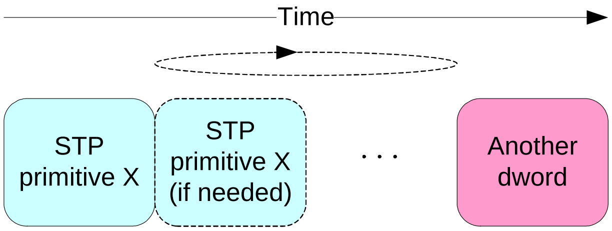
Receivers count each primitive received that is labeled as a single primitive sequence as a distinct single primitive sequence.

ALIGNs, NOTIFYs, and MUXs are called deletable primitives (see 7.3).

7.2.4.3 Repeated primitive sequence

Primitives that form repeated primitive sequences (e.g., SATA_PMACK) shall be transmitted one or more times. Only STP primitives form repeated primitive sequences. ~~ALIGNs and NOTIFYs~~Any number of deletable primitives may be sent inside repeated primitive sequences as described in 7.2.4.1.

Figure 18 shows an example of transmitting a repeated primitive sequence.

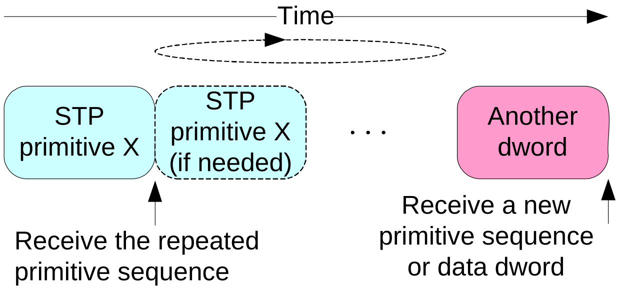


NOTE: Another dword is a dword other than a deletable primitive or STP primitive X

Figure 18 — Transmitting a repeated primitive sequence [changed]

Receivers do not count the number of times a repeated primitive is received (i.e., receivers are simply in the state of receiving the primitive).

Figure 19 shows an example of receiving a repeated primitive sequence.


Time
STP primitive X
STP primitive X (if needed)
. . .
Another dword
Receive the repeated primitive sequence
Receive a new primitive sequence or data dword

NOTE: Another dword is a dword other than a deletable primitive or STP primitive X

Figure 19 — Receiving a repeated primitive sequence [changed]

7.2.4.4 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be transmitted as specified in 7.17.4. ~~ALIGNs and NOTIFYs~~Any number of deletable primitives may be sent inside continued primitive sequences as described in 7.2.4.1.

7.2.4.5 Triple primitive sequence

Primitives that form triple primitive sequences (e.g., CLOSE (NORMAL)) shall be sent three times consecutively. ~~ALIGNs and NOTIFYs~~Any number of deletable primitives 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) a~~n ALIGN or NOTIFY~~deletable primitive.

Figure 20 shows examples of triple primitive sequences.

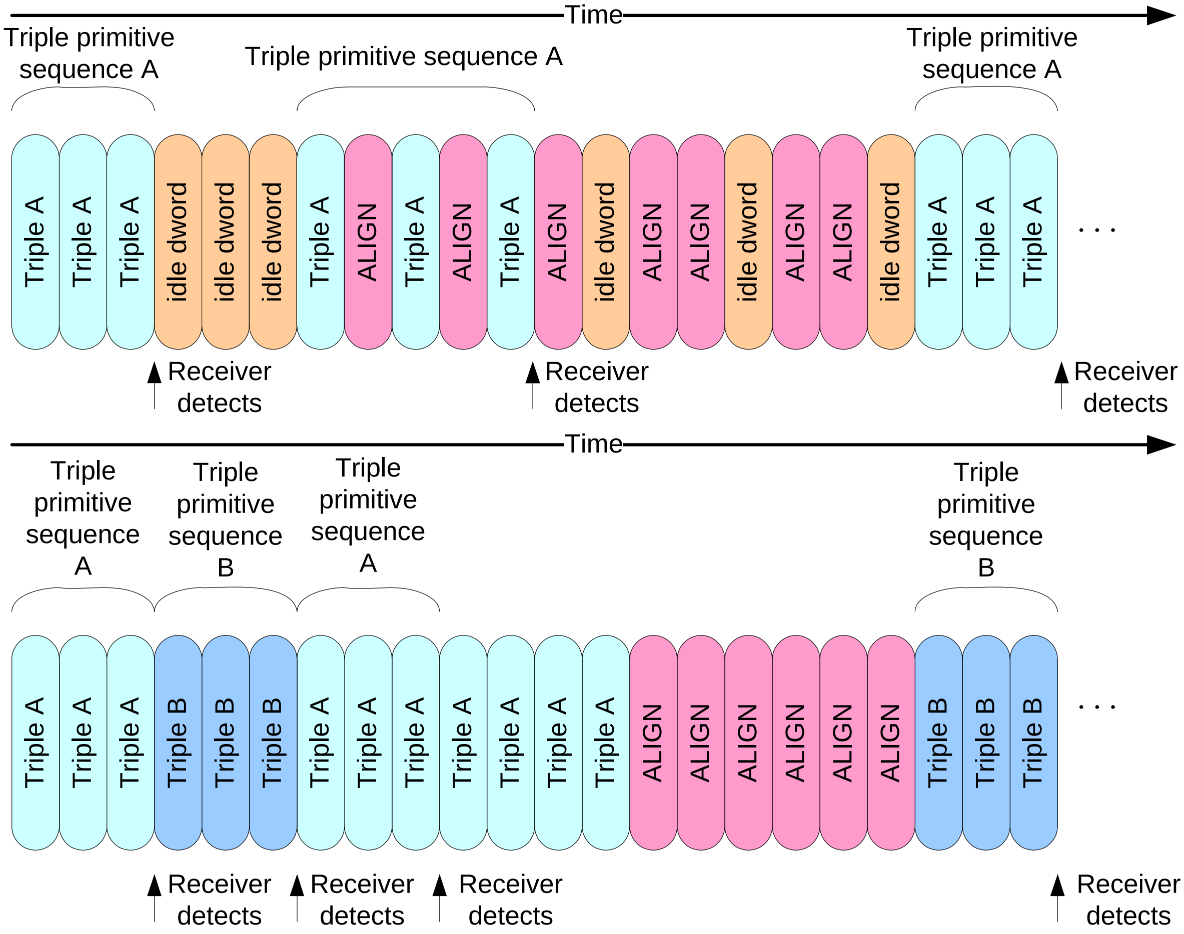


Figure 20 — Triple primitive sequence [no changes]

7.2.4.6 Redundant primitive sequence

Primitives that form redundant primitive sequences (e.g., BROADCAST (CHANGE)) shall be sent six times consecutively. ~~ALIGNs and NOTIFYs~~Any number of deletable primitives 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 any three of six 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) a~~n ALIGN or NOTIFY~~deletable primitive.

Figure 21 shows examples of redundant primitive sequences.

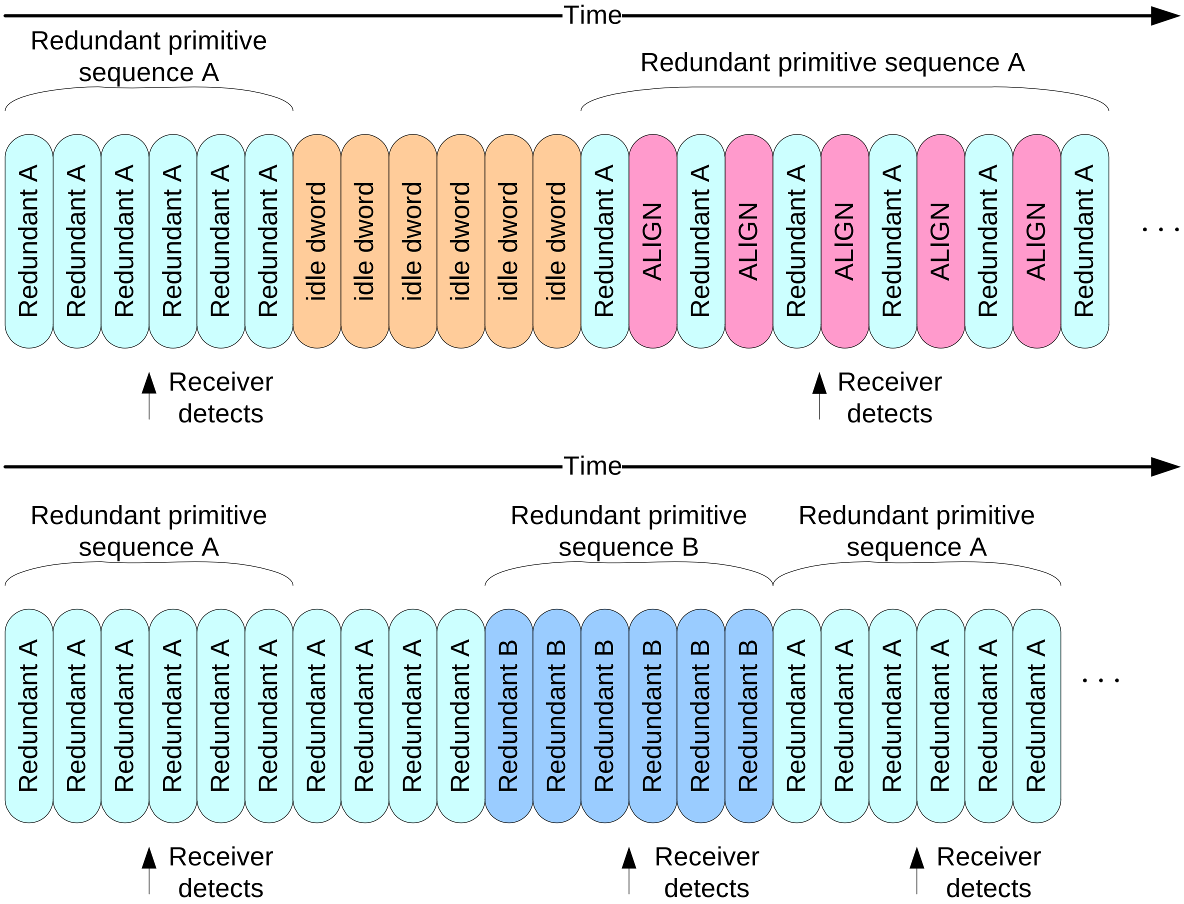

Time
Redundant primitive sequence A
Redundant primitive sequence A
Redundant A
idle dword
ALIGN
. . .
Receiver detects
Receiver detects
Time
Redundant primitive sequence A
Redundant primitive sequence B
Redundant primitive sequence A
Redundant A
Redundant B
. . .
Receiver detects
Receiver detects
Receiver detects

Figure 21 — Redundant primitive sequence [no changes]

7.2.5 Primitives not specific to type of 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 (see 7.3);
d) rate matching during connections (see 7.13); and
e) STP initiator phy throttling during STP connections (see).

