To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 4 March 2006
Subject: 05-381r2 SAS-2 Multiplexing

Revision history

Revision 0 (2 November 2005) First revision

Revision 1 (23 December 2005) Incorporated feedback from 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 SAS protocol WG. Changed references to "ALIGNs/NOTIFYs" and its many variations into "deletable primitives", which are defined as ALIGN, NOTIFY, and MUX.

Related documents

sas2r02 - Serial Attached SCSI - 2 revision 02

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.

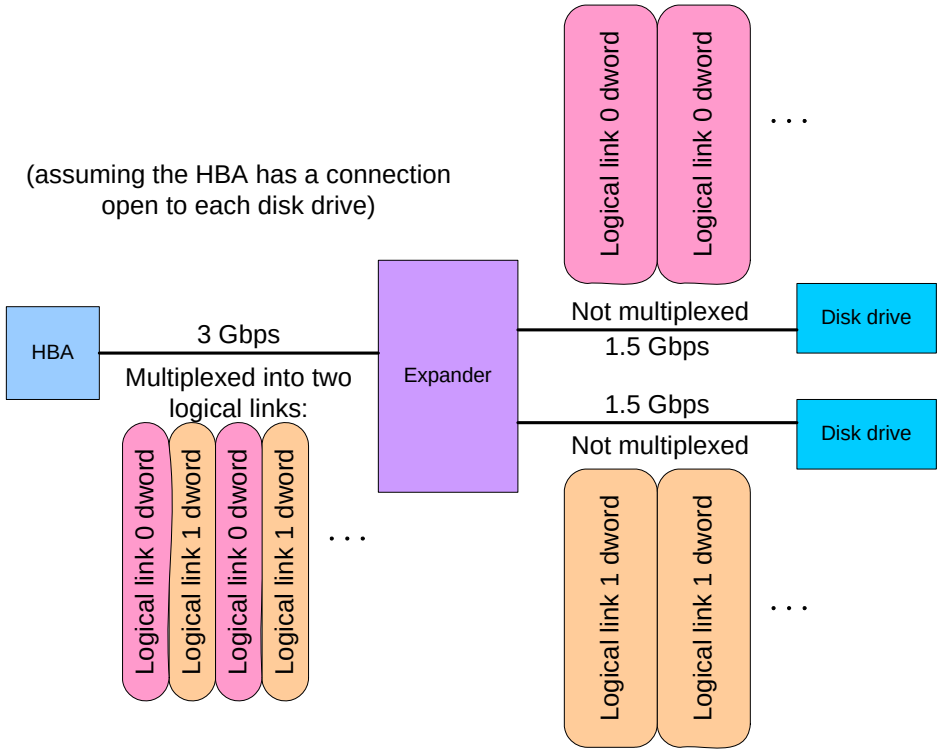


Figure 1 — Multiplexing overview

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) Multiplex 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) Four-way multiplexing (e.g. a 6 Gbps physical link into four 1.5 Gbps logical links)
   D) No nesting (e.g. no 6 Gbps physical link to one 3 Gbps and two 1.5 Gbps logical links)
d) IDENTIFY address frame modifications
   A) Indicate support for multiplexing (none, 2-way, 4-way)
   B) Indicate true bandwidth needs (for targets)
e) Negotiate to turn on multiplexing after the link reset sequence
   A) Define new MUX(0)/(1)/(2)/(3) primitives
   B) Transmit MUX and receive MUX to begin multiplexing
   C) Only transmit MUX if both IDENTIFY address frames agreed that multiplexing is supported
   D) Negotiate to the highest common level of multiplexing supported by the two phys (e.g. favor 4-way over 2-way, but 1-way wins over all)
   E) Rotate between MUX (0)/(1)/(2)/(3) to reduce EMI while negotiating
   F) MUXing happens outside elasticity buffers, so no ALIGN/NOTIFY during MUX exchange
   G) Periodically resend MUX to confirm logical link numbers (mainly for logic analyzers)
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; 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. 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 the model clause

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

3.1.5 logical link: A physical link or a multiplexed portion of a physical link. See 4.x.

3.1.6 logical phy: A phy or a multiplexed portion of a phy. See 4.x.

3.1.7 multiplexing: Dividing a physical link into multiple logical links. See 4.xx.

3.1.8 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.

3.1.9 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.

3.1.10 phy: A object in a device that is used to interface to other devices (e.g., an expander phy (see 3.1.4) or a SAS phy (see 3.1.15)). See 4.1.2.

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

3.1.12 physical phy: A phy (see 3.1.10) 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.

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

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

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

3.1.16 unit interval (UI): The normalized, dimensionless, nominal duration of a signal transmission bit (e.g., 666,6 ps at 1,5 Gbps ~~and~~, 333,3 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.

3.1.17 virtual phy: A phy (see 3.1.10) 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)

Abbreviation	Meaning
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)
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, and multiplexing state machines (see 7.9.5)
...	...

Changes to the model clause

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

4 General

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.

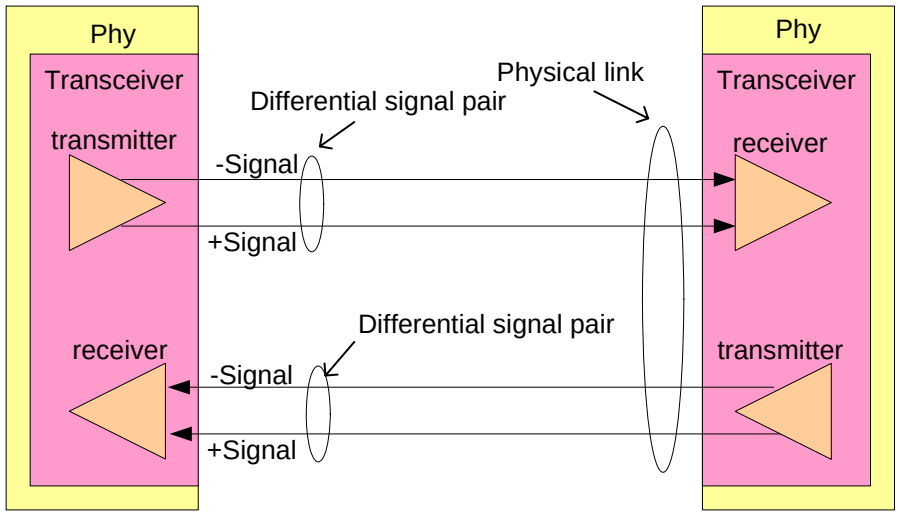


Figure 2 — Physical links and phys

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.

During the identification sequence (see 7.9), a 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 phy, the SAS address of the SAS port or expander device containing the 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 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 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).

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

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).

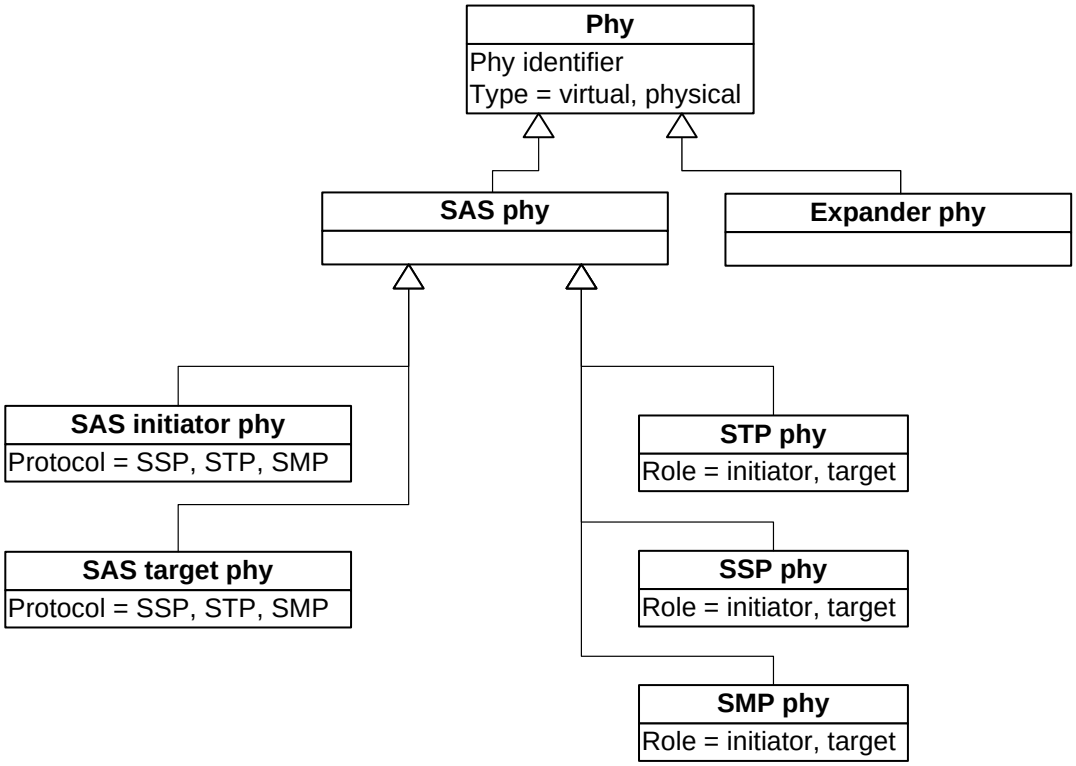


Figure 3 — Phy class diagram

Editor's Note 1: Logical phys probably need to be included in the UML model, perhaps just as another attribute or a new value in the Type attribute.

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.

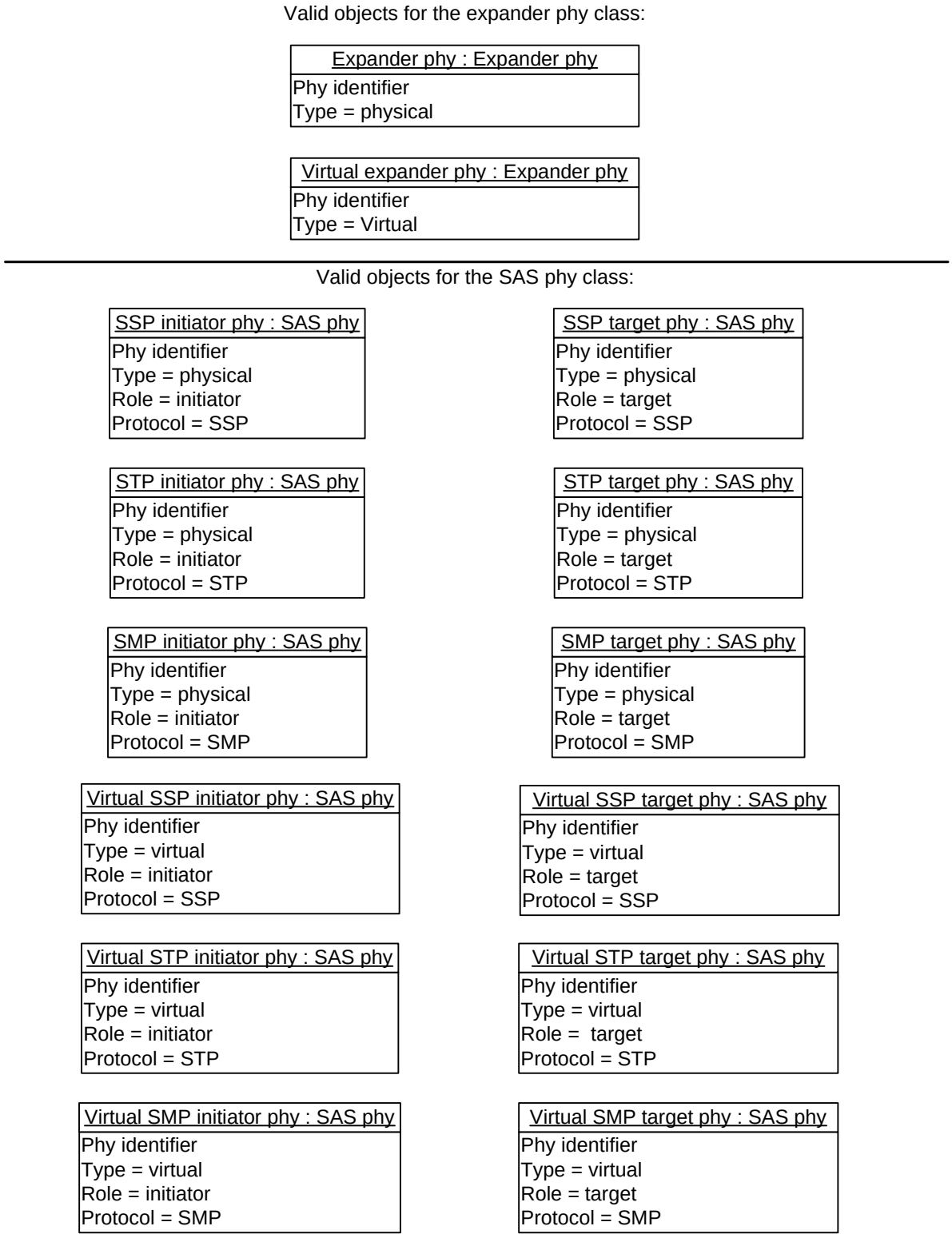


Figure 4 — Phy object diagram

4.x Logical links

A physical link with a physical link rate greater than 1,5 Gbps may be multiplexed into two or four logical links

as defined in table 1.

Table 1 — Logical links

Physical link rate	Logical links
6 Gbps	One 6 Gbps logical link
	Two 3 Gbps logical links
	Four 1,5 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

Logical links are negotiated using MUX primitives (see 7.xx).

4.3.1 State machine overview

Figure 5 shows the state machines for SAS devices, their relationships to each other and to the SAS device, SAS port, and SAS phy classes.

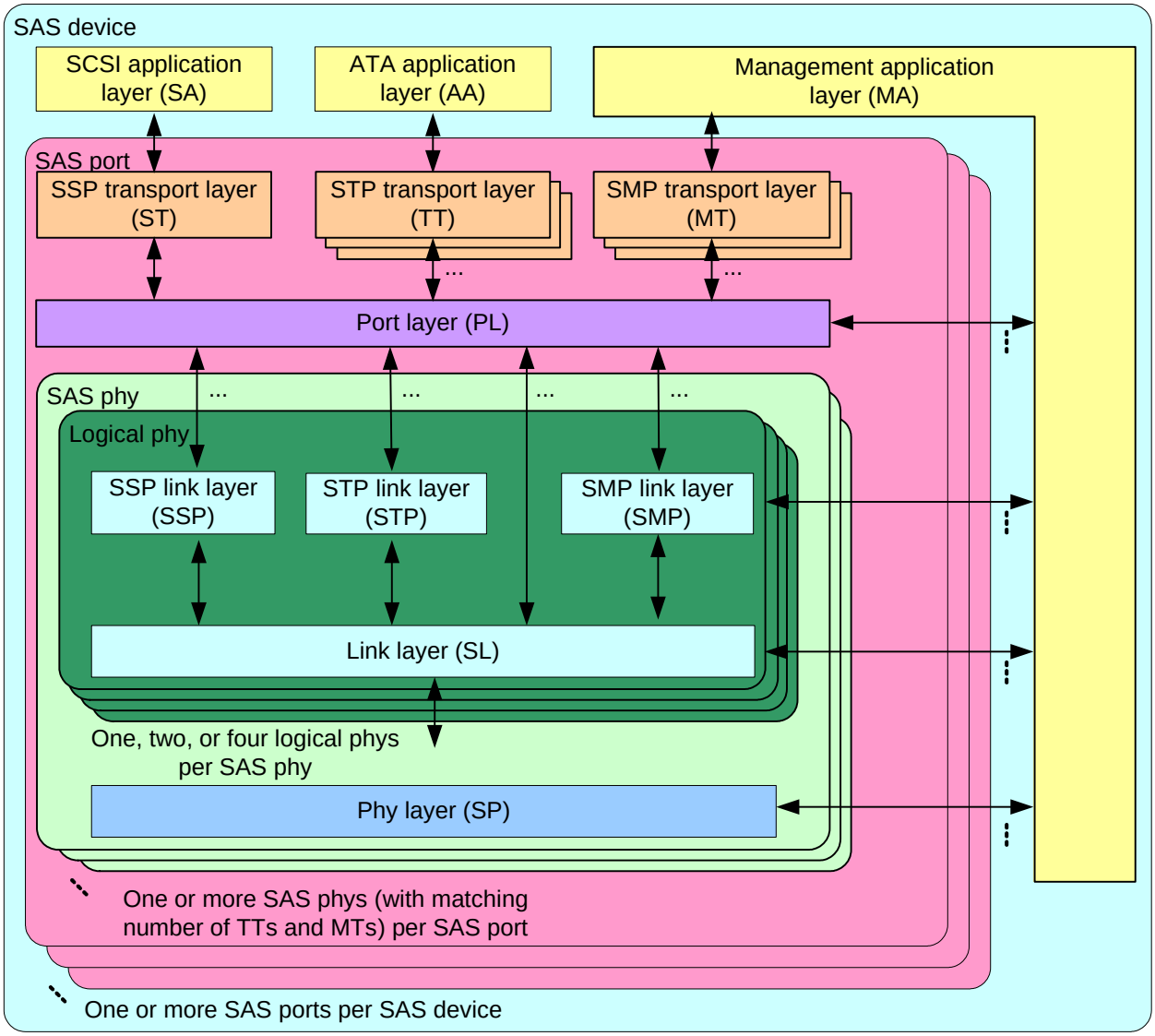


Figure 5 — State machines for SAS devices [updated to include logical phys]

Figure 6 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.

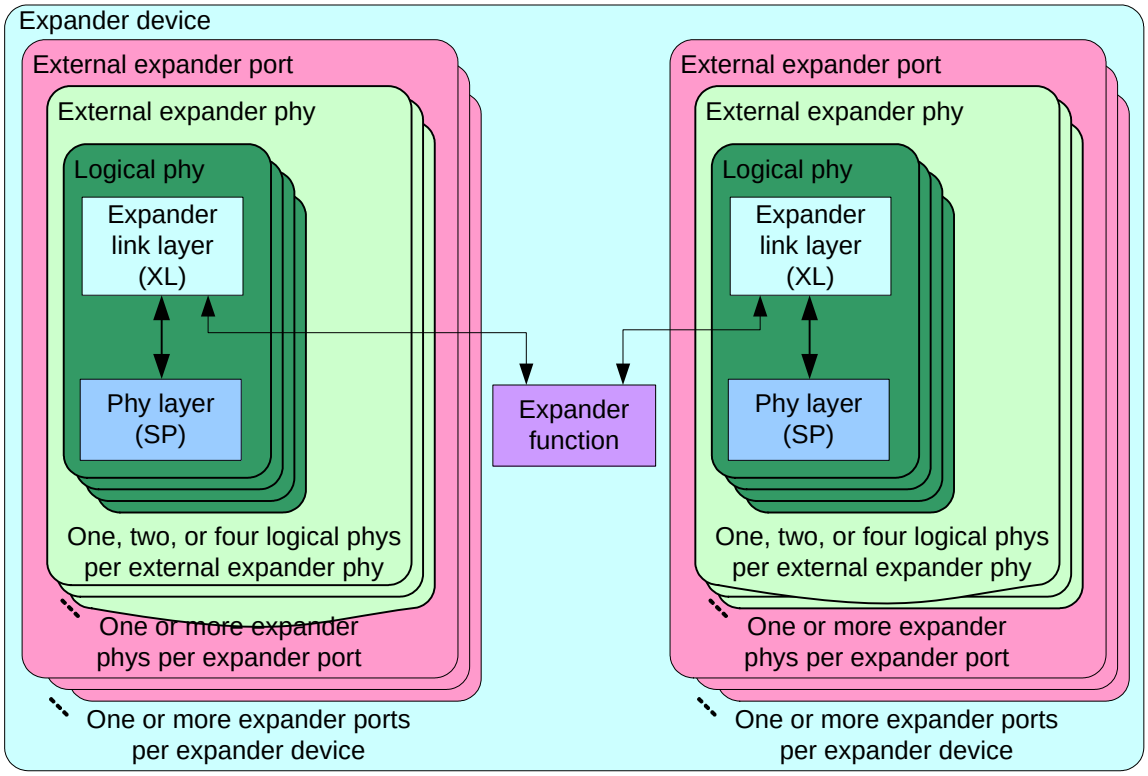


Figure 6 — State machines for expander devices [updated to include logical phys]

Annex K contains a list of messages between state machines.

4.3.2 Transmit data path

Figure 7 shows the transmit data path in a SAS phy.

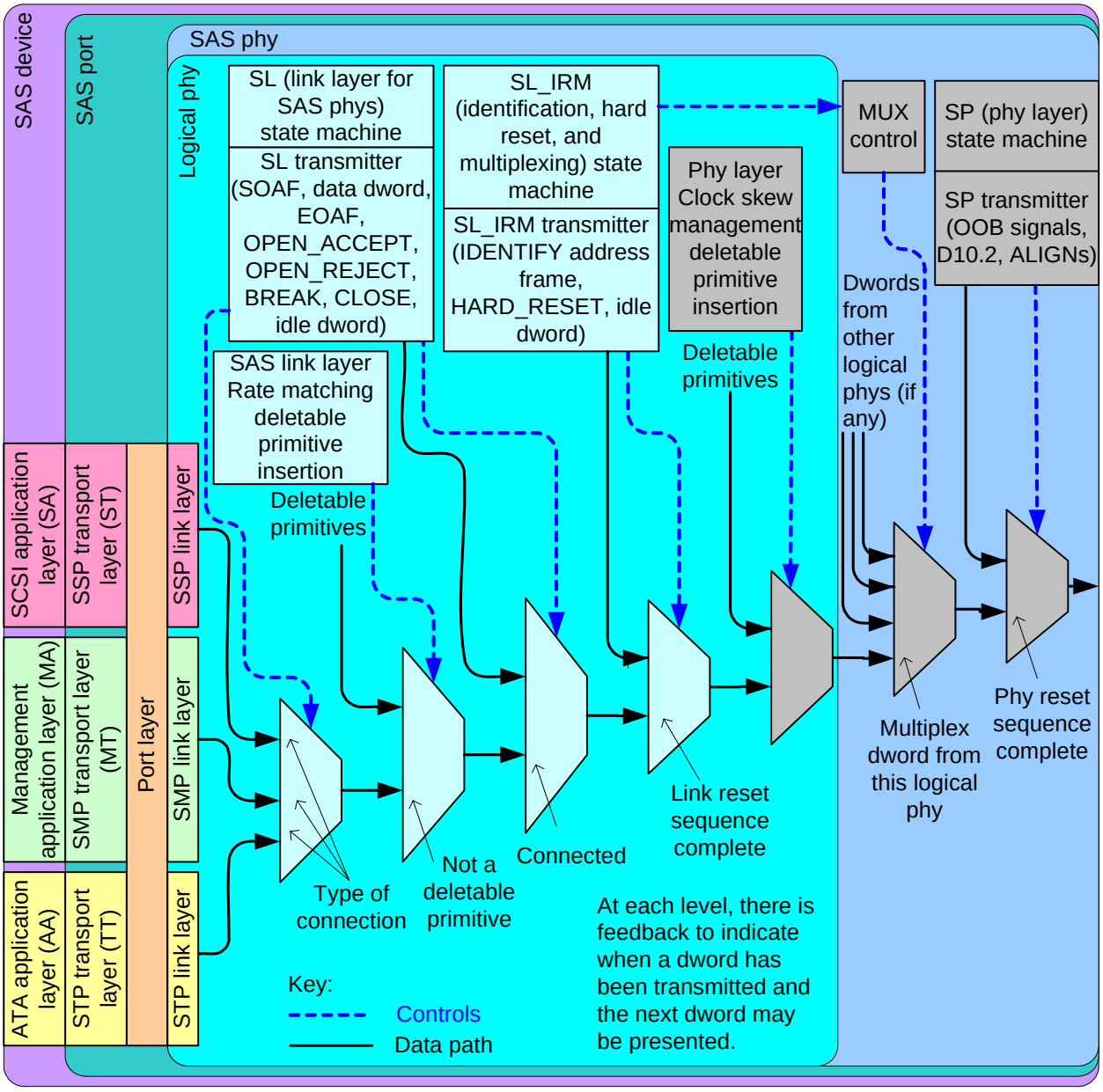


Figure 7 — Transmit data path in a SAS phy [updated to include logical phy]

...