ALIGNs are deletable primitives (see 7.3).

Table 79 defines the different versions of ALIGN primitives.

Table 79 — ALIGN primitives

Primitive	Description
ALIGN (0)	Used for OOB signals, the speed negotiation sequence, clock skew management, rate matching, and STP initiator phy throttling.
ALIGN (1)	Used for the speed negotiation sequence, clock skew management and rate matching, and STP initiator phy throttling.
ALIGN (2)	Used for clock skew management, rate matching, and STP initiator phy throttling.
ALIGN (3)	Used for clock skew management, rate matching, and STP initiator phy throttling.

Phys shall use ALIGN (0) to construct OOB signals as described in 6.6. Phys shall use ALIGN (0) and ALIGN (1) during the speed negotiation sequence as described in 6.7.4.2. Phys shall rotate through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) for all ALIGNs sent after the phy reset sequence.

Phys receiving ALIGNs after the phy reset sequence shall not verify the rotation and shall accept any of the ALIGNs at any time.

Phys shall only detect an ALIGN after decoding all four characters in the primitive.

NOTE 4 - 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, rate matching, and STP initiator phy throttling, ALIGNs may be replaced by NOTIFYs (see) or MUXs (see 7.2.5.n). ALIGNs shall not be replaced by NOTIFYs or MUXs during OOB signals and speed negotiation.

7.2.5.7 ERROR

ERROR should be sent by an expander device when it is forwarding dwords from a SAS ~~physical~~logical link or SATA physical link to a SAS ~~physical~~logical link and it receives an invalid dword or an ERROR.

...

7.2.5.n MUX (Multiplex)

MUX is used if multiplexing is enabled (see table 14 in x.y) as follows:

a) transmitted during the multiplexing sequence (see 7.xx); and
b) substituted for an ALIGN (see 7.2.5.2) being transmitted for clock skew management (see 7.3), rate matching (see 7.13), or STP initiator phy throttling (see 7.17.2) to confirm the logical link number.

Substitution of a MUX for an ALIGN may or may not affect the ALIGN rotation (i.e., the MUX may take the place of one of the ALIGNs in the rotation through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3), or it may delay the rotation).

MUXs are deletable primitives (see 7.3).

The versions of MUX are defined in table 80.

Table 80 — MUX primitives

Primitive	Description
MUX (LOGICAL LINK 0)	Establishes the position of dwords in logical link 0.
MUX (LOGICAL LINK 1)	Establishes the position of dwords in logical link 1.

See 7.xx for details on multiplexing.

7.2.5.9 NOTIFY

7.2.5.9.1 NOTIFY overview

NOTIFY may be transmitted in place of any ALIGN (see) being transmitted for clock skew management (see 7.3), rate matching (see 7.13), or STP initiator phy throttling (see). Substitution of a NOTIFY for an ALIGN may or may not affect the ALIGN rotation (i.e., the NOTIFY may take the place of one of the ALIGNs in the rotation through ALIGN (0), ALIGN (1), ALIGN (2), ~~or~~and ALIGN (3), or it may delay the rotation). A specific NOTIFY shall not be transmitted in more than three consecutive dwords until at least three other dwords have been transmitted.

NOTIFYs are deletable primitives (see 7.3).

NOTIFY shall not be forwarded through expander devices. Expander devices shall substitute an ALIGN for a NOTIFY if necessary.

SAS target devices are not required to detect every transmitted NOTIFY.

The versions of NOTIFY representing different reasons are defined in table 81.

Table 81 — NOTIFY primitives

Primitive	Description	Reference
NOTIFY (ENABLE SPINUP)	Specify to a SAS target device that it may temporarily consume additional power while transitioning into the active or idle power condition state.	7.2.5.9.2
NOTIFY (POWER LOSS EXPECTED)	Specify to a SAS target device that power loss may occur within the time specified by the POWER LOSS TIMEOUT field in the Protocol-Specific Logical Unit mode page (see 10.2.7.3.2).	7.2.5.9.3
NOTIFY (RESERVED 1)	Reserved.	
NOTIFY (RESERVED 2)	Reserved.	

NOTIFY (RESERVED 1) and NOTIFY (RESERVED 2) shall be ignored by all devices.

7.2.5.9.3 NOTIFY (POWER LOSS EXPECTED)

...

If a SAS target device supports NOTIFY (POWER LOSS EXPECTED) and receives NOTIFY (POWER LOSS EXPECTED) on an SSP target port, then each SAS logical phy within the SAS target device shall:

a) if there is an SSP connection, then transmit a BREAK on that connection; and
b) respond to SSP connection requests with OPEN_REJECT (RETRY) until the power loss timeout timer expires or power is lost.

...

7.2.5.11 OPEN_REJECT

...

NOTE 21 - Some SAS logical phys also transmit OPEN_REJECT (RETRY) if they receive an OPEN address frame while their SL_CC state machines are in the SL_CC5:BreakWait state (see 7.14.4.7).

When a SAS logical phy detects more than one reason to transmit an OPEN_REJECT, the SL_CC state machine determines the priority in the SL_CC2:Selected state (see 7.14.4.4).

When an expander logical phy detects more than one reason to transmit an OPEN_REJECT, the ECM determines the priority (see 7.12.4).

See 7.12 for details on connection requests.

7.2.7.1 SATA_ERROR

SATA_ERROR should be sent by an expander device when it is forwarding dwords from a SAS ~~physical~~logical link to a SATA physical link and it receives an invalid dword or an ERROR.

...

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~~logical 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 up to the physical link rate tolerance defined in table 48 (see 5.3.3). 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~~ deletable primitives in the dword stream. Receivers may pass ~~ALIGNs and NOTIFYs~~ deletable primitives through to their internal buffers, or may strip them out when an overrun occurs. Receivers add ~~ALIGNs or NOTIFYs~~ deletable primitives when an underrun occurs. The internal logic shall ignore all ~~ALIGNs and NOTIFYs~~ deletable primitives that arrive in the internal buffers.

Elasticity buffer circuitry, as shown in figure 22, 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 5.3.3. The depth of the elasticity buffer is vendor-specific but shall accommodate the clock skew management ~~ALIGN~~deletable primitive insertion requirements in table 82.

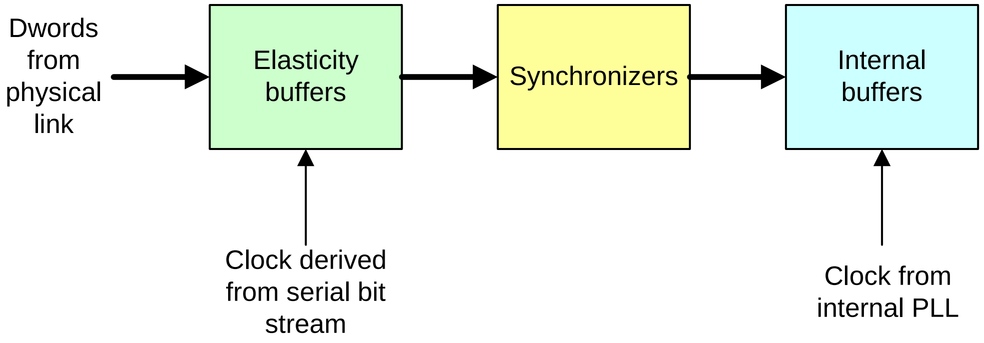


Figure 22 — Elasticity buffers [no changes]

A logical phy that is the original source for the dword stream (i.e., a phy that is not an expander logical phy forwarding dwords from another expander logical phy) shall insert ~~one~~ ~~ALIGN or NOTIFY~~deletable primitives for clock skew management after the phy reset sequence completes as described in table 82.

Table 82 — Clock skew management ~~ALIGN~~deletable primitive insertion requirement

~~Physical~~ Logical link rate	Requirement
1,5 Gbps	One ~~ALIGN or NOTIFY~~deletable primitive within every 2 048 dwords
3,~~0~~ Gbps	Two ~~ALIGNs or NOTIFYs~~deletable primitives within every 4 096 dwords
6 Gbps	Four deletable primitives within every 8 192 dwords

Editor's Note 9: 06-263 changed this in sas2r06 and made the ALIGN rate 1/64 dwords. This proposal defers to that change - it just introduces the term "deletable primitive" in this section

~~ALIGNs and NOTIFYs~~Deletable primitives inserted for clock skew management are in addition to ~~ALIGNs and NOTIFYs~~deletable primitives inserted for rate matching (see 7.13) and STP initiator phy throttling (see). See Annex H for a summary of their combined requirements.

See for details on rotating through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3). NOTIFYs may also be ~~used~~transmitted in place of ALIGNs (see) on SAS ~~physical~~logical links. MUXs may also be transmitted in place of ALIGNs on multiplexed 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 and/or NOTIFYs~~deletable primitives as required to match the transmit and receive connection rates. It is not required to transmit the number of ~~ALIGNs and/or NOTIFYs~~deletable primitives for clock skew management described in table 82 when forwarding to a SAS ~~physical~~logical link. It may increase or reduce that number based on clock frequency differences between the phy transmitting the dwords to the expander device and the expander device's receiving phy.

NOTE 23 - One possible implementation for expander devices forwarding dwords is for the expander device to delete all ~~ALIGNs and NOTIFYs~~deletable primitives received and to insert ~~ALIGNs and/or NOTIFYs~~deletable primitives 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 and/or NOTIFYs~~deletable primitives as required to match the transmit and receive connection rates. It is not required to transmit any particular number of ~~ALIGNs and/or NOTIFYs~~deletable primitives for clock skew management when forwarding to a SAS ~~physical~~logical link and is not required to ensure that any ~~ALIGNs and/or NOTIFYs~~deletable primitives it transmits are in pairs.

NOTE 24 - Due to clock skew ~~ALIGN and NOTIFY~~deletable primitive removal, the STP target port may not receive a pair of ~~ALIGNs and/or NOTIFYs~~deletable primitives 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 SATA host port of the STP/SATA bridge. One possible implementation is for the STP/SATA bridge to delete all ~~ALIGNs and NOTIFYs~~deletable primitives received by the STP target port and to insert two consecutive ALIGNs at the SATA host port when its elasticity buffer is empty or when 254 non-ALIGN dwords have been transmitted. It may need to buffer up to 2 dwords concurrently being received by the STP target port while it does so.

7.6 Scrambling

Scrambling is used to reduce the probability of long strings of repeated patterns appearing on the physical link.

All data dwords are scrambled. Table 83 lists the scrambling for different types of data dwords.