Figure 8 shows the transmit data path in an expander phy.

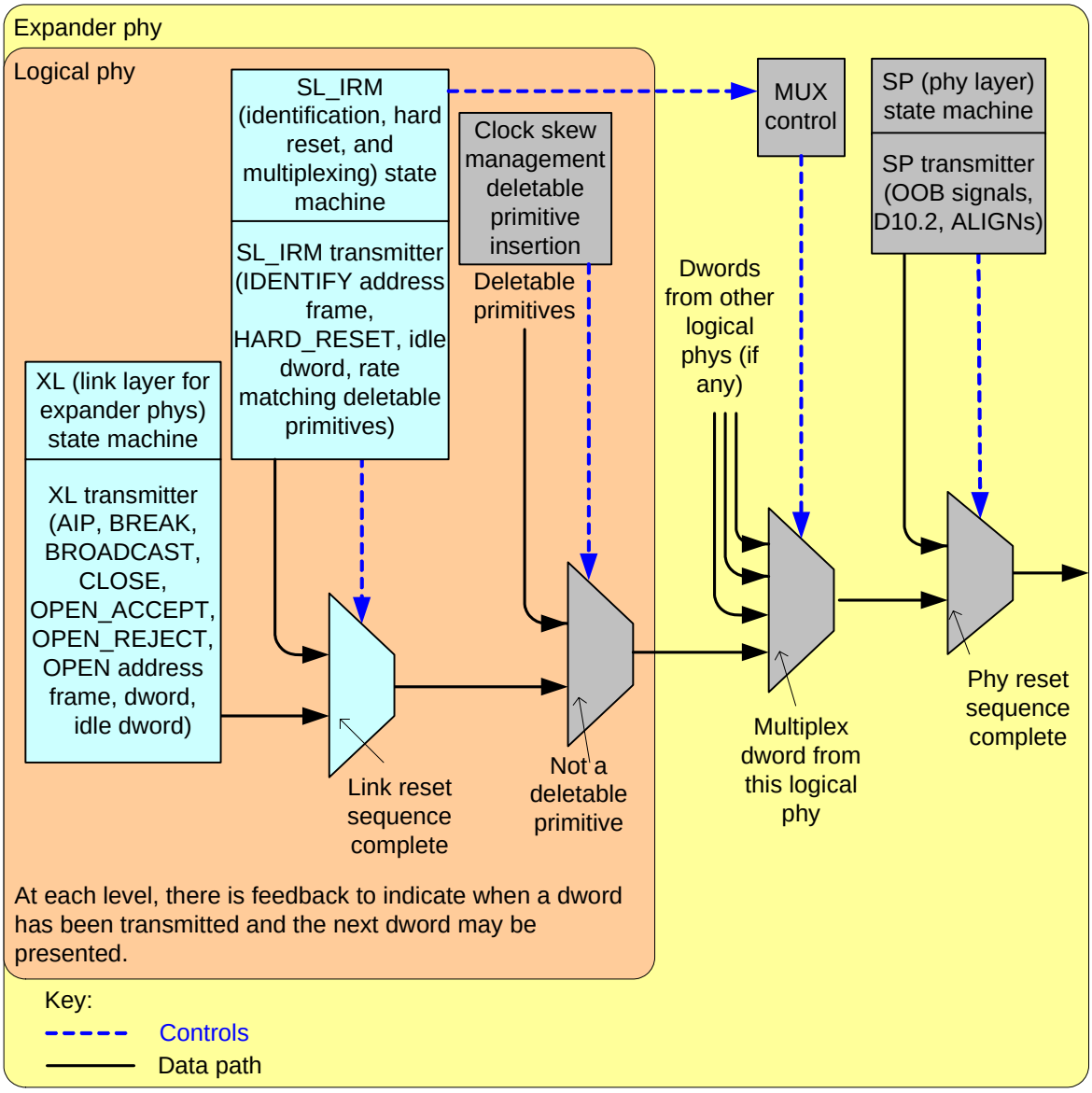


Figure 8 — Transmit data path and state machines in an expander phy [updated to include logical phy]

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.

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

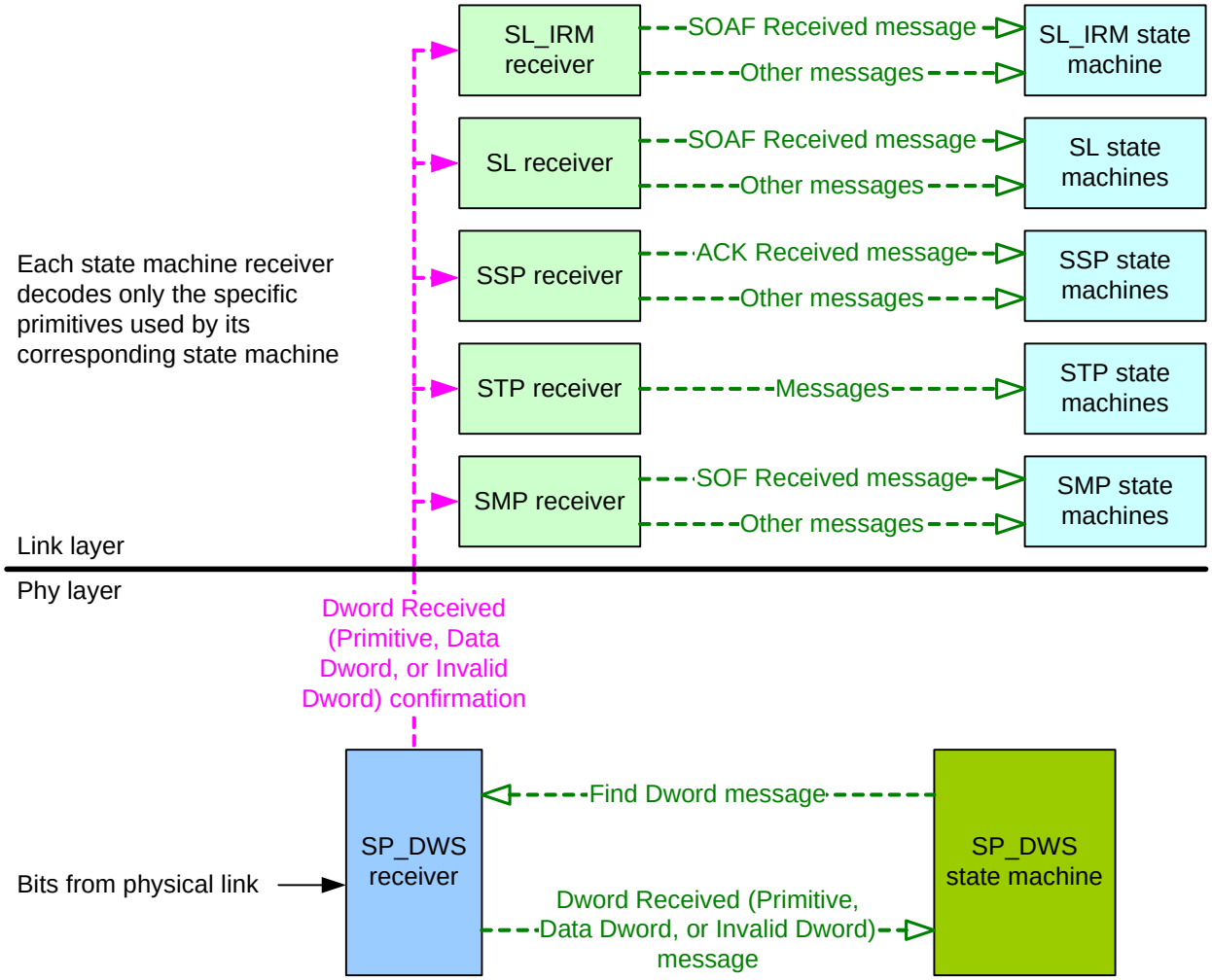


Figure 9 — Receive data path in a SAS phy [changed SL_IR to SL_IRM]

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

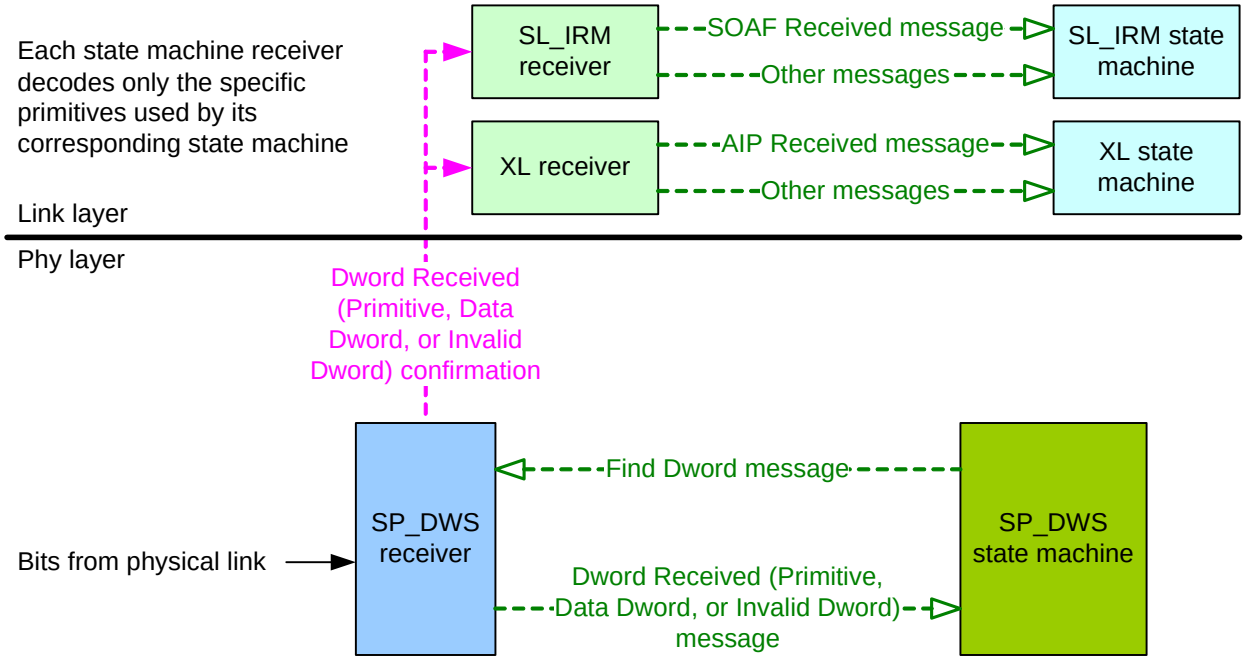


Figure 10 — Receive data path in an expander phy [changed SL_IR to SL_IRM]

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 11 shows examples of potential pathways.

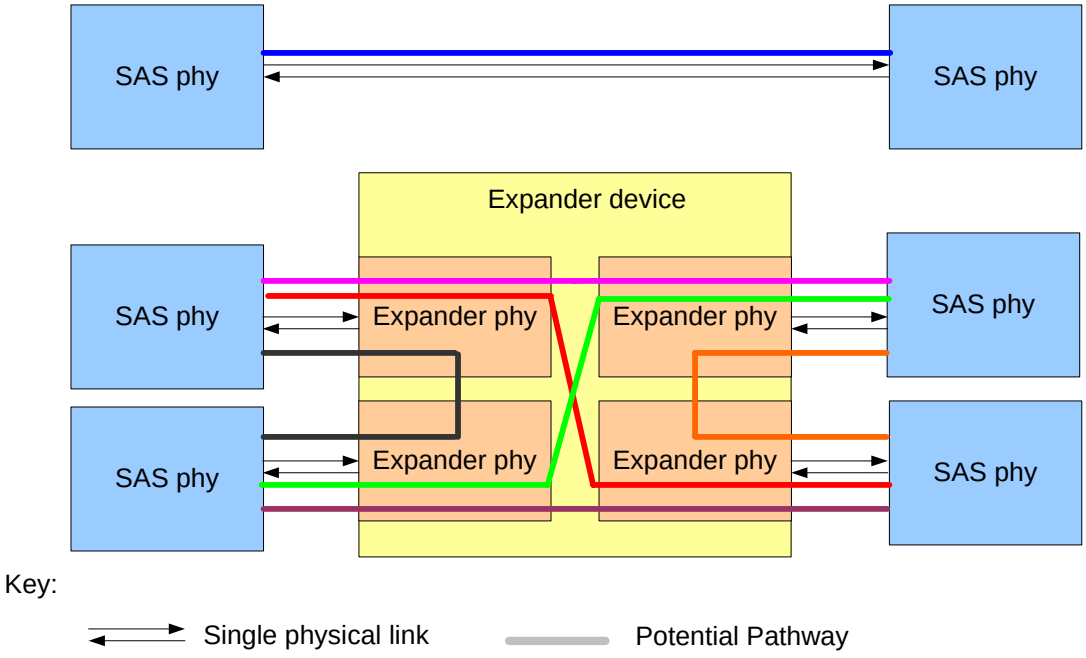


Figure 11 — Potential pathways [no changes]

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 and a SAS target port. During a connection all dwords from the SAS initiator port are forwarded to the SAS target port, and all dwords from the SAS target port are forwarded to the SAS initiator port.

A connection is pending when an OPEN address frame has been delivered along a completed pathway to the destination phy but the destination phy has not yet responded to the connection request. A connection is established when an OPEN_ACCEPT is 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 phys within the SAS port (i.e., only one connection per SAS 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:

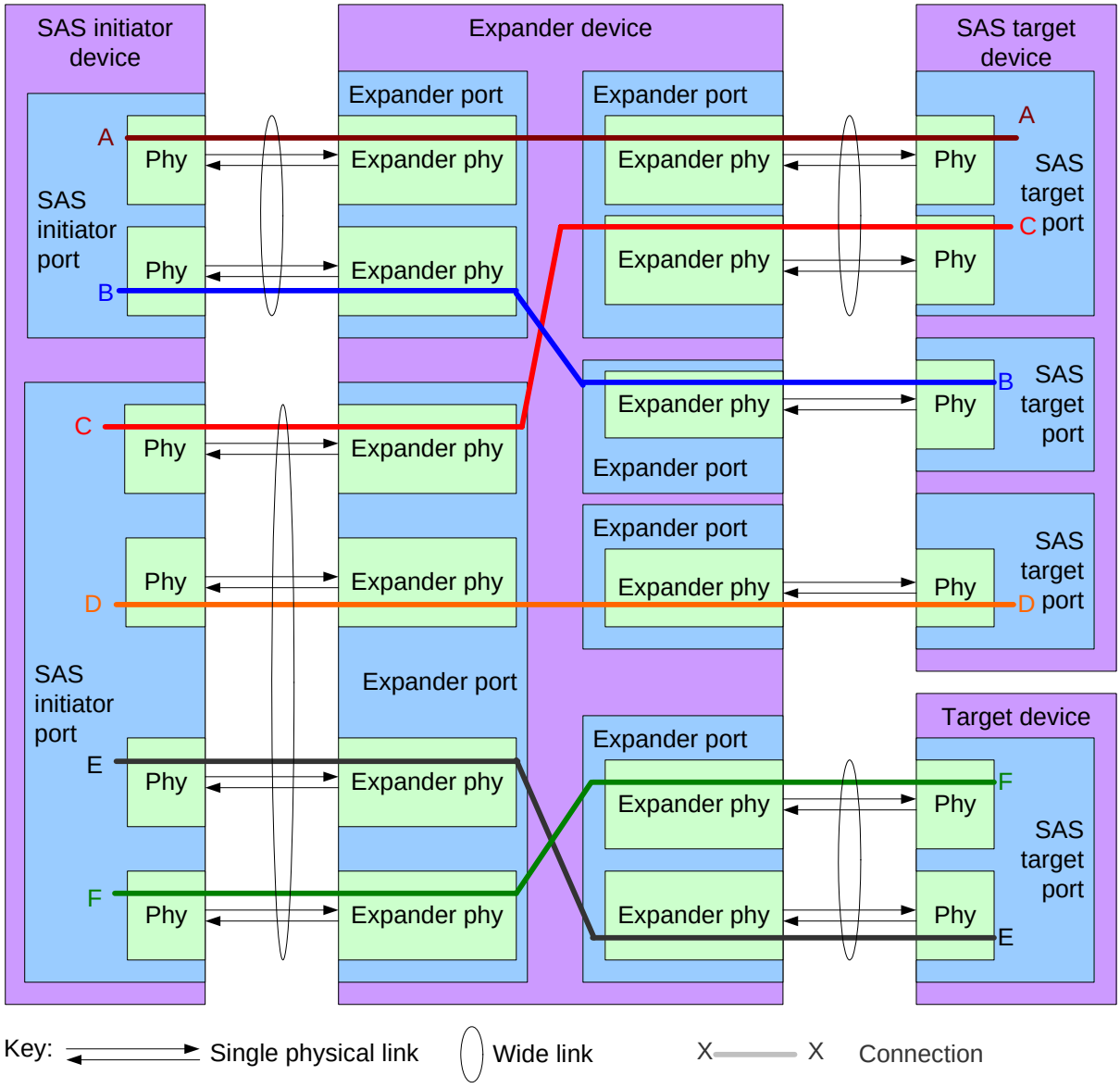
a) one SAS initiator port to multiple SAS target ports;
b) one SAS target port to multiple SAS initiator ports; or

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 12 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 12 — Multiple connections on wide ports [no changes]

4.3.4 State machines and SAS device, SAS port, and SAS phy classes

Editor's Note 2: Change SL_IR to SL_IRM in figure 39 and figure 40

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);
h) identification sequence (see 7.9); and
i) multiplexing sequence (see 7.9.xx).

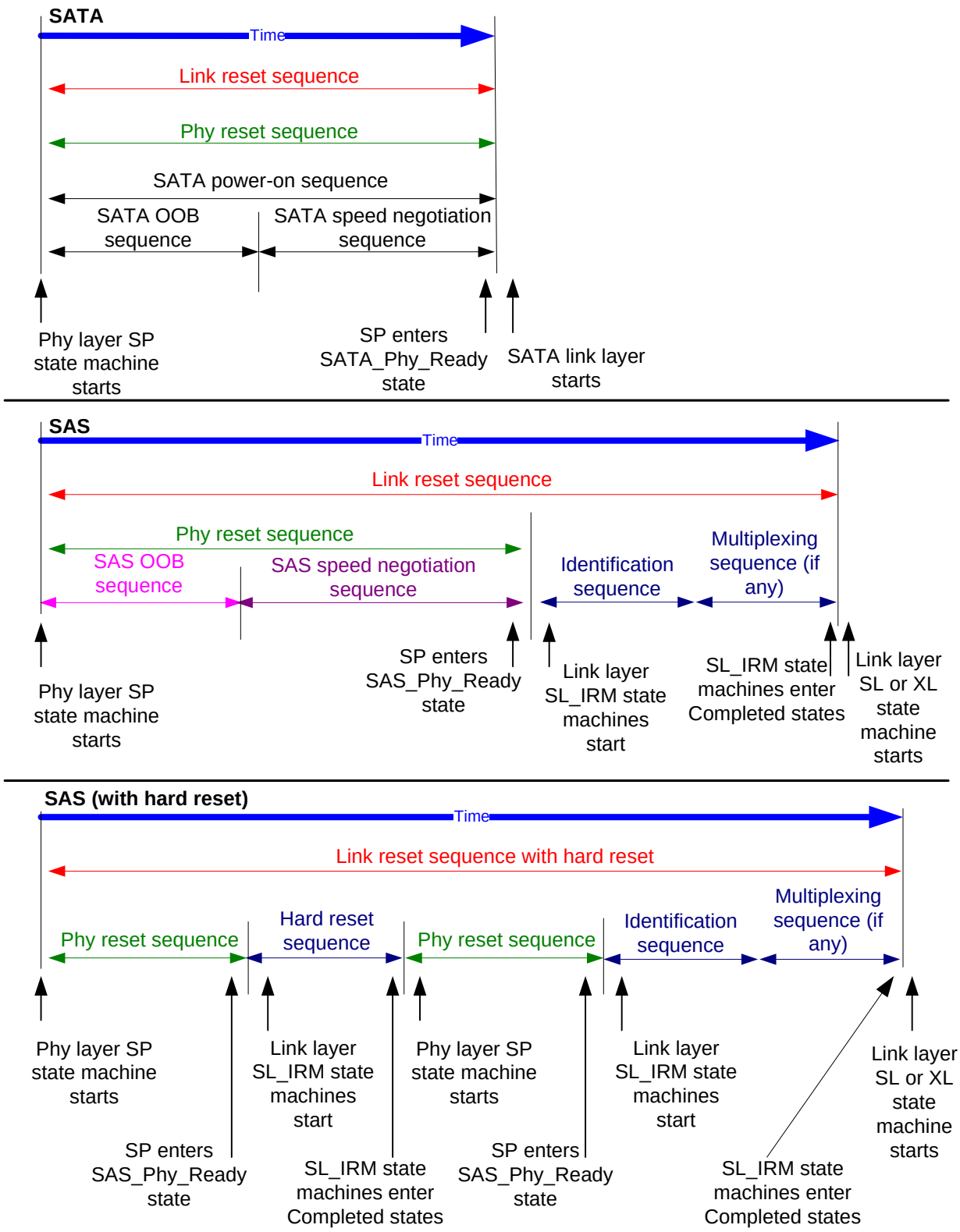


Figure 13 — Reset terminology [updated to add multiplexing sequence]

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_IRM 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

...

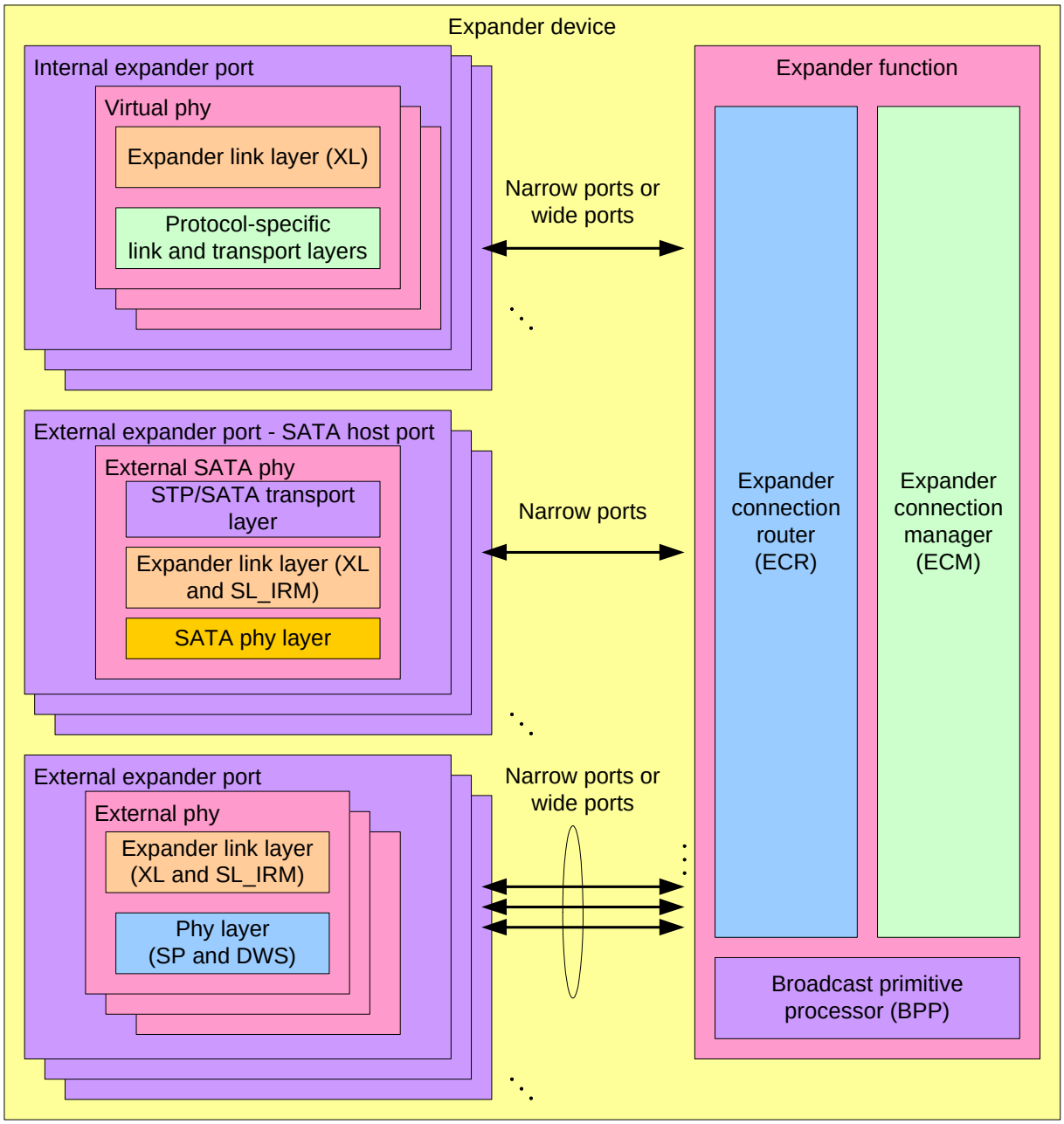


Figure 14 — Expander device model [changed SL_IR to SL_IRM]

4.6.2 Expander ports

...