Table 83 — Scrambling for different data dword types

Connection state	Data dword type	Description of scrambling
Outside connections	SAS idle dword	When a connection is not open and there are no other dwords to transmit, vendor-specific scrambled data dwords shall be transmitted.
	Address frame	After an SOAF, all data dwords shall be scrambled until the EOAF.
Inside SSP connection	SSP frame	After an SOF, all data dwords shall be scrambled until the EOF.
	SSP idle dword	When there are no other dwords to transmit, vendor-specific scrambled data dwords shall be transmitted.
Inside SMP connection	SMP frame	After an SOF, all data dwords shall be scrambled until the EOF.
	SMP idle dword	When there are no other dwords to transmit, vendor-specific scrambled data dwords shall be transmitted.
Inside STP connection	STP frame	After a SATA_SOF, all data dwords shall be scrambled until the SATA_EOF.
	Continued primitive	After a SATA_CONT, vendor-specific scrambled data dwords shall be sent until a primitive other than ~~ALIGN or NOTIFY~~a deletable primitive is transmitted.

...

7.8 Address frames

7.8.2 IDENTIFY address frame

Table 84 defines the IDENTIFY address frame format used for the identification sequence. The IDENTIFY address frame is sent by each logical phy after the phy reset sequence completes if the physical link is a SAS physical link. The IDENTIFY address frame sent by each logical phy in a physical phy shall be identical.

Table 84 — IDENTIFY address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved	DEVICE TYPE			ADDRESS FRAME TYPE (0h)			
1	Reserved				REASON			
2	Reserved				SSP INITIATOR PORT	STP INITIATOR PORT	SMP INITIATOR PORT	Restricted (for OPEN address frame)
3	Reserved				SSP TARGET PORT	STP TARGET PORT	SMP TARGET PORT	Restricted (for OPEN address frame)
4	DEVICE NAME							
11								
12	SAS ADDRESS							
19								
20	PHY IDENTIFIER							
21	Reserved					INSIDE ZPSDS PERSISTENT	REQUESTED INSIDE ZPSDS	BREAK_REPLY CAPABLE
22	Reserved							
27								
28	(MSB)	CRC						
31								(LSB)

The DEVICE TYPE field specifies the type of device containing the phy, and is defined in table 85.

Table 85 — DEVICE TYPE field

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.

...

7.8.3 OPEN address frame

...

The CONNECTION RATE field specifies the connection rate (see 4.1.10) being requested between the source and destination, and is defined in table 86.

Table 86 — CONNECTION RATE field

Code	Description
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
Bh - Fh	Reserved for future connection rates
All others	Reserved

...

7.xx Multiplexing

If SNW-3 indicates multiplexing is supported, the phy shall begin multiplexing immediately after the speed negotiation sequence and transmit 3 MUX primitives on each logical link as the first part of the identification sequence or hard reset sequence. This is called the multiplexing sequence. The phy shall not transmit deletable primitives for clock skew management (see 7.3) during this time.

If SNW-3 indicates multiplexing is not supported, the phy shall not transmit MUX primitives.

The phy shall align its incoming dword streams based on the first MUX it receives. The phy shall process MUX primitives in logic running off the received clock without using an elasticity buffer, because they are not accompanied by additional deletable primitives (e.g., ALIGNs and/or NOTIFYs).

Figure 25 shows no multiplexing (i.e., multiplexing into one logical link).

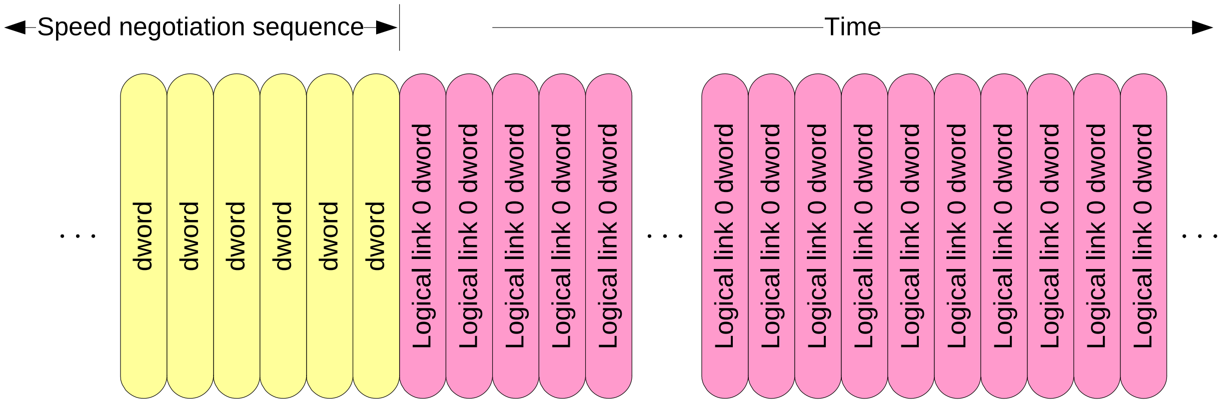


Figure 23 — No multiplexing

Figure 25 shows multiplexing into two logical links.

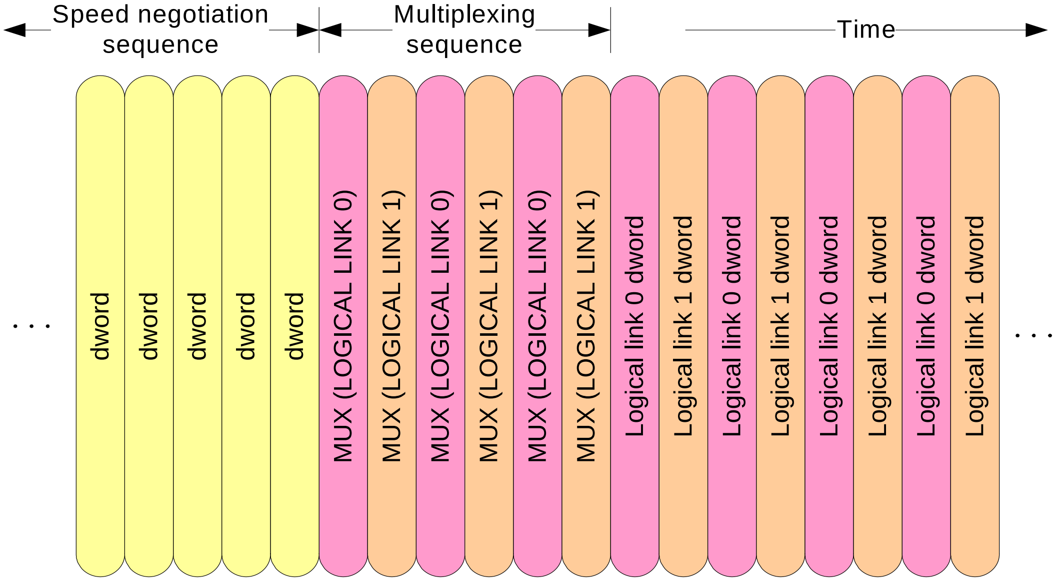


Figure 24 — Multiplexing

After the multiplexing sequence completes, each logical phy shall honor the deletable primitive insertion requirements for clock skew management defined in 7.3. The logical phys shall ignore MUX primitives.

The phy shall assign the incoming logical links to its logical phys based on the received MUX primitives (e.g., MUX (LOGICAL LINK 0) indicates the position of logical link 0).

The phy shall handle errors during the multiplexing sequence (i.e., after receiving the first MUX primitive) as follows:

a) If the phy receives a dword that is not a MUX primitive before receiving the MUX primitive expected in that position, it shall discard the dword;
b) If the phy receives an invalid dword, it shall discard the dword; and
c) If the phy receives a MUX primitive that does not match the MUX primitive expected in that position (i.e., it receives MUX (LOGICAL LINK 1) on logical link 0 or receives MUX (LOGICAL LINK 0) on logical link 1), it shall restart the link reset sequence.

If a phy with multiplexing enabled ever loses dword synchronization, it shall restart a link reset sequence rather than attempt to reestablish dword synchronization.

Once the multiplexing sequence is complete, the phy shall not perform another multiplexing sequence until a new link reset sequence.

Once the multiplexing sequence is complete:

a) a logical phy originating dwords shall transmit MUX as a deletable primitive (e.g., in place on an ALIGN) at least once every millisecond; and
b) a logical phy forwarding dwords should transmit MUX as a deleteable primitive at least once every millisecond to confirm the logical link numbers.

Transmitting NOTIFY has higher priority than transmitting MUX.

NOTE 25 - Period MUX transmission is for the convenience of logic analyzers and to provide additional assurance that the receiving phy is in agreement with the transmitting phy.

During or after the multiplexing sequence, if a phy receives a MUX primitive that does not match the MUX primitive expected in that position (i.e., it receives MUX (LOGICAL LINK 1) on logical link 0 or receives MUX (LOGICAL LINK 0) on logical link 1), it shall perform a link reset sequence.

7.9 Identification and hard reset sequence

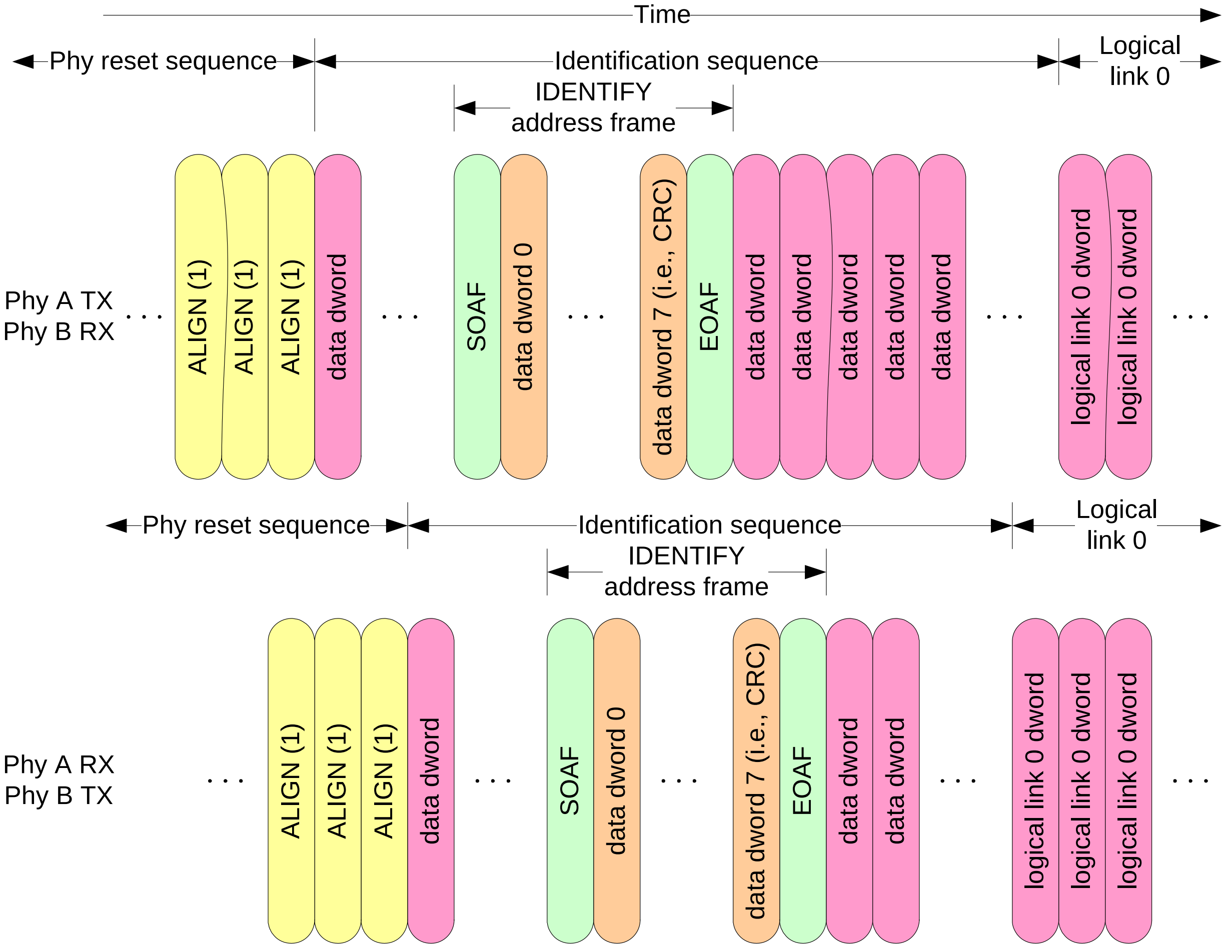
7.9.1 Identification and hard reset sequence overview