Each expander phy contains an expander link layer with an XL state machine (see 7.15) and an SL_IRM 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.

...

4.6.4 Expander connection router (ECR)

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

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).

4.6.7.2 Connection request routing

The ECM shall determine how to route a connection request from a source expander phy to a destination expander phy in a different expander port if the destination expander phy is enabled and operating at a valid physical link rate (e.g., the DISCOVER function reports a NEGOTIATED PHYSICAL LINK RATE field set to G1 (i.e., 8h) or G2 (i.e., 9h)) using the following precedence:

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

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

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

An expander phy that is multiplexing supports multiple connections at the same time, each with a connection rate limited by the logical link rate.

Editor's Note 3: are more changes needed here to clarify that logical phys are the ones that communicate with the ECM?

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 algorithm used to determine 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 four 1,5 Gbps logical links; and
b) multiplex each 3 Gbps physical link into two 1,5 Gbps logical links.

4.9 Phy event information

...

Table 2 — 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 the phy layer

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.

6.8 SP state machine

6.8.4 SAS speed negotiation states

6.8.4.1 SP state machine overview

...

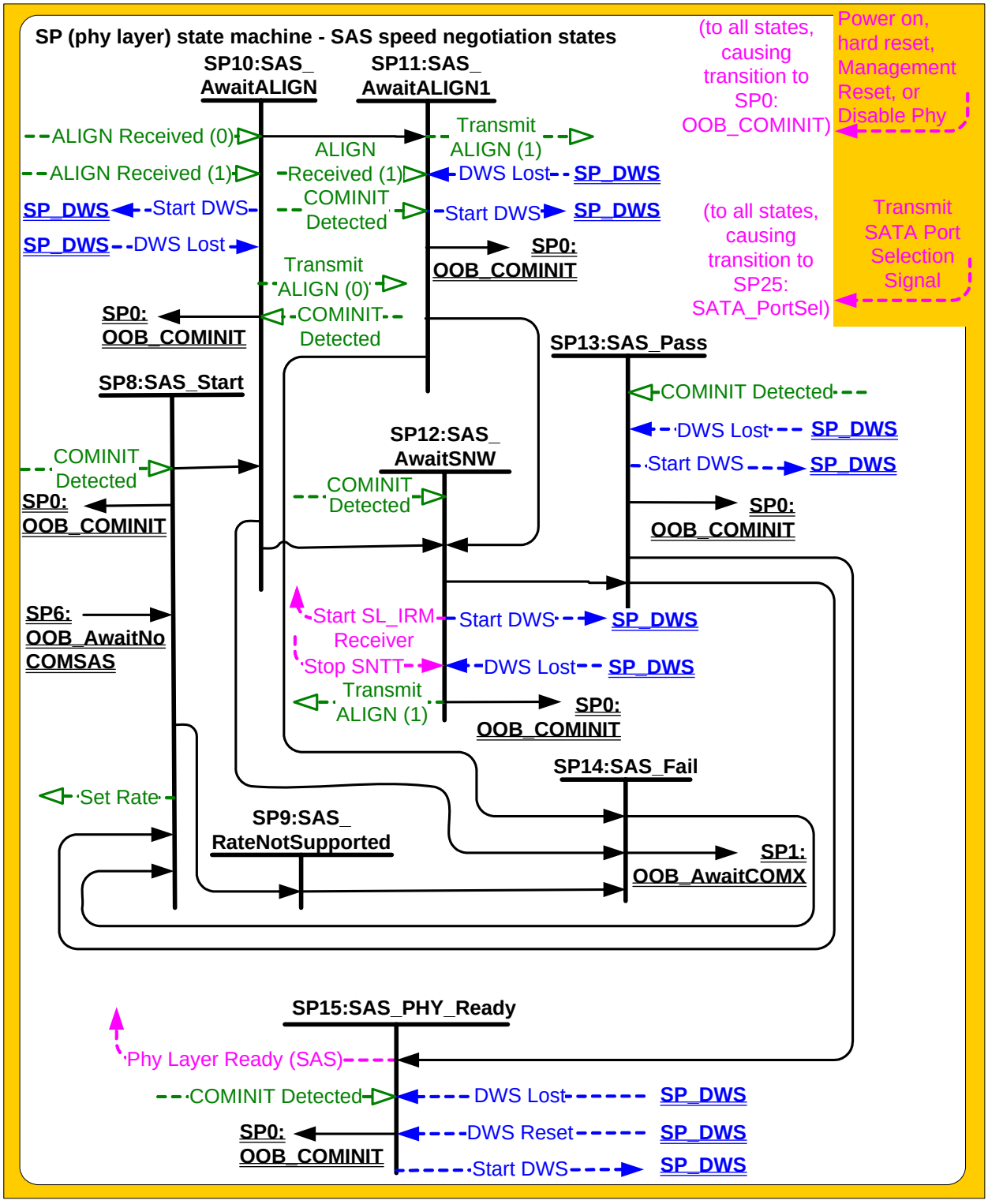


Figure 15 — SP (phy layer) state machine - SAS speed negotiation states

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_IRM Receiver confirmation to the link layer.

...

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 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 phy is not multiplexed into more than one logical phy, 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.

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;
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.

If the phy is not multiplexed into more than one logical phy, 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.

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.

6.9.2 SP_DWS receiver

...

The SP_DWS receiver also sends Dword Received confirmations to the link layer state machine receivers (e.g., SL_IRM, SL, SSP, SMP, and XL).

Changes to the link layer

Define the multiplexing sequence and the MUX primitives.

7.2.2 Primitive summary

Table 3 defines the primitives not specific to the type of connection.

Table 3 — 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)	NoConn				I	E	T	Single
AIP (RESERVED 1)	NoConn				I	E	T	Single
AIP (RESERVED 2)	NoConn				I	E	T	Single
AIP (RESERVED WAITING ON PARTIAL)	NoConn				I	E	T	Single
AIP (WAITING ON CONNECTION)	NoConn		E		I	E	T	Single
AIP (WAITING ON DEVICE)	NoConn		E		I	E	T	Single
AIP (WAITING ON PARTIAL)	NoConn		E		I	E	T	Single
ALIGN (0)	All	I	E	T	I	E	T	Single
ALIGN (1)	All	I	E	T	I	E	T	Single
ALIGN (2)	All	I	E	T	I	E	T	Single
ALIGN (3)	All	I	E	T	I	E	T	Single
BREAK	All	I	E	T	I	E	T	Redundant
BROADCAST (CHANGE)	NoConn	I	E		I	E	T	Redundant
BROADCAST (SES)	NoConn			T	I	E	T	Redundant
BROADCAST (RESERVED 1)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 2)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 3)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 4)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 0)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 1)	NoConn				I	E	T	Redundant
CLOSE (CLEAR AFFILIATION)	STP	I					T	Triple
CLOSE (NORMAL)	Conn	I		T	I		T	Triple
CLOSE (RESERVED 0)	Conn				I		T	Triple
CLOSE (RESERVED 1)	Conn				I		T	Triple
EOAF	NoConn	I	E	T	I	E	T	Single
ERROR	All		E		I	E	T	Single
HARD_RESET	NoConn	I	E		I	E	T	Redundant

Table 3 — 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	
MUX (0)	NoConn	I	E	T	I	E	T	Single
MUX (1)	NoConn	I	E	T	I	E	T	Single
MUX (2)	NoConn	I	E	T	I	E	T	Single
MUX (3)	NoConn	I	E	T	I	E	T	Single
NOTIFY (ENABLE SPINUP)	All	I	E				T	Single
NOTIFY (POWER FAILURE EXPECTED)	All	I	E				T	Single
NOTIFY (RESERVED 1)	All				I	E	T	Single
NOTIFY (RESERVED 2)	All				I	E	T	Single
OPEN_ACCEPT	NoConn	I		T	I		T	Single
OPEN_REJECT (BAD DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	NoConn	I	E	T	I		T	Single
OPEN_REJECT (NO DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (PATHWAY BLOCKED)	NoConn		E		I		T	Single
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	NoConn	I		T	I		T	Single
OPEN_REJECT (RESERVED ABANDON 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 2)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 3)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 1)	NoConn				I		T	Single
OPEN_REJECT (RETRY)	NoConn	I		T	I		T	Single
OPEN_REJECT (STP RESOURCES BUSY)	NoConn		E	T	I			Single
OPEN_REJECT (WRONG DESTINATION)	NoConn	I		T	I		T	Single
SOAF	NoConn	I	E	T	I	E	T	Single

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

[b] The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:
a) I for SAS initiator ports;
b) E for expander ports; and
c) T for SAS target ports.
Expander ports are not considered originators of primitives that are passing through from expander port to expander port.

[c] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

7.2.3 Primitive encodings

Table 4 defines the primitive encoding for primitives not specific to type of connection.

Table 4 — 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
MUX (0)	K28.5	D02.0	D16.7	D31.4
MUX (1)	K28.5	D07.3	D04.7	D30.0
MUX (2)	K28.5	D16.7	D24.0	D27.4

Table 4 — Primitive encoding for primitives not specific to type of connection (part 2 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
MUX (3)	K28.5	D24.0	D01.4	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
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.

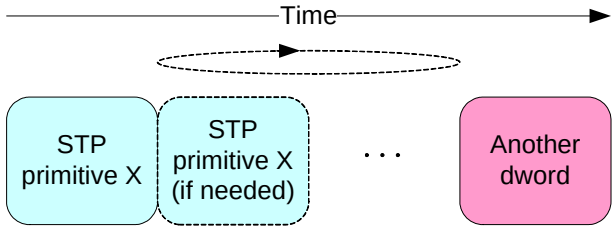
Receivers count each primitive received that is labeled as a single primitive sequence as a distinct single primitive sequence.

ALIGNs, NOTIFYs, and MUXs are called deletable primitives.

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 16 shows an example of transmitting a repeated primitive sequence.

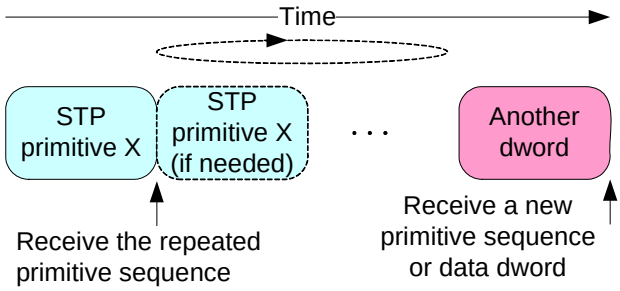


NOTE: Another dword is a dword other than a deletable primitive or STP primitive X

Figure 16 — 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 17 shows an example of receiving a repeated primitive sequence.



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

Figure 17 — 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 Figure 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 18 shows examples of triple primitive sequences.

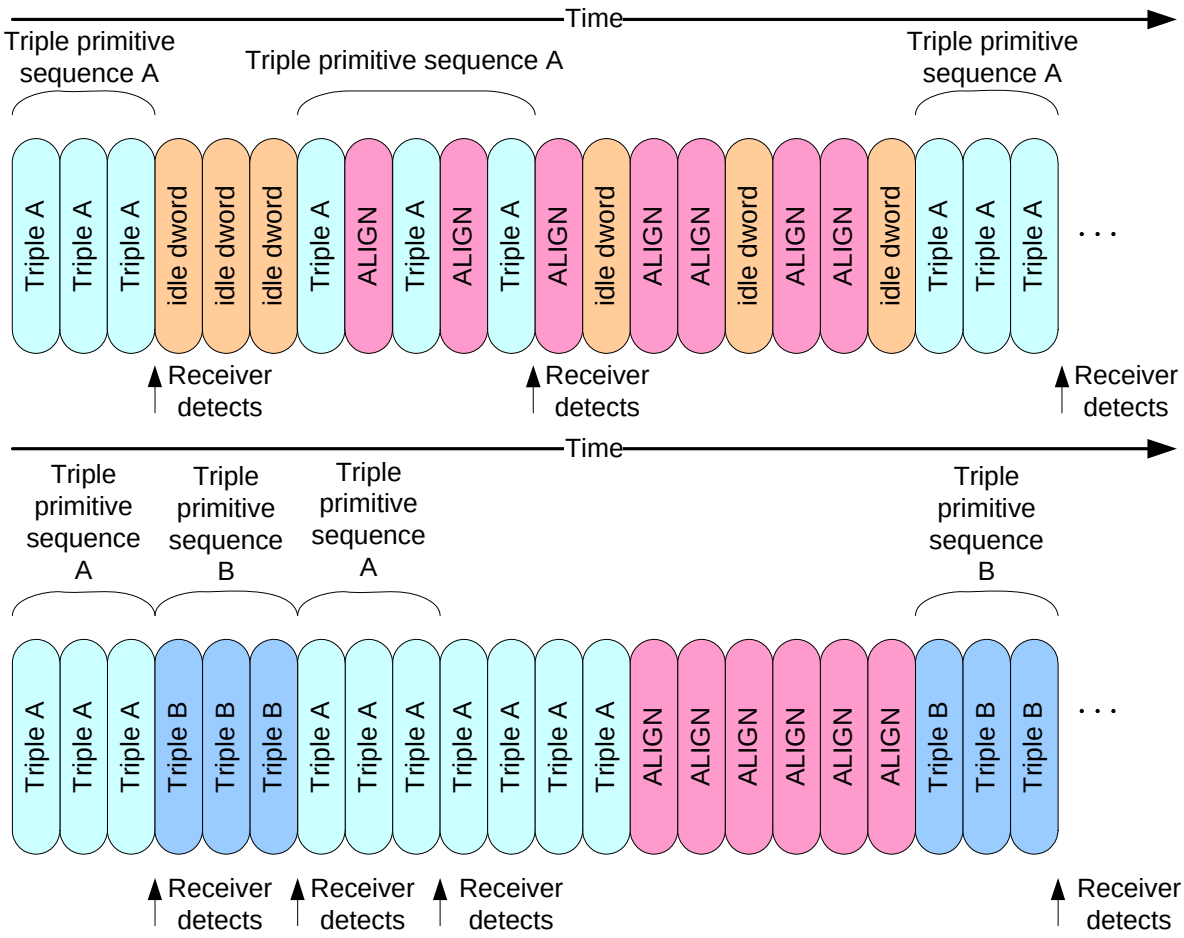


Figure 18 — Triple primitive sequence [no changes]

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 19 shows examples of redundant primitive sequences.

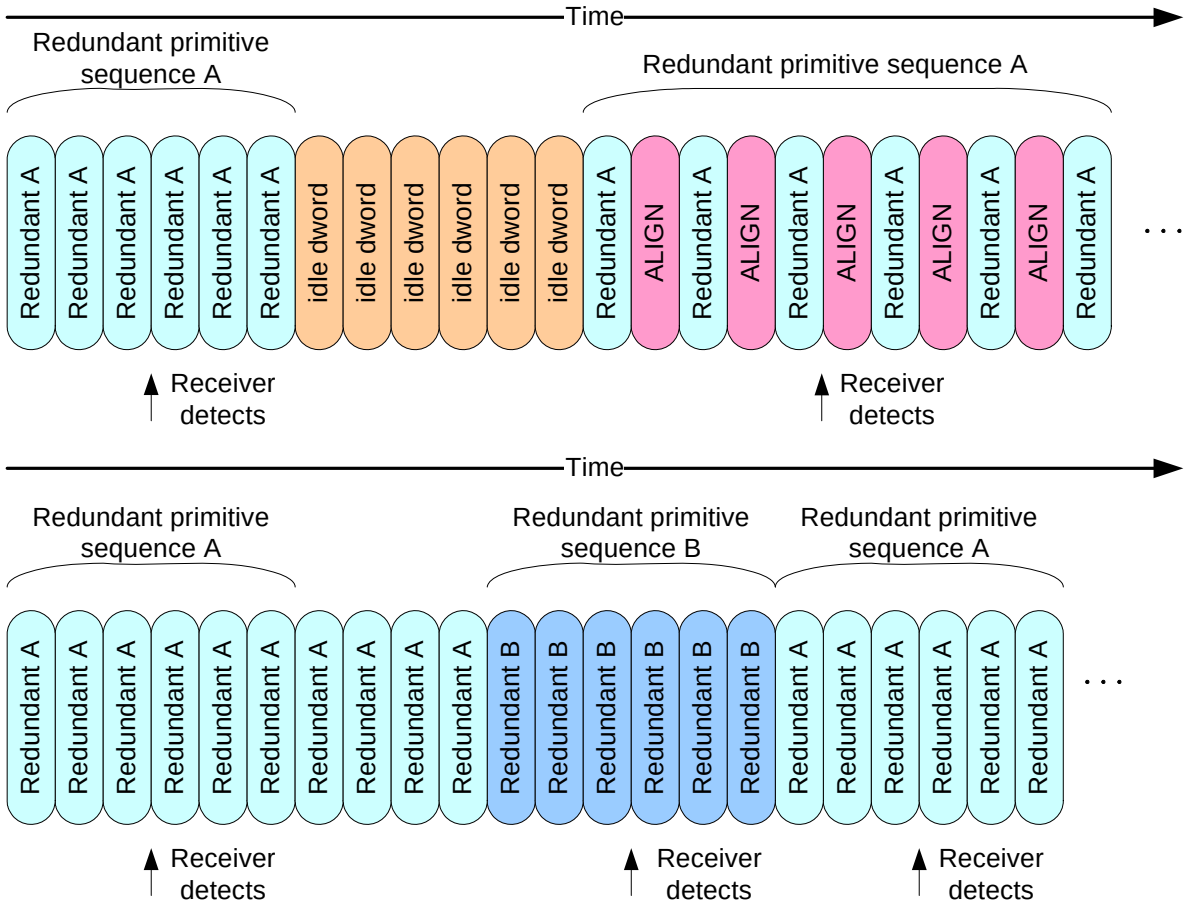


Figure 19 — Redundant primitive sequence [no changes]

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.

Table 5 defines the different versions of ALIGN primitives.

Table 5 — 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 1 - 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.n MUX (Multiplex)

MUX is sent by a phy to negotiate multiplexing during the multiplexing sequence and may be transmitted in place of any 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) if multiplexing is enabled to periodically 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.

The versions of MUX are defined in table 6.

Table 6 — MUX primitives

Primitive	Description
MUX (0)	Establishes the position of dwords in logical link 0.
MUX (1)	Establishes the position of dwords in logical link 1.
MUX (2)	For a physical link multiplexed into two logical links, establishes the position of dwords in logical link 0. For a physical link multiplexed into four logical links, establishes the position of dwords in logical link 2.
MUX (3)	For a physical link multiplexed into two logical links, establishes the position of dwords in logical link 1. For a physical link multiplexed into four logical links, establishes the position of dwords in logical link 3.

Phys shall rotate through MUX (0), MUX (1), MUX (2), and MUX (3) for all MUXs transmitted during the multiplexing sequence.

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.

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 7.

Table 7 — 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.

7.3 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 43 (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 20, 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 8.

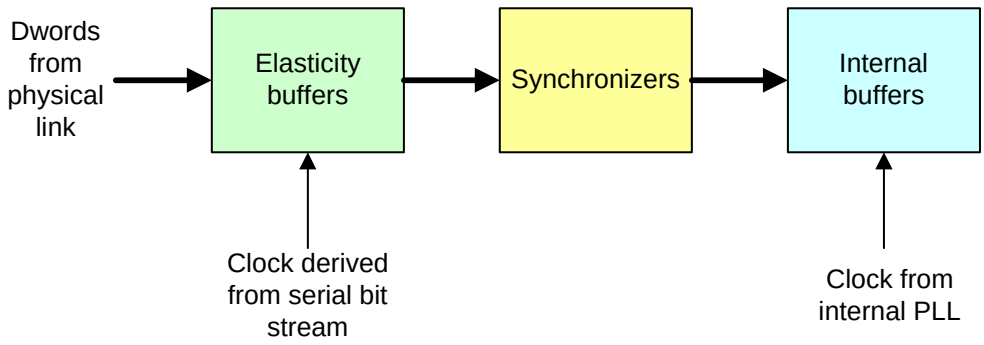


Figure 20 — Elasticity buffers [no changes]

A phy that is the original source for the dword stream (i.e., a phy that is not an expander phy forwarding dwords from another expander phy) shall insert one ~~ALIGN or NOTIFY~~deletable primitive for clock skew management as described in table 8.

Table 8 — 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

~~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 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 8 when forwarding to a SAS physical 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 2 - 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 link and is not required to ensure that any ~~ALIGNs and/or NOTIFYs~~deletable primitives it transmits are in pairs.

NOTE 3 - 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 9 lists the scrambling for different types of data dwords.

Table 9 — 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.

...

7.8.2 IDENTIFY address frame

Table 10 defines the IDENTIFY address frame format used for the identification sequence. The IDENTIFY address frame is sent after the phy reset sequence completes if the physical link is a SAS physical link.

Table 10 — IDENTIFY address frame format

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

The DEVICE TYPE field specifies the type of device containing the phy, and is defined in table 11.

Table 11 — 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.

The REQUESTED LOGICAL LINKS field indicates the maximum number of logical links the phy supports via multiplexing and is defined in table 30.

Table 12 — REQUESTED LOGICAL LINKS field

Code	Physical link rate		
	6 Gbps	3 Gbps	1,5 Gbps
00b	One logical link (i.e., no multiplexing)		
01b	Two logical links		One logical link
10b	Four logical links	Two logical links	
11b	Reserved		

If the phy is controlled by an SMP target port, the REQUESTED LOGICAL LINKS field is based on the REQUESTED LOGICAL LINK RATE field in the SMP PHY CONTROL and DISCOVER functions as described in table 30.

Table 13 — REQUESTED LOGICAL LINKS field

REQUESTED LOGICAL LINK RATE field in DISCOVER	Physical link rate	Resulting REQUESTED LOGICAL LINKS field
8h (i.e., 1,5 Gbps)	1,5 Gbps	00b, 01b, or 10b (i.e., one 1,5 Gbps logical link)
	3 Gbps	01b or 10b (i.e., two 1,5 Gbps logical links)
	6 Gbps	10b (i.e., four 1,5 Gbps logical links)
9h (i.e., 3 Gbps)	1,5 Gbps	00b, 01b, or 10b (i.e., one 1,5 Gbps logical link)
	3 Gbps	00b (i.e., two 1,5 Gbps logical links)
	6 Gbps	01b or 10b (i.e., two 3 Gbps logical links)
Ah (i.e., 6 Gbps)	1,5 Gbps	00b, 01b, or 10b (i.e., one 1,5 Gbps logical link)
	3 Gbps	00b or 01b (i.e., one 3 Gbps logical link)
	6 Gbps	00b (i.e., one 6 Gbps logical link)
All others	Any	Not defined

An SSP INITIATOR PORT bit set to one ~~specifies~~indicates that an SSP initiator port is present. An SSP INITIATOR PORT bit set to zero ~~specifies~~indicates that an SSP initiator port is not present. Expander devices shall set the SSP INITIATOR PORT bit to zero.