After the phy reset sequence (see 6.7) has been completed indicating the physical link is using SAS rather than SATA, each logical phy ~~transmits~~ either:

a) performs an identification sequence and transmits an IDENTIFY address frame (see 7.8.2); or
b) performs a hard reset sequence and transmits a HARD_RESET primitive sequence (see 7.2.5.8).

If multiplexing is enabled (see table 14 in x.y), the identification sequence and hard reset sequence begin with a multiplexing sequence (see 7.xx).

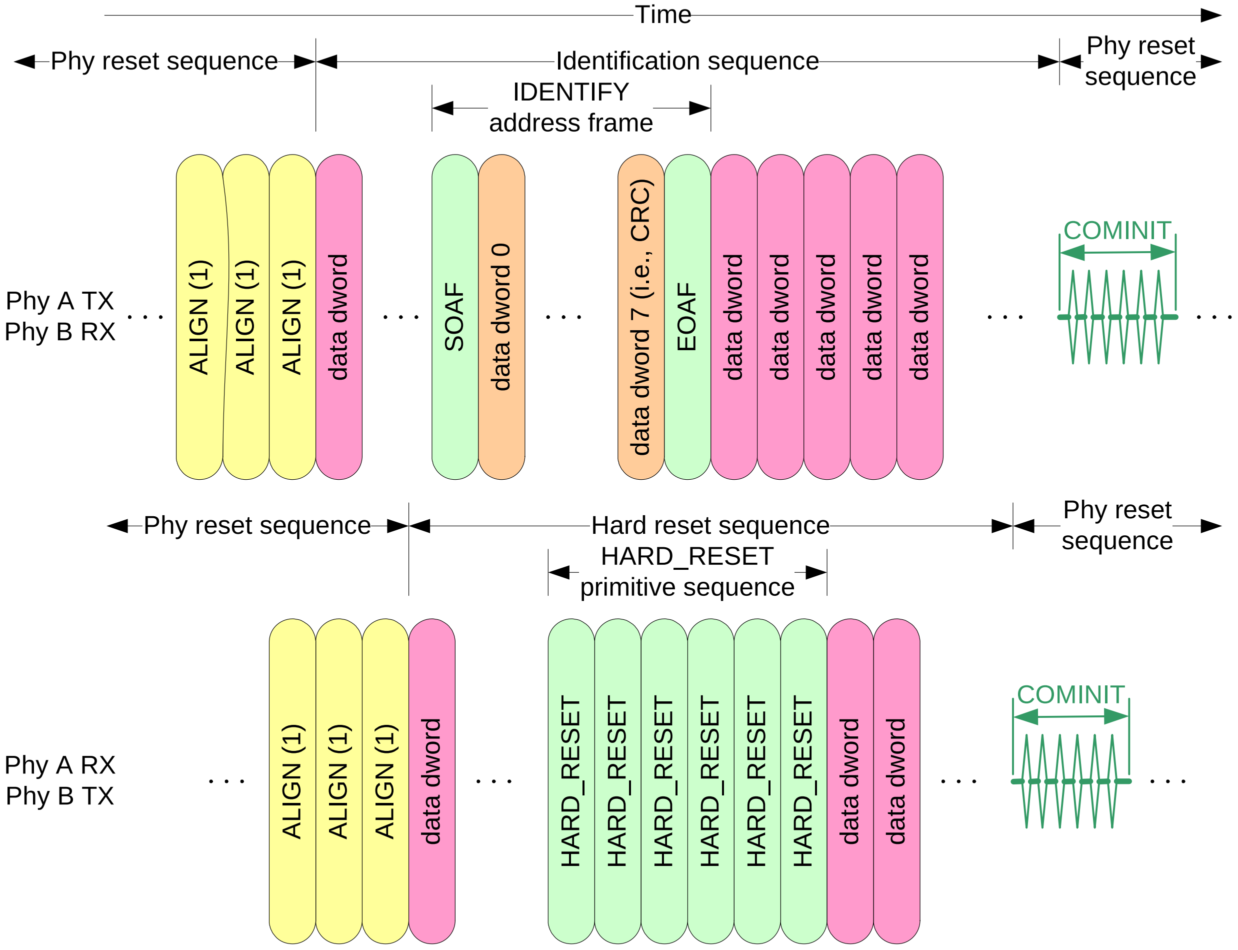
Figure 25 shows two phys with multiplexing disabled performing the identification sequence.



NOTE: Phys transmit deletable primitives for clock skew management after the phy reset sequence.

Figure 25 — Identification sequence

Figure 26 shows phy A performing the identification sequence and phy B performing the hard reset sequence. Multiplexing is disabled in this example.



NOTE: Phys transmit deletable primitives for clock skew management after the phy reset sequence.

Figure 26 — Hard reset sequence

Each logical phy receives an IDENTIFY address frame or a HARD_RESET primitive sequence from the logical phy to which it is attached. The combination of a phy reset sequence, an optional hard reset sequence followed by another phy reset sequence, and an identification sequence is called a link reset sequence (see 4.4.1).

If a phy receives a valid IDENTIFY address frame within 1 ms of phy reset sequence completion, the SAS address in the outgoing IDENTIFY address frame and the SAS address in the incoming IDENTIFY address frame determine the port to which a phy belongs (see 4.1.3). The phy ignores subsequent IDENTIFY address frames and HARD_RESET primitives until another phy reset sequence occurs.

If a phy receives a HARD_RESET primitive sequence within 1 ms of phy reset sequence completion, it shall be considered a reset event and cause a hard reset (see 4.4.2) of the port containing that phy.

If a phy does not receive a HARD_RESET primitive sequence or a valid IDENTIFY address frame within 1 ms of phy reset sequence completion, it shall restart the phy reset sequence.

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 4.7).

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 (see 7.8.3).

7.9.3 Expander device rules

After completing the ~~identification~~link reset sequence on a phy and completing internal initialization, the ECM within an expander device shall be capable of routing connection requests through that phy. The expander device may return OPEN_REJECT (NO DESTINATION) until it is ready to process connection requests.

After a link reset sequence, or after receiving a Broadcast (Change), the management application client behind an SMP initiator port in a self-configuring expander device shall follow the SAS initiator device rules (see 7.9.2) to perform a discover process (see 4.7).

The ECM of an externally configurable expander device is dependent on the completion of the discover process (see 4.7) for routing connection requests using the table routing method.

7.12 Connections

7.12.5 Aborting a connection request

...

When a phy ~~sourcing~~transmitting 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~~ a connection response does not occur (i.e., the expander device shall not forward dwords from the destination any more). Figure 144 shows an example of BREAK usage.

...

7.13 Rate matching

Each successful connection request contains the connection rate (see 4.1.10) of the pathway.

Each phy in the pathway shall insert ~~ALIGNs and/or NOTIFYs~~deletable primitives between dwords if its ~~physical~~logical link rate is faster than the connection rate as described in table 87.

Table 87 — Rate matching ~~ALIGN and/or NOTIFY~~ deletable primitive insertion requirements

~~Physical~~ Logical link rate	**Connection rate**	**Requirement**
1,5 Gbps	1,5 Gbps	None
3~~,0~~ Gbps	1,5 Gbps	One ~~ALIGN or NOTIFY~~deletable primitive within every 2 dwords that are not clock skew management ~~ALIGNs or NOTIFYs~~deletable primitives (i.e., every overlapping window of 2 dwords)(e.g., a repeating pattern of a~~n ALIGN or NOTIFY~~ deletable primitive followed by a dword or a repeating pattern of a dword followed by a~~n ALIGN or NOTIFY~~ deletable primitive)
	3~~,0~~ Gbps	None
6 Gbps	1,5 Gbps	Three deletable primitives within every 4 dwords that are not clock skew management deletable primitives (i.e., 3 in every overlapping window of 4 dwords)
	3 Gbps	One deletable primitive within every 2 dwords that are not clock skew management deletable primitives (i.e., every overlapping window of 2 dwords)(e.g., a repeating pattern of a deletable primitive followed by a dword or a repeating pattern of a dword followed by a deletable primitive)
	6 Gbps	None

~~ALIGNs and NOTIFYs~~Deletable primitives inserted for rate matching are in addition to ~~ALIGNs and NOTIFYs~~deletable primitives inserted for clock skew management (see 7.3) and STP initiator phy throttling (see 7.17.2). See Annex H for a summary of their combined requirements.

Figure 27 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 them.

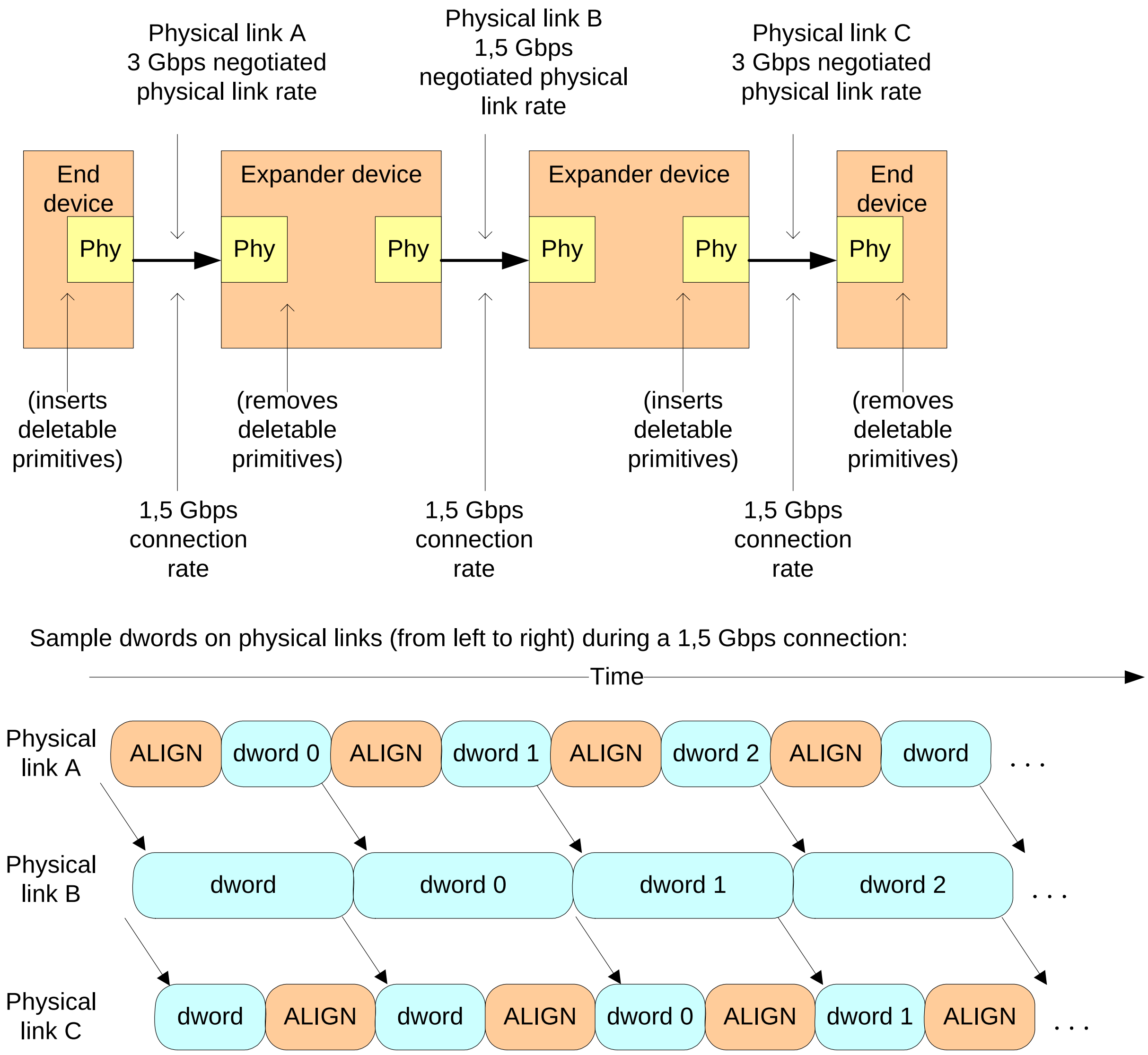


Figure 27 — Rate matching example [changed]

A phy shall start rate matching at the selected connection rate starting with the first dword that is not a~~n ALIGN or NOTIFY~~deletable primitive inserted for clock skew management 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~~deletable primitives at the selected connection rate while waiting for the connection response. This enables each expander device to start forwarding dwords from the source phy to the destination phy after forwarding an OPEN_ACCEPT.

A phy shall stop inserting ~~ALIGNs and/or NOTIFYs~~deletable primitives 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 expander phy attached to a SATA phy is using a physical link rate greater than the maximum connection rate supported by the pathway from an STP initiator port, a management application client should use the

SMP PHY CONTROL function (see 10.4.3.13) to set the PROGRAMMED MAXIMUM PHYSICAL LINK RATE field of the expander phy to the maximum connection rate supported by the pathway from that STP initiator port.

7.14 SL (link layer for SAS logical phys) state machines

7.14.1 SL state machines overview

The SL (link layer for SAS logical 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.

Editor's Note 10: change SAS phys to SAS logical phys in figure 145 and figure 146

7.14.4.1 SL_CC state machine overview

...

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 SL_IR state machines (see 7.9.5):

...

7.14.4.4.1 State description

...

NOTE 36 - Possible livelock scenarios can occur when a SAS logical phy transmits BREAK to abort a connection request (e.g., if its Open Timeout timer expires). SAS logical phys should respond to OPEN Address frames faster than 1 ms to reduce susceptibility to this problem.

7.14.4.6.1 State description

...

NOTE 37 - Possible livelock scenarios can occur when a SAS logical phy transmits BREAK to abort a connection request (e.g., if its Open Timeout timer expires). SAS logical phys should respond to OPEN Address frames faster than 1 ms to reduce susceptibility to this problem.

7.14.4.7.1 State description

...

NOTE 38 - Some SAS logical phys send a Transmit OPEN_REJECT (Retry) message to the SL transmitter in response to each OPEN Address Frame Received message received while in this state.

7.14.4.9.1 State description

...

NOTE 39 - Possible livelock scenarios can occur when a SAS logical phy transmits BREAK to break a connection (e.g., if its Close Timeout timer expires). SAS logical phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

7.15 XL (link layer for expander logical phys) state machine

7.15.1 XL state machine overview

The XL state machine controls the flow of dwords on the ~~physical~~logical link and establishes and maintains connections with another XL state machine as facilitated by the expander function (e.g., the ECM and ECR).

...

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 machines:

Editor’s Note 11: change expander phy to expander logical phy in figure 147, figure 148, and figure 149

7.15.2 XL transmitter and receiver

...

The XL transmitter shall ensure clock skew management requirements are met (see 7.3) during and after switching from forwarding dwords to originating dwords, including, for example:

a) when transmitting BREAK;
b) when transmitting CLOSE;
c) when transmitting an idle dword after closing a connection (i.e., after receiving BREAK or CLOSE);
d) while transmitting a SATA frame to a SAS ~~physical~~logical link, when transmitting the first SATA_HOLDA in response to detection of SATA_HOLD; and
e) while receiving dwords of a SATA frame from a SAS ~~physical~~logical link, when transmitting SATA_HOLD.

NOTE 40 - The XL transmitter may always insert ~~an ALIGN or NOTIFY~~a deletable primitive before transmitting a BREAK, CLOSE, or SATA_HOLDA to meet clock skew management requirements.

The XL transmitter shall insert ~~an ALIGN or NOTIFY~~a deletable primitive before switching from originating dwords to forwarding dwords, including, for example:

a) when transmitting OPEN_ACCEPT;
b) when transmitting the last idle dword before a connection is established (i.e., after receiving OPEN_ACCEPT);
c) while transmitting a SATA frame to a SAS ~~physical~~logical link, when transmitting the last dword from the SATA flow control buffer in response to release of SATA_HOLD;
d) while transmitting a SATA frame to a SAS ~~physical~~logical link, when transmitting the last SATA_HOLDA in response to release of SATA_HOLD (e.g., if the SATA flow control buffer is empty); and
e) while receiving dwords of a SATA frame from a SAS ~~physical~~logical link, when transmitting the last SATA_HOLD.

...

7.15.10 XL7:Connected state

7.15.10.1 State description

If:

a) an Invalid Dword Received message is received; and
b) the expander logical phy is forwarding to an expander logical phy attached to a SAS physical link,

the expander logical phy shall:

a) send an ERROR primitive with the Forward Dword request instead of the invalid dword; or
b) delete the invalid dword.

If:

a) an ERROR primitive is received with the Dword Received message or an Invalid Dword Received message is received; and
b) the expander logical phy is forwarding to an expander phy attached to a SATA phy,

the expander logical phy shall:

a) send a SATA_ERROR with the Forward Dword request instead of the invalid dword or ERROR primitive; or
b) delete the ERROR primitive or invalid dword.

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 Forward Close indication, then shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

> NOTE 41 - Possible livelock scenarios can occur when a SAS logical phy transmits BREAK to break a connection (e.g., if its Close Timeout timer expires). SAS logical phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

If a Dword Received message is received containing a valid dword except a BREAK or CLOSE primitive, this state shall send Forward Dword requests to the ECR containing that dword.

If:

a) an Invalid Dword Received message is received; and
b) the expander logical phy is forwarding to an expander logical phy attached to a SAS ~~physical~~logical link,

the expander logical phy shall:

a) send an ERROR primitive with the Forward Dword request instead of the invalid dword; or
b) delete the invalid dword.

If:

a) an ERROR primitive is received with the Dword Received message or an Invalid Dword Received message is received; and
b) the expander logical phy is forwarding to an expander phy attached to a SATA physical link,

the expander logical phy shall:

a) send a SATA_ERROR with the Forward Dword request instead of the invalid dword or ERROR primitive; or
b) delete the ERROR primitive or invalid dword.

If a CLOSE Received message is received, this state shall release path resources and send a Forward Close request to the ECR with the argument from the CLOSE Received message (see 7.15.11.2).

If a BREAK Received message is received, this state shall send a Forward Break request to the ECR (see 7.15.11.3).

This state shall repeatedly send a Phy Status (Connection) response to the ECM.

7.15.2 XL transmitter and receiver

...

NOTE 40 - The XL transmitter may always insert a~~n ALIGN or NOTIFY~~deletable primitive before transmitting a BREAK, CLOSE, or SATA_HOLDA to meet clock skew management requirements.

...

NOTE 41 - This ensures that clock skew management requirements are met, even if the forwarded dword stream does not include a~~n ALIGN or NOTIFY~~deletable primitive until the last possible dword.

...

7.16.8.6.5 SSP_TF4:Transmit_DONE state

...

NOTE 46 - Possible livelock scenarios can occur when a SAS logical phy transmits BREAK to break a connection (e.g., if its Done Timeout timer expires). SAS logical phys should respond to DONE faster than 1 ms to reducesusceptibility to this problem.

7.17 STP link layer

7.17.2 STP initiator phy throttling

On a SATA physical link, phys are required to transmit two consecutive ALIGN (0) primitives within every 256 dwords. To ensure an STP/SATA bridge is able to meet this requirement, an STP initiator phy has to reduce (i.e., throttle) the rate at which it is sourcing dwords by the same amount.

During an STP connection, an STP initiator phy shall insert two ~~ALIGNs or NOTIFYs~~deletable primitives within every 256 dwords (i.e., within every overlapping window of 256 dwords) that are not ~~ALIGNs or NOTIFYs~~deletable primitives for clock skew management or rate matching. They are not required to be inserted consecutively, because a phy in the pathway may delete one of them for clock skew management since STP initiator phy throttling ~~ALIGNs and NOTIFYs~~deletable primitives are indistinguishable from clock skew management ~~ALIGNs and NOTIFYs~~deletable primitives.