An STP INITIATOR PORT bit set to one ~~specifies~~indicates that an STP initiator port is present. An STP INITIATOR PORT bit set to zero ~~specifies~~indicates that an STP initiator port is not present. Expander devices shall set the STP INITIATOR PORT bit to zero.

An SMP INITIATOR PORT bit set to one ~~specifies~~indicates that an SMP initiator port is present. An SMP INITIATOR PORT bit set to zero ~~specifies~~indicates that an SMP initiator port is not present. Expander devices may set the SMP INITIATOR PORT bit to one.

An SSP TARGET PORT bit set to one ~~specifies~~indicates that an SSP target port is present. An SSP TARGET PORT bit set to zero ~~specifies~~indicates that an SSP target port is not present. Expander devices shall set the SSP TARGET PORT bit to zero.

An STP TARGET PORT bit set to one ~~specifies~~indicates that an STP target port is present. An STP TARGET PORT bit set to zero ~~specifies~~indicates that an STP target port is not present. Expander devices shall set the STP TARGET PORT bit to zero.

An SMP TARGET PORT bit set to one ~~specifies~~indicates that an SMP target port is present. An SMP TARGET PORT bit set to zero ~~specifies~~indicates that an SMP target port is not present. Expander devices shall set the SMP TARGET PORT bit to one.

For SAS ports, the SAS ADDRESS field ~~specifies~~indicates the port identifier (see 4.2.6) of the SAS port transmitting the IDENTIFY address frame. For expander ports, the SAS ADDRESS field ~~specifies~~indicates the device name (see 4.2.4) of the expander device transmitting the IDENTIFY address frame.

The PHY IDENTIFIER field ~~specifies~~indicates the phy identifier of the phy transmitting the IDENTIFY address frame.

See 4.1.3 for additional requirements concerning the DEVICE TYPE field, SSP INITIATOR PORT bit, STP INITIATOR PORT bit, SMP INITIATOR PORT bit, SSP TARGET PORT bit, STP TARGET PORT bit, SMP TARGET PORT bit, and SAS ADDRESS field.

The CRC field is defined in 7.8.1.

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 14.

Table 14 — CONNECTION RATE field

Code	Description
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
All others	Reserved

...

7.9 Identification and hard reset sequence

7.9.1 Identification and hard reset sequence overview

After the phy reset sequence has been completed indicating the physical link is using SAS rather than SATA, each phy transmits either:

a) an IDENTIFY address frame (see); or
b) a HARD_RESET primitive sequence.

Each phy receives an IDENTIFY address frame or a HARD_RESET primitive sequence from the phy to which it is attached. The combination of a phy reset sequence, an optional hard reset sequence, ~~and~~ an identification sequence, and an optional multiplexing 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 Fanout expander device rules

After completing the ~~identification~~link reset sequence on a phy and completing internal initialization, the ECM within a fanout expander device shall be capable of routing connection requests through that phy. The expander device may return OPEN_REJECT (NO DESTINATION) until it is ready 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 fanout expander device that does not have a configurable expander route table shall follow the SAS initiator device rules (see 7.9.2) to perform a discover process.

The ECM of a fanout expander device that has a configurable expander route table is dependent on the completion of the discover process (see 4.7) for routing connection requests using the table routing method.

7.9.4 Edge expander device rules

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

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

7.xx Multiplexing

If a phy both transmits and receives IDENTIFY address frames during the identification sequence indicating that multiplexing is supported, it shall enable multiplexing to the highest common number of logical links after the identification sequence completes. This is called the multiplexing sequence.

The phy shall transmit MUX repeatedly, rotating through MUX (0), MUX (1), MUX (2), and MUX (3) in order. The phy shall not transmit ALIGNs and/or NOTIFYs (e.g., for clock skew management (see 7.3)) during the multiplexing sequence.

The phy shall ignore all incoming dwords except MUX primitives. Incoming MUXes during the multiplexing sequence are not accompanied by ALIGNs and/or NOTIFYs, so the phy shall process them in logic running off the received clock, without using an elasticity buffer.

After the phy receives at least 3 MUX primitives confirming the position of dwords in each logical link, it shall continue transmitting at least 24 MUX primitives. The phy shall then stop transmitting MUX and the logical phys shall start transmitting dwords for the logical links in the corresponding positions as shown in figure 21.

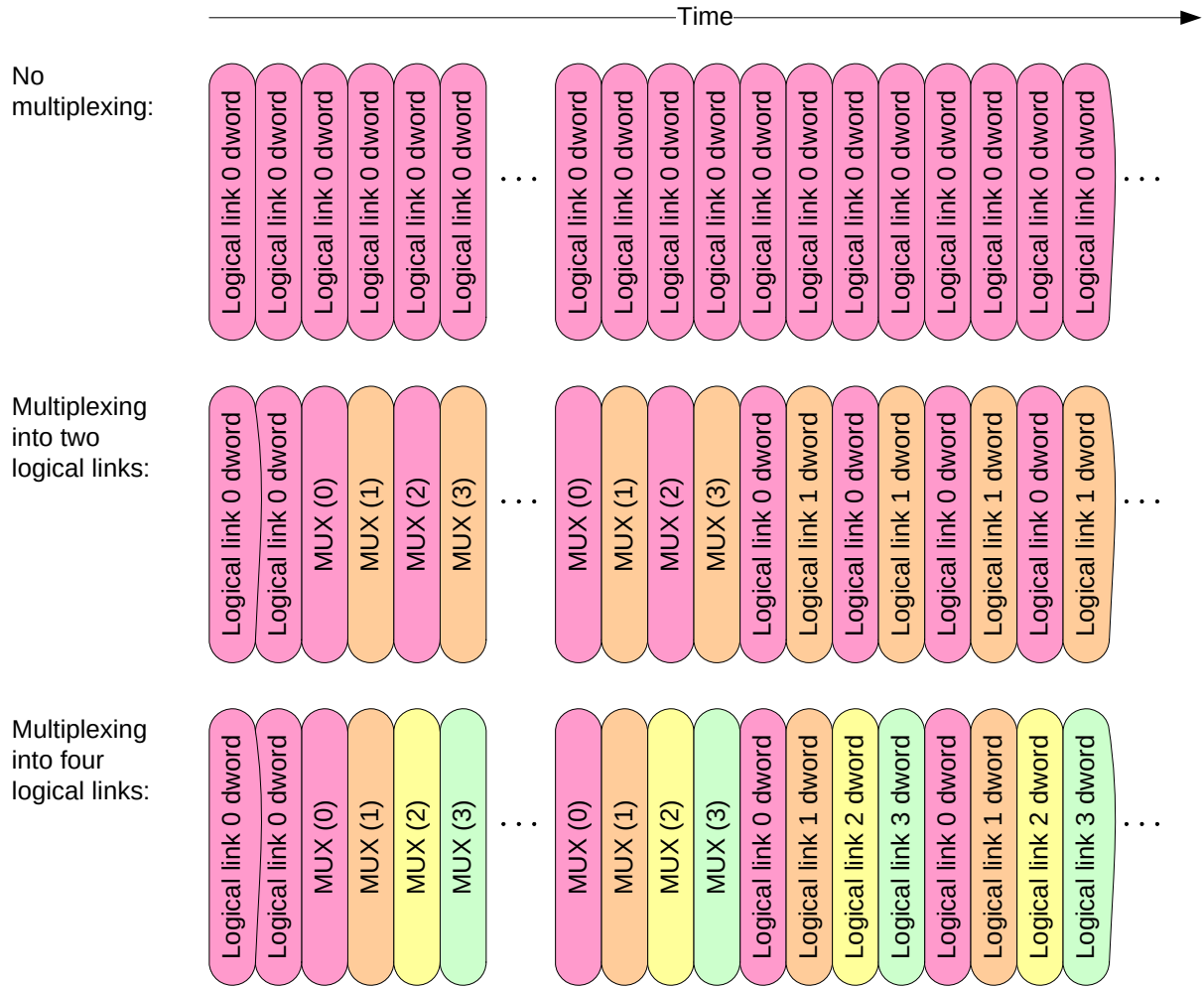


Figure 21 — Multiplexing

The first dword in a logical link may be any type of dword (e.g., any primitive including an ALIGN or NOTIFY, or a data dword). Each logical phy shall honor the ALIGN insertion rate rules in 7.xx. The logical phys shall ignore MUX primitives.

The phy shall establish the incoming logical links based on the received MUX primitives (e.g., MUX (1) indicates the position of logical link 1). It shall receive 3 MUX primitives confirming each logical link before using the logical link.

The phy shall handle errors during the multiplexing sequence 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;
c) If the phy receives a MUX primitive that does not match the MUX primitive expected in that position (e.g., it receives MUX(0) followed by MUX (2)), it shall shift the expected positions;
d) If the phy transmits MUX primitives for 1 ms without receiving MUX identifying the positions of each logical link, it shall restart the link reset sequence; and
e) If the phy finishes transmitting MUX primitives and starts transmitting non-MUX primitives, but does not stop receiving MUX primitives in all logical links for 1 ms, it shall restart the link reset sequence.

If the phy 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 change multiplexing until a new link reset sequence. The phy shall not transmit MUX and shall ignore any incoming MUX primitives.

Editor's Note 4: Goals: Must tolerate single bit errors and up to seven bit (because of DFE - see 06-028) errors, should tolerate more. Errors could happen during the first MUX primitives, middle ones, or the last ones, or after multiplexing is established.

Once the multiplexing sequence is complete, the phy shall periodically transmit MUX instead of ALIGN to confirm the logical link numbers (e.g., for convenience for logic analyzers). It should transmit a MUX on each logical link once every millisecond. Transmitting NOTIFY has higher priority than transmitting MUX.

Editor's Note 5: Is one MUX per ms sufficient?

7.9.5 SL_IRM (link layer identification, ~~and~~ hard reset, and multiplexing) state machines

---

Editor's Note 6: Move 7.9.5 up one level so it becomes 7.11.

7.9.5.1 SL_IRM state machines overview

The SL_IRM (link layer identification, ~~and~~ hard reset, and multiplexing) state machines control the flow of dwords on the physical link that are associated with the identification and hard reset sequences. The state machines are as follows:

a) SL_IRM_TIR (transmit IDENTIFY or HARD_RESET) state machine (see 7.9.5.3);
b) SL_IRM_RIF (receive IDENTIFY address frame) state machine (see 7.9.5.4); and
c) SL_IRM_IRC (identification and hard reset control) state machine (see 7.9.5.5).

The SL_IRM state machines send the following messages to the SL state machines (see 7.14) in SAS devices or the XL (see 7.15) state machine in expander devices:

a) Enable Disable SAS Link (Enable); and
b) Enable Disable SAS Link (Disable).

The SL_IRM_IRC state machine shall maintain the timers listed in table 15.

Table 15 — SL_IRM_IRC timers

Timer	Initial value
Receive Identify Timeout timer	1 ms

Figure 22 shows the SL_IRM state machines.

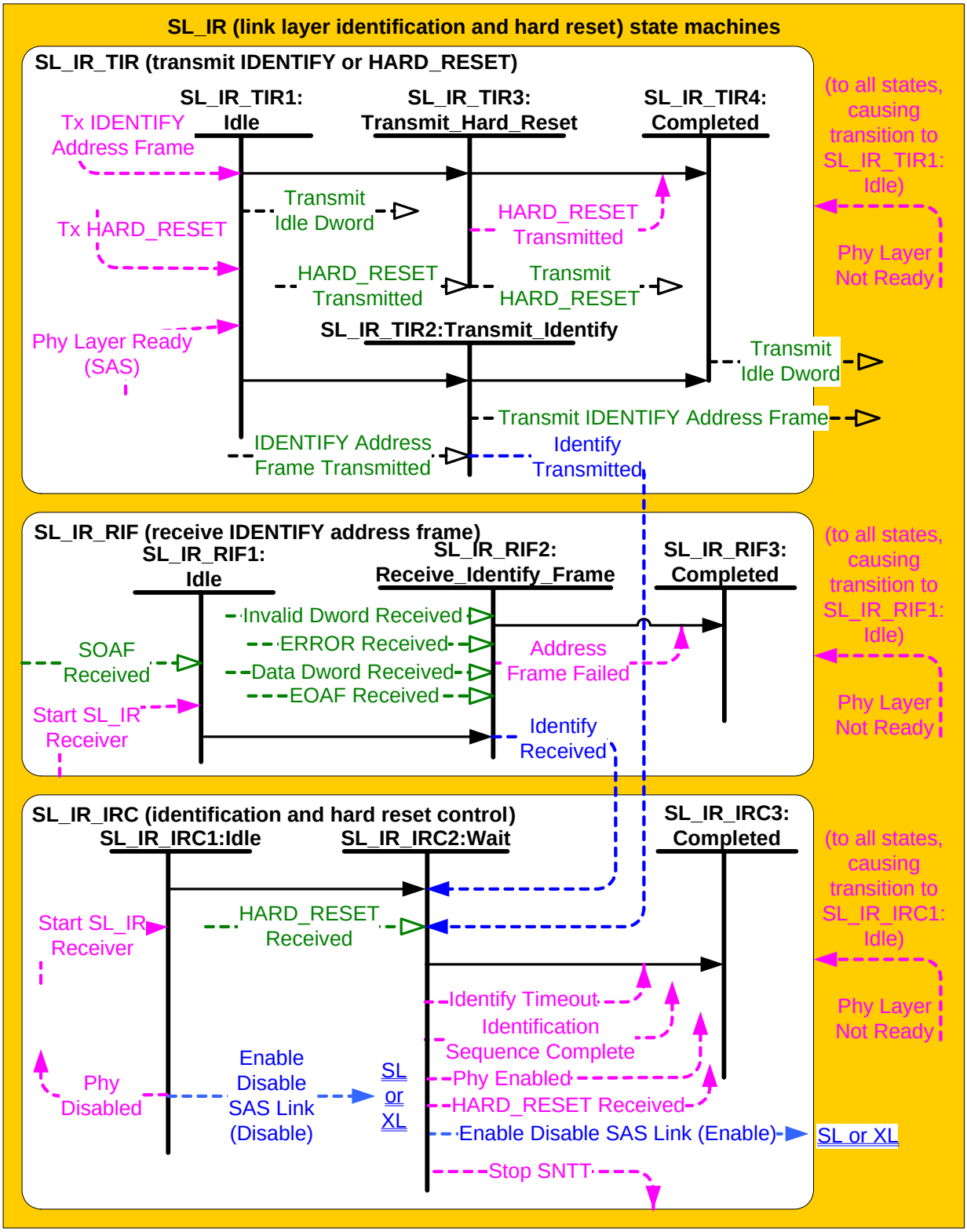


Figure 22 — SL_IRM (link layer identification, ~~and~~ hard reset, and multiplexing) state machines

7.9.5.2 SL_IRM transmitter and receiver

The SL_IRM transmitter receives the following messages from the SL_IRM state machines indicating primitive sequences, frames, and dwords to transmit:

a) Transmit IDENTIFY Address Frame;
b) Transmit HARD_RESET; and
c) Transmit Idle Dword.

The SL_IRM transmitter sends the following messages to the SL_IRM state machines:

a) HARD_RESET Transmitted; and
b) IDENTIFY Address Frame Transmitted.

The SL_IRM receiver sends the following messages to the SL_IRM state machines indicating primitive sequences and dwords received from the SP_DWS receiver (see 6.9.2):

a) SOAF Received;
b) Data Dword Received;
c) EOAF Received;
d) ERROR Received;
e) Invalid Dword Received; and
f) HARD_RESET Received.

The SL_IRM receiver shall ignore all other dwords.

7.9.5.3 SL_IRM_TIR (transmit IDENTIFY or HARD_RESET) state machine

7.9.5.3.1 SL_IRM_TIR state machine overview

The SL_IRM_TIR state machine's function is to transmit a single IDENTIFY address frame or HARD_RESET primitive after the phy layer enables the link layer. This state machine consists of the following states:

a) SL_IRM_TIR1:Idle (see 7.9.5.3.2)(initial state);
b) SL_IRM_TIR2:Transmit_Identify (see 7.9.5.3.3);
c) SL_IRM_TIR3:Transmit_Hard_Reset (see 7.9.5.3.4); and
d) SL_IRM_TIR4:Completed (see 7.9.5.3.5).

This state machine shall start in the SL_IRM_TIR1:Idle state. This state machine shall transition to the SL_IRM_TIR1:Idle state from any other state after receiving a Phy Layer Not Ready confirmation.

7.9.5.3.2 SL_IRM_TIR1:Idle state

7.9.5.3.2.1 State description

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IRM transmitter.

7.9.5.3.2.2 Transition SL_IRM_TIR1:Idle to SL_IRM_TIR2:Transmit_Identify

This transition shall occur after both:

a) a Phy Layer Ready (SAS) confirmation is received; and
b) a Tx IDENTIFY Address Frame request is received.

7.9.5.3.2.3 Transition SL_IRM_TIR1:Idle to SL_IRM_TIR3:Transmit_Hard_Reset

This transition shall occur after both:

a) a Phy Layer Ready (SAS) confirmation is received; and
b) a Tx HARD_RESET request is received.

7.9.5.3.3 SL_IRM_TIR2:Transmit_Identify state

7.9.5.3.3.1 State description

Upon entry into this state, this state shall send a Transmit IDENTIFY Address Frame message to the SL_IRM transmitter.

After this state receives an IDENTIFY Address Frame Transmitted message, this state shall send an Identify Transmitted message to the SL_IRM_IRC state machine.

7.9.5.3.3.2 Transition SL_IRM_TIR2:Transmit_Identify to SL_IRM_TIR4:Completed

This transition shall occur after sending an Identify Transmitted message to the SL_IRM_IRC state machine.

7.9.5.3.4 SL_IRM_TIR3:Transmit_Hard_Reset state

7.9.5.3.4.1 State description

Upon entry into this state, this state shall send a Transmit HARD_RESET message to the SL_IRM transmitter.

After this state receives a HARD_RESET Transmitted message, this state shall send a HARD_RESET Transmitted confirmation to the management application layer.

7.9.5.3.4.2 Transition SL_IRM_TIR3:Transmit_Hard_Reset to SL_IRM_TIR4:Completed

This transition shall occur after sending a HARD_RESET Transmitted confirmation to the management application layer.

Editor's Note 7: Add state(s) to handle the multiplexing sequence.

7.9.5.3.5 SL_IRM_TIR4:Completed state

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IRM transmitter.

7.9.5.4 SL_IRM_RIF (receive IDENTIFY address frame) state machine

7.9.5.4.1 SL_IRM_RIF state machine overview

The SL_IRM_RIF state machine receives an IDENTIFY address frame and checks the IDENTIFY address frame to determine if the frame should be accepted or discarded by the link layer.

This state machine consists of the following states:

a) SL_IRM_RIF1:Idle (see 7.9.5.4.2)(initial state);
b) SL_IRM_RIF2:Receive_Identify_Frame (see 7.9.5.4.3); and
c) SL_IRM_RIF3:Completed (see 7.9.5.4.4).

This state machine shall start in the SL_IRM_RIF1:Idle state. This state machine shall transition to the SL_IRM_RIF1:Idle state from any other state after receiving a Phy Layer Not Ready confirmation.

7.9.5.4.2 SL_IRM_RIF1:Idle state

7.9.5.4.2.1 State description

This state waits for an SOAF to be received from the physical link, indicating an address frame is arriving.

7.9.5.4.2.2 Transition SL_IRM_RIF1:Idle to SL_IRM_RIF2:Receive_Identify_Frame

This transition shall occur after both:

a) a Start SL_IRM Receiver confirmation is received; and

b) an SOAF Received message is received.

7.9.5.4.3 SL_IRM_RIF2:Receive_Identify_Frame state

7.9.5.4.3.1 State description

This state receives the dwords of an address frame and the EOAF.

If this state receives an SOAF Received message, then this state shall discard the address frame (i.e., the subsequent Data Dword Received and EOAF Received messages) and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

If this state receives more than eight Data Dword Received messages after an SOAF Received message and before an EOAF Received message, then this state shall discard the address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

If this state receives an Invalid Dword Received message or an ERROR Received message after an SOAF Received message and before an EOAF Received message, then this state shall:

a) ignore the invalid dword or ERROR; or
b) discard the address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

After receiving an EOAF Received message, this state shall check if it the received frame is a valid IDENTIFY address frame.

This state shall accept an IDENTIFY address frame and send an Identify Received message to the SL_IRM_IRC state machine if:

a) the ADDRESS FRAME TYPE field is set to Identify;
b) the number of bytes between the SOAF and EOAF is 32; and
c) the CRC field contains a valid CRC.

Otherwise, this state shall discard the IDENTIFY address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

7.9.5.4.3.2 Transition SL_IRM_RIF2:Receive_Identify_Frame to SL_IRM_RIF3:Completed

This transition shall occur after sending an Identify Received message or Address Frame Failed confirmation.

Editor's Note 8: Add state(s) to handle the multiplexing sequence.

7.9.5.4.4 SL_IRM_RIF3:Completed state

This state waits for a Phy Layer Not Ready confirmation.

7.9.5.5 SL_IRM_IRC (identification and hard reset control) state machine

7.9.5.5.1 SL_IRM_IRC state machine overview

The SL_IRM_IRC state machine ensures that IDENTIFY address frames have been both received and transmitted before enabling the rest of the link layer, and notifies the link layer if a HARD_RESET primitive sequence is received before an IDENTIFY address frame has been received.

This state machine consists of the following states:

a) SL_IRM_IRC1:Idle (see 7.9.5.5.2)(initial state);
b) SL_IRM_IRC2:Wait (see 7.9.5.5.3); and
c) SL_IRM_IRC3:Completed (see 7.9.5.5.4).

This state machine shall start in the SL_IRM_IRC1:Idle state. This state machine shall transition to the SL_IRM_IRC1:Idle state from any other state after receiving a Phy Layer Not Ready confirmation.

7.9.5.5.2 SL_IRM_IRC1:Idle state

7.9.5.5.2.1 State description

This state waits for the link layer to be enabled. Upon entry into this state, this state shall:

a) send an Enable Disable SAS Link (Disable) message to SL state machines (see 7.14) or XL state machine (see 7.15) halting any link layer activity; and
b) send a Phy Disabled confirmation to the port layer and the management application layer indicating that the phy is not ready for use.

7.9.5.5.2.2 Transition SL_IRM_IRC1:Idle to SL_IRM_IRC2:Wait

This transition shall occur after a Start SL_IRM Receiver confirmation is received.

7.9.5.5.3 SL_IRM_IRC2:Wait state

7.9.5.5.3.1 State description

This state ensures that an IDENTIFY address frame has been received by the SL_IRM_RIF state machine and that a IDENTIFY address frame has been transmitted by the SL_IRM_TIR state machine before enabling the rest of the link layer. The IDENTIFY address frames may be transmitted and received on the physical link in any order.

After this state receives an Identify Received message, it shall send a Stop SNTT request to the phy layer.

After this state receives an Identify Transmitted message, it shall initialize and start the Receive Identify Timeout timer. If an Identify Received message is received before the Receive Identify Timeout timer expires, this state shall:

a) send an Identification Sequence Complete confirmation to the management application layer, with arguments carrying the contents of the incoming IDENTIFY address frame;
b) send an Enable Disable SAS Link (Enable) message to the SL state machines (see 7.14) in a SAS phy or the XL state machine (see 7.15) in an expander phy indicating that the rest of the link layer may start operation; and
c) send a Phy Enabled confirmation to the port layer and the management application layer indicating that the phy is ready for use.

If the Receive Identify Timeout timer expires before an Identify Received message is received, this state shall send an Identify Timeout confirmation to the management application layer to indicate that an identify timeout occurred.

If this state receives a HARD_RESET Received message before an Identify Received message is received, this state shall send a HARD_RESET Received confirmation to the port layer and a Stop SNTT request to the phy layer.

If this state receives