STP target phys are not required to insert extra ~~ALIGNs and/or NOTIFYs~~deletable primitives, because SATA hosts are not supported by SAS domains. STP initiator phys, the only recipients of data from STP target phys, do not require extra ~~ALIGNs or NOTIFYs~~deletable primitives.

~~ALIGNs and NOTIFYs~~Deletable primitives inserted for STP initiator phy throttling are in addition to ~~ALIGNs and NOTIFYs~~deletable primitives inserted for clock skew management (see 7.3) and rate matching (see 7.13). See Annex H for a summary of their combined requirements.

A phy shall start inserting ~~ALIGNs and NOTIFYs~~deletable primitives for STP initiator phy throttling after:

a) transmitting an OPEN_ACCEPT; or
b) sending the first SATA primitive after receiving an OPEN_ACCEPT.

A phy shall stop inserting ~~ALIGNs and NOTIFYs~~deletable primitives for STP initiator phy throttling after:

a) transmitting the first dword in a CLOSE; or
b) transmitting the first dword in a BREAK.

7.17.3 STP flow control

Each STP phy (i.e., STP initiator phy and STP target phy) and expander logical phy through which the STP connection is routed shall implement the SATA flow control protocol on each ~~physical~~logical link in the pathway. The flow control primitives are not forwarded through expander devices like other dwords.

When an STP phy 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 data dwords at a 1,5 Gbps connection rate; ~~or~~
b) 28 datadwords at a 3~~,0~~ Gbps connection rate; and
c) 36 data dwords at a 6 Gbps connection rate.

When an STP phy 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 47 - The receive buffer requirements are based on (20 + (4 × 2^n)) where n is 0 for 1,5 Gbps~~ and~~, 1 for 3,0 Gbps, and 2 for 6 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see ATA/ATAPI-7 V3). The (4 × 2^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.

...

7.17.4 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be transmitted two times, then be followed by SATA_CONT, if needed, then be followed by vendor-specific scrambled data dwords, if needed. ~~ALIGNs and NOTIFYs~~Deletable primitives may be sent inside continued primitive sequences as described in 7.2.4.1.

...

7.17.6 Opening an STP connection

...

The first dword that an STP phy sends inside an STP connection after OPEN_ACCEPT that is not a~~n ALIGN or NOTIFY~~ deletable primitive shall be an STP primitive (e.g., SATA_SYNC).

Changes to chapter 8 (port layer)

None so far. It may need to be made clear that the port layer talks to logical phys, not physical phys.

Changes to chapter 10 (application layer)

Define the SMP functions to enable multiplexing and discover if it is supported/being used.

10.2.9.1 Protocol-Specific diagnostic page

...

The PHY TEST PATTERN PHYSICAL LINK RATE field specifies the physical link rate at which the phy test pattern shall be transmitted and is defined in table 88. If the physical link rate specified by the PHY TEST PATTERN PHYSICAL LINK RATE field is less than the hardware minimum physical link rate or greater than the hardware maximum physical link rate, then the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 88 — PHY TEST PATTERN PHYSICAL LINK RATE field

Code	Description
0h - 7h	Reserved
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
~~Ah~~Bh - Fh	Reserved for future physical link rates

...

10.4 Management application layer

10.4.3 SMP functions

10.4.3.2 SMP function response frame format

...

The FUNCTION RESULT field is defined in table 89.

Table 89 — FUNCTION RESULT field (part 1 of 3)

Code	Name	SMP function(s)	Description
00h	SMP FUNCTION ACCEPTED	All	The management device server supports the SMP function. The ADDITIONAL RESPONSE BYTES field contains the requested information.
01h	UNKNOWN SMP FUNCTION	Unknown	The management device server 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 management device server 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 management device server supports the SMP function, but the SMP 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.

Table 89 — FUNCTION RESULT field (part 2 of 3)

Code	Name	SMP function(s)	Description
04h	INVALID EXPANDER CHANGE COUNT	CONFIGURE GENERAL, CONFIGURE ROUTE INFORMATION, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT INFORMATION	The management device server supports the SMP function, but the EXPECTED EXPANDER CHANGE COUNT field does not match the current expander change count. The ADDITIONAL RESPONSE BYTES may be present but shall be ignored.
10h	PHY DOES NOT EXIST	DISCOVER, DISCOVER LIST, REPORT PHY ERROR LOG, REPORT PHY SATA, REPORT ROUTE INFORMATION, REPORT PHY EVENT INFORMATION, CONFIGURE ROUTE INFORMATION, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT INFORMATION	The phy specified by the PHY IDENTIFIER field or the STARTING PHY IDENTIFIER field in the SMP request frame does not exist (e.g., the value is not within the range of zero to the value of the NUMBER OF PHYS field reported in the SMP REPORT GENERAL response). 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 in the SMP request frame does not have the table routing attribute (see 4.6.7.1), or the expander route index specified by the EXPANDER ROUTE INDEX field does not exist (i.e., the value is not in the range of 0000h to the value of the EXPANDER ROUTE INDEXES field in the SMP REPORT GENERAL response). The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
12h	PHY DOES NOT SUPPORT SATA	REPORT PHY SATA and PHY CONTROL (TRANSMIT SATA PORT SELECTION SIGNAL)	The phy specified by the PHY IDENTIFIER field in the SMP request frame 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 in the SMP request frame is unknown. The SMP function had no affect. The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
14h	UNKNOWN PHY TEST FUNCTION	PHY TEST FUNCTION	The operation specified by the PHY TEST FUNCTION field in the SMP request frame is unknown. The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.

Table 89 — FUNCTION RESULT field (part 3 of 3)

Code	Name	SMP function(s)	Description
15h	PHY TEST FUNCTION IN PROGRESS	PHY TEST FUNCTION	The specified phy is already performing a phy test function. The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
16h	PHY VACANT	DISCOVER, REPORT PHY ERROR LOG, REPORT PHY SATA, REPORT ROUTE INFORMATION, REPORT PHY EVENT INFORMATION, CONFIGURE ROUTE INFORMATION, PHY CONTROL, CONFIGURE PHY EVENT INFORMATION	The management device server processing the SMP request frame does not have access to the phy (e.g., because of zoning or vendor-specific reasons), although the value is within the range of zero to the value of the NUMBER OF PHYS field reported in the SMP REPORT GENERAL response. The ADDITIONAL RESPONSE BYTES field may be present but shall be ignored.
17h	PHY EVENT INFORMATION SOURCE NOT SUPPORTED	CONFIGURE PHY EVENT INFORMATION	The phy event information source specified by a PHY EVENT INFORMATION SOURCE field is not supported. The ADDITIONAL RESPONSE BYTES may be present but shall be ignored.
18h	LOGICAL LINK RATE NOT SUPPORTED	PHY CONTROL	The logical link rate specified by the REQUESTED LOGICAL LINK RATE field is not supported.
20h	SMP ZONE VIOLATION	CONFIGURE GENERAL, ENABLE DISABLE ZONING, ZONED BROADCAST, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT INFORMATION	The management device server supports the function, but zoning is enabled and the SMP initiator port does not have access to a necessary zone group according to the zone permission table (see 4.9.3.2). The ADDITIONAL RESPONSE BYTES may be present but shall be ignored.
21h	PHYSICAL PRESENCE NOT ASSERTED	ENABLE DISABLE ZONING	Physical presence was required but was not detected by the expander device when the SMP function was requested.
22h	UNKNOWN ENABLE DISABLE ZONING VALUE	ENABLE DISABLE ZONING	The ENABLE DISABLE ZONING field is set to 11b (i.e., Reserved).
All others	Reserved		

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 202 defines the request format.

Table 202 — DISCOVER request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (10h)							
2	Reserved							
8								
9	PHY IDENTIFIER							
10	Reserved							
11								
12	(MSB)	CRC						
15								(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.

Table 203 defines the response format.

Table 203 — DISCOVER response (part 1 of 3)

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (10h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH (~~17h~~18h)							
4	Reserved							
8								
9	PHY IDENTIFIER							
10	Reserved							
11								
12	Reserved	ATTACHED DEVICE TYPE			~~Reserved~~ATTACHED REASON			
13	Reserved				NEGOTIATED ~~PHYSICAL~~LOGICAL LINK RATE			
14	Reserved				ATTACHED SSP INITIATOR	ATTACHED STP INITIATOR	ATTACHED SMP INITIATOR	ATTACHED SATA HOST

Table 203 — DISCOVER response (part 2 of 3)

Byte\Bit	7	6	5	4	3	2	1	0
15	ATTACHED SATA PORT SELECTOR	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	CONNECTOR TYPE						
46	CONNECTOR ELEMENT INDEX							
47	CONNECTOR PHYSICAL LINK							
48	Reserved							
49								
50	Vendor specific							
51								
52	ATTACHED DEVICE NAME							
59								
60	Reserved	REQUESTED INSIDE ZPSDS CHANGED BY EXPANDER	INSIDE ZPSDS PERSISTENT	REQUESTED INSIDE ZPSDS	ZONE ADDRESS RESOLVED	ZONE GROUP PERSISTENT	INSIDE ZPSDS	ZONING ENABLED
61	Reserved							
62								
63	ZONE GROUP							
64	SELF-CONFIGURATION STATUS							
65	SELF-CONFIGURATION LEVELS COMPLETED							
66	Reserved							
67								

Table 203 — DISCOVER response (part 3 of 3)

Byte\Bit	7	6	5	4	3	2	1	0
68		SELF-CONFIGURATION SAS ADDRESS						
75								
bytes 76-91 proposed by 06-363r2...								
76	(bit 0)	SNW-3 INFORMATION CHANGEABLE [bits 3:0 contain the REQUESTED LOGICAL LINK RATE field]						(bit 7)
79	(bit 24)							(bit 31)
80	(bit 0)	PROGRAMMED SNW-3 INFORMATION SUPPORTED						(bit 7)
83	(bit 24)							(bit 31)
84	(bit 0)	SNW-3 INFORMATION						(bit 7)
87	(bit 24)							(bit 31)
88	(bit 0)	ATTACHED SNW-3 INFORMATION						(bit 7)
91	(bit 24)							(bit 31)
92		Reserved						
93								
94	Reserved				NEGOTIATED PHYSICAL LINK RATE			
95	Reserved							HARDWARE MUXING SUPPORTED
~~52~~ 96	(MSB)	CRC						
~~55~~ 99								(LSB)

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 RESPONSE LENGTH field shall be set to ~~17h~~18h. For compatibility with previous versions of this standard, a RESPONSE LENGTH field set to 00h indicates that there are 12 dwords before the CRC field.

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~~ device type detected during the link reset sequence and is defined in table 85.

Table 204 — ATTACHED DEVICE TYPE field

Code	Description
000b	No device attached
001b	End device
010b	Edge expander device
011b	Fanout expander device
All others	Reserved

The ATTACHED DEVICE TYPE field shall only be set to a value other than 000b after:

a) the identification sequence is complete if a SAS device or expander device is attached; or
b) the initial Register - Device to Host FIS has been received if a SATA phy is attached.

The ATTACHED REASON field indicates the value of the REASON field received in the IDENTIFY address frame (see 7.8.2) during the identification sequence.

The NEGOTIATED LOGICAL LINK RATE field and NEGOTIATED PHYSICAL LINK RATE field ~~is~~are defined in table 205 and indicates the physical or logical 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 205 — NEGOTIATED PHYSICAL LINK RATE field and NEGOTIATED LOGICAL LINK RATE field (part 1 of 2)

Code	Name	Description
0h	UNKNOWN	Phy is enabled; unknown physical or logical link rate. [a]
1h	DISABLED	Phy is disabled.
2h	PHY_ RESET_ PROBLEM	Phy is enabled; the phy obtained dword synchronization for at least one physical link rate during the SAS speed negotiation sequence (see 6.7.4.2), but the SAS 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 Port log page (see 10.2.8.1).
3h	SPINUP_ HOLD	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) may be used to release the phy. This field shall be updated to this value at SATA spinup hold time (see 6.8.7 and 6.10)(i.e., after the COMSAS Detect Timeout timer expires during the SATA OOB sequence) if SATA spinup hold is supported.
4h	PORT_ SELECTOR	Phy is enabled; detected a SATA port selector. The physical link rate has not been negotiated since the last time the phy's SP state machine entered the SP0:OOB_COMINIT state. The SATA spinup hold state has not been entered since the last time the phy's SP state machine entered the SP0:OOB_COMINIT state. The value in this field may change to 3h, 8h, ~~or~~ 9h, or Ah if attached to the active phy of the SATA port selector. Presence of a SATA port selector is indicated by the ATTACHED SATA PORT SELECTOR bit.

Table 205 — NEGOTIATED PHYSICAL LINK RATE field and NEGOTIATED LOGICAL LINK RATE field (part 2 of 2)

Code	Name	Description
8h	G1	Phy is enabled; 1,5 Gbps physical or logical link link rate. ~~This~~ The NEGOTIATED PHYSICAL LINK RATE field shall be updated to this value after the speed negotiation sequence completes if it negotiates 1,5 Gbps. The NEGOTIATED LOGICAL LINK RATE field shall be updated to this value after the speed negotiation sequence completes if it negotiates a single 1,5 Gbps logical link or negotiates two multiplexed 1,5 Gbps logical links.
9h	G2	Phy is enabled; 3~~,0~~ Gbps physical or logical link link rate. ~~This~~ The NEGOTIATED PHYSICAL LINK RATE field shall be updated to this value after the speed negotiation sequence completes if it negotiates 3 Gbps. The NEGOTIATED LOGICAL LINK RATE field shall be updated to this value after the speed negotiation sequence completes if it negotiates a single 3 Gbps logical link or negotiates two multiplexed 3 Gbps logical links.
Ah	G3	Phy is enabled; 6 Gbps physical or logical link rate. The NEGOTIATED PHYSICAL LINK RATE field shall be updated to this value after the speed negotiation sequence completes if it negotiates 6 Gbps. The NEGOTIATED LOGICAL LINK RATE field shall be updated to this value after the speed negotiation sequence completes if it negotiates a single 6 Gbps logical link.
Bh - Fh	Reserved	Phy is enabled; reserved for future physical or logical link rates.
All others	Reserved.	

[a] This code may be used by an application client in its local data structures to indicate an unknown negotiated ~~physical~~logical link rate (e.g., before the discover process has queried the phy).

NOTE 48 - In previous versions of this standard that did not define multiplexing, the NEGOTIATED LOGICAL LINK RATE field was called the NEGOTIATED PHYSICAL LINK RATE field and the NEGOTIATED PHYSICAL LINK RATE field in byte 94 did not exist.

Table 206 describes the ATTACHED SATA PORT SELECTOR bit and the ATTACHED SATA DEVICE bit.

Table 206 — ATTACHED SATA PORT SELECTOR and ATTACHED SATA DEVICE bits

ATTACHED SATA PORT SELECTOR **bit value** [a] [b]	ATTACHED SATA DEVICE **bit value** [c] [d]	**Description**
0	0	Neither a SATA port selector nor a SATA device is attached and ready on the selected phy.
0	1	The attached phy is a SATA device phy. No SATA port selector is present (i.e., the SP state machine did not detect COMWAKE in response to the initial COMINIT, but sequenced through the normal (non-SATA port selector) SATA device OOB sequence).
1	0	The attached phy is a SATA port selector host phy, and either: a) the attached phy is the inactive host phy, or b) the attached phy is the active host phy and a SATA device is either not present or not ready behind the SATA port selector (i.e., the SP state machine detected COMWAKE while waiting for COMINIT).
1	1	The attached phy is a SATA port selector's active host phy and a SATA device is present behind the SATA port selector (i.e., the SP state machine detected COMWAKE while waiting for COMINIT, timed out waiting for COMSAS, and exchanged COMWAKE with an attached SATA device).

[a] The ATTACHED SATA PORT SELECTOR bit is invalid if the NEGOTIATED ~~PHYSICAL~~LOGICAL LINK RATE field is set to UNKNOWN (i.e., 0h) or DISABLED (i.e., 1h).

[b] Whenever the ATTACHED SATA PORT SELECTOR bit changes, the phy shall generate a BROADCAST(CHANGE) notification.

[c] For the purposes of the ATTACHED SATA DEVICE bit, the SATA port selector is not considered a SATA device.

[d] The ATTACHED SATA DEVICE bit shall be updated at SATA spin-up hold time (see 6.8.7 and 6.10).

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 49 - Support for SATA hosts is outside the scope of this standard.

If a SAS phy reset sequence occurs (see 6.7.4)(i.e., one or more of the ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, the ATTACHED SMP INITIATOR PORT bit, the ATTACHED SSP TARGET PORT bit, the ATTACHED STP TARGET PORT bit, and/or the ATTACHED SMP TARGET PORT bit is set to one), then the ATTACHED SATA PORT SELECTOR bit, the ATTACHED SATA DEVICE bit, and the ATTACHED SATA HOST bit shall each be set to zero.

The ATTACHED SSP INITIATOR PORT bit indicates the value of the SSP INITIATOR PORT field received in the IDENTIFY address frame (see 7.8.2) during the identification sequence.

The ATTACHED STP INITIATOR PORT bit indicates the value of the STP INITIATOR PORT field received in the IDENTIFY address frame (see 7.8.2) during the identification sequence.

The ATTACHED SMP INITIATOR PORT bit indicates the value of the SMP INITIATOR PORT field received in the IDENTIFY address frame (see 7.8.2) during the identification sequence.

The ATTACHED SSP TARGET PORT bit indicates the value of the SSP TARGET PORT field received in the IDENTIFY address frame (see 7.8.2) during the identification sequence.

The ATTACHED STP TARGET PORT bit indicates the value of the STP TARGET PORT field received in the IDENTIFY address frame (see 7.8.2) during the identification sequence.

The ATTACHED SMP TARGET PORT bit indicates the value of the SMP TARGET PORT field received in the IDENTIFY address frame (see 7.8.2) during the identification 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, and ATTACHED SMP TARGET PORT bit shall be updated at the end of the identification sequence.

If a SATA phy reset sequence occurs (see 6.7.3)(i.e., the ATTACHED SATA PORT SELECTOR bit is set to one, the ATTACHED SATA DEVICE bit is set to one, or the ATTACHED SATA HOST bit is set to one), then 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, and ATTACHED SMP TARGET PORT bit shall each be set to zero.

The SAS ADDRESS field contains the value of the SAS ADDRESS field transmitted in the IDENTIFY address frame during the identification sequence. If the phy is an expander phy, the SAS ADDRESS field contains the SAS address of the expander device (see 4.2.4). If the phy is a SAS phy, the SAS ADDRESS field contains the SAS address of the SAS port (see 4.2.6).

The ATTACHED SAS ADDRESS field contains the value of the SAS ADDRESS field received in the IDENTIFY address frame during the identification sequence. If the attached port is an expander port, the ATTACHED SAS ADDRESS field contains the SAS address of the attached expander device (see 4.2.4). If the attached port is a SAS port, the ATTACHED SAS ADDRESS field contains SAS address of the attached SAS port (see 4.2.6). If the attached port is a SATA device port, the ATTACHED SAS ADDRESS field contains the SAS address of the STP/SATA bridge (see 4.6.2).

The ATTACHED SAS ADDRESS field shall be updated:

a) after the identification sequence completes, if a SAS phy or expander phy is attached; or
b) after the COMSAS Detect Timeout timer expires (see 6.8.3.9), if a SATA phy is attached.

An STP initiator port should not make a connection request to the attached SAS address until the ATTACHED DEVICE TYPE field is set to a value other than 000b.

The ATTACHED PHY IDENTIFIER field contains a phy identifier for the attached phy:

a) If the attached phy is a SAS phy or an expander phy, the ATTACHED PHY IDENTIFIER field contains the value of the PHY IDENTIFIER field received in the IDENTIFY address frame during the identification sequence:
    A) If the attached phy is a SAS phy, the ATTACHED PHY IDENTIFIER field contains the phy identifier of the attached SAS phy in the attached SAS device;
    B) If the attached phy is an expander phy, the ATTACHED PHY IDENTIFIER field contains the phy identifier (see 4.2.7) of the attached expander phy in the attached expander device; and
b) If the attached phy is a SATA device phy, the ATTACHED PHY IDENTIFIER field contains 00h;
c) If the attached phy is a SATA port selector phy and the expander device is able to determine the port of the SATA port selector to which it is attached, the ATTACHED PHY IDENTIFIER field contains 00h or 01h; and
d) If the attached phy is a SATA port selector phy and the expander device is not able to determine the port of the SATA port selector to which it is attached, the ATTACHED PHY IDENTIFIER field contains 00h.

The ATTACHED PHY IDENTIFIER field shall be updated:

a) after the identification sequence completes, if a SAS phy or expander phy is attached; or
b) after the COMSAS Detect Timeout timer expires (see 6.8.3.9), if a SATA phy is attached.

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field indicates the minimum physical link rate set by the PHY CONTROL function (see). The values are defined in table 207. 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 208.

The PROGRAMMED MAXIMUM PHYSICAL LINK RATE field indicates the maximum physical link rate set by the PHY CONTROL function (see). The values are defined in table 207. The default value shall be the value of the HARDWARE MAXIMUM PHYSICAL LINK RATE field.

Table 207 — PROGRAMMED MINIMUM PHYSICAL LINK RATE and PROGRAMMED MAXIMUM PHYSICAL LINK rate fields

Code	Description
0h	Not programmable
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
Bh - Fh	Reserved for future physical link rates
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 208.

Table 208 — HARDWARE MINIMUM PHYSICAL LINK RATE and HARDWARE MAXIMUM PHYSICAL LINK RATE fields

Code	Description
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
Bh - Fh	Reserved for future physical link rates
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 any reason described in 7.11 originating from the expander phy other than forwarding a BROADCAST (CHANGE).

The expander device is not required to increment the PHY CHANGE COUNT field again unless a DISCOVER response is transmitted. This field shall not be incremented when forwarding a BROADCAST (CHANGE) from another expander device.The PHY CHANGE COUNT field shall wrap to zero after the maximum value (i.e., FFh) has been reached.

NOTE 50 - 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.

A VIRTUAL PHY bit set to one indicates the phy is part of an internal port and the attached device is contained within the expander device. A 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 7.12.4.5).

NOTE 51 - 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).

The ROUTING ATTRIBUTE field indicates the routing attribute supported by the phy (see 4.6.7.1) and is defined in table 209.

Table 209 — ROUTING ATTRIBUTE field

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 CONNECTOR TYPE field indicates the type of connector used to access the phy, as reported by the enclosure services process for the enclosure (see the SAS Connector element in SES-2). A CONNECTOR TYPE field set to 00h indicates no connector information is available and that the CONNECTOR ELEMENT INDEX field and the CONNECTOR PHYSICAL LINK fields are invalid and shall be ignored.

The CONNECTOR ELEMENT INDEX indicates the element index of the SAS Connector element representing the connector used to access the phy, as reported by the enclosure services process for the enclosure (see the SAS Connector element in SES-2).

The CONNECTOR PHYSICAL LINK field indicates the physical link in the connector used to access the phy, as reported by the enclosure services process for the enclosure (see the SAS Connector element in SES-2).

A HARDWARE MUXING SUPPORTED bit set to one indicates the phy supports multiplexing. A HARDWARE MUXING SUPPORTED bit set to zero indicates the phy supports multiplexing. This value is not adjusted based on the negotiated physical link rate.

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 be implemented by any SMP target port.

Table 227 defines the request format.

Table 227 — PHY CONTROL request

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (40h)							
1	FUNCTION (91h)							
2	Reserved							
3	REQUEST LENGTH (0Ah)							
4	Reserved							
8								
9	PHY IDENTIFIER							
10	PHY OPERATION							
11	Reserved							UPDATE PARTIAL PATHWAY TIMEOUT VALUE
12	Reserved							
31								
32	PROGRAMMED MINIMUM PHYSICAL LINK RATE				Reserved			
33	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				Reserved			
34	Reserved							
35								
36	Reserved				PARTIAL PATHWAY TIMEOUT VALUE			
37	Reserved							
39								
bytes 40-43 proposed by 06-363r2 and 06-324r8...								
40	(bit 0)	PROGRAMMED SNW-3 INFORMATION SUPPORTED [bits 3:0 contain the REQUESTED LOGICAL LINK RATE field]						(bit 7)
43	(bit 24)							(bit 31)
40	(MSB)	CRC						
43								(LSB)

The SMP FRAME TYPE field shall be set to 40h.

The FUNCTION field shall be set to 91h.

The REQUEST LENGTH field shall be set to 0Ah. For compatibility with previous versions of this standard, a REQUEST LENGTH field set to 00h specifies that there are 9 dwords before the CRC field.

The PHY IDENTIFIER field specifies the phy (see 4.2.7) to which the PHY CONTROL request applies.

Table 228 defines the PHY OPERATION field.

Table 228 — PHY OPERATION field (part 1 of 2)

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. See 7.11 for BROADCAST (CHANGE) requirements related to this phy operation in an expander device. Any affiliation (see 7.17.5) shall continue to be present. The phy shall bypass the SATA spinup hold state, if implemented (see 6.8.3.9). 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 a SAS phy or an expander 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. See 7.11 for BROADCAST (CHANGE) requirements related to this phy operation in an expander device. If the specified phy is a virtual phy, perform an internal reset and enable the specified phy. Any affiliation (see 7.17.5) 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. See 7.11 for BROADCAST (CHANGE) requirements related to this phy operation in an expander device.
04h	Reserved	
05h	CLEAR ERROR LOG	Clear the error log counters (see 10.4.3.6) for the specified phy.

Table 228 — PHY OPERATION field (part 2 of 2)

Code	Operation	Description
06h	CLEAR AFFILIATION	Clear an affiliation (see 7.17.5) 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.
07h	TRANSMIT SATA PORT SELECTION SIGNAL	This function shall only be supported by phys in an expander device. If the expander phy incorporates an STP/SATA bridge and supports SATA port selectors, the phy shall transmit the SATA port selection signal (see 6.6) which causes the SATA port selector to select the attached phy as the active host phy and make its other host phy inactive. See 7.11 for BROADCAST (CHANGE) requirements related to this phy operation in an expander device. Any affiliation (see 7.17.5) shall be cleared. If the expander phy does not support SATA port selectors, then the SMP target port shall return a function result of PHY DOES NOT SUPPORT SATA. If the expander phy supports SATA port selectors but is attached to a SAS phy or an expander phy, the SMP target port shall return a function result of SMP FUNCTION FAILED.
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 229 defines the values for this field. If this field is changed along with a phy operation of LINK RESET or HARD RESET, that phy operation shall utilize the new value for this field.

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 229 defines the values for this field. If this field is changed along with a phy operation of LINK RESET or HARD RESET, that phy operation shall utilize the new value for this field.

Table 229 — PROGRAMMED MINIMUM PHYSICAL LINK RATE and PROGRAMMED MAXIMUM PHYSICAL LINK RATE fields

Code	Description
0h	Do not change current value
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
Bh - Fh	Reserved for future physical link rates
All others	Reserved

If the PROGRAMMED MINIMUM PHYSICAL LINK RATE field or the PROGRAMMED MAXIMUM PHYSICAL LINK RATE field is set to an unsupported or reserved value, or the PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field are set to an invalid combination of values (e.g., the minimum is greater than the maximum), the SMP target port shall not change either of their values and may return a function result of SMP FUNCTION FAILED in the response frame. If it returns a function result of SMP FUNCTION FAILED, it shall not perform the requested phy operation.

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 7.12.4.6). 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 PROGRAMMED SNW-3 INFORMATION SUPPORTED field specifies the outgoing SNW-3 information the phy shall use in every subsequent link reset sequence containing an SNW-3. If the phy does not support the value (e.g., a non-changeable bit is set to one) or the value does not agree with the PROGRAMMED MINIMUM PHYSICAL LINK RATE field and the PROGRAMMED MAXIMUM PHYSICAL LINK RATE field, the management device server shall not change the current SNW-3 information, shall return a function result of SMP FUNCTION FAILED in the response frame, and shall not perform the requested phy operation. This value is reported in the DISCOVER response (see 10.4.3.5).

The CRC field is defined in 10.4.3.1.

Table 230 defines the response format.

Table 230 — 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.

10.4.3.12 PHY TEST FUNCTION function

...

The PHY TEST PATTERN PHYSICAL LINK RATE field specifies the physical link rate at which the phy test function, if any, shall be performed. Table 231 defines the values for this field.

Table 231 — PHY TEST PATTERN PHYSICAL LINK RATE field

Code	Description
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
Bh - Fh	Reserved for future physical link rates
All others	Reserved

...

Changes to the annexes

Add in 6 Gbps support and define the MUX primitive encodings.

A.2 Compliant jitter tolerance pattern (CJTPAT)

...

Other SSP frame header information (see 9.2.1) may be included in the CJTPAT. ~~ALIGN and/or NOTIFY~~Deletable primitives may be included in the transmission of the CJTPAT, but the number of ~~ALIGN and/or NOTIFY~~deletable primitives transmitted should be as small as possible so that the percentage of the transfer that is the JTPAT is as high as possible.

...

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 and/or NOTIFYswithin the frame should be avoided because of the possible disruption of the pattern on the physical link.~~

...

Annex H ~~ALIGN and/or NOTIFY~~Deletable primitive insertion rate summary

Table 0.1 shows all the possible combinations of ~~ALIGN and/or NOTIFY~~deletable primitive insertion rates for clock skew management (see 7.3), rate matching (see 7.13), and STP initiator phy throttling (see 7.17.2).

Table 0.1 — ~~ALIGN and/or NOTIFY~~Deletable primitive insertion rate examples

Physical link rate	Connection rate	Type of dword stream	~~ALIGN and/or NOTIFY~~Deletable primitive insertion rate (per specified number of dwords)
6 Gbps	6 Gbps	all but to STP target	4 per 8 196 (clock skew management)
		to STP target	4 per 8 196 (clock skew management) + 2 per 256 (STP initiator phy throttling)
	3 Gbps	all but to STP target	4 per 8 196 (clock skew management) + 1 per 2 (rate matching)
		to STP target	4 per 8 196 (clock skew management) + 1 per 2 (rate matching) + 2 per 256 (STP initiator phy throttling)
	1,5 Gbps	all but to STP target	4 per 8 196 (clock skew management) + 3 per 4 (rate matching)
		to STP target	4 per 8 196 (clock skew management) + 3 per 4 (rate matching) + 2 per 256 (STP initiator phy throttling)
3~~,0~~ Gbps	3~~,0~~ Gbps	all but to STP target	2 per 4 096 (clock skew management)
		to STP target	2 per 4 096 (clock skew management) + 2 per 256 (STP initiator phy throttling)
	1,5 Gbps	all but to STP target	2 per 4 096 (clock skew management) + 1 per 2 (rate matching)
		to STP target	2 per 4 096 (clock skew management) + 1 per 2 (rate matching) + 2 per 256 (STP initiator phy throttling)
1,5 Gbps	1,5 Gbps	all but to STP target	1 per 2 048 (clock skew management)
		to STP target	1 per 2 048 (clock skew management) + 2 per 256 (STP initiator phy throttling)

Annex J Primitive encoding

The MUX (LOGICAL LINK 0) and MUX (LOGICAL LINK 1) primitive encodings were selected to avoid having as many duplicate characters as possible (to reduce EMI, since these primitives are transmitted back-to-back). There are two overlaps in this set (D16.7 and D24.0), which is the best available in the unused values.

...

Table 0.2 — Primitives with Hamming distance of 8 (part 1 of 4)

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

Table 0.2 — Primitives with Hamming distance of 8 (part 2 of 4)

1st	2nd	3rd	4th	Assignment
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	MUX (LOGICAL LINK 0)
K28.5	D02.0	D24.0	D07.3	BREAK
K28.5	D02.0	D29.7	D16.7	MUX (LOGICAL LINK 1)
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 (SES)
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	
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	

Table 0.2 — Primitives with Hamming distance of 8 (part 3 of 4)

1^{st}	2^{nd}	3^{rd}	4^{th}	Assignment
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)
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)

Table 0.2 — Primitives with Hamming distance of 8 (part 4 of 4)

1st	2nd	3rd	4th	Assignment
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 (POWER FAILURE EXPECTED)
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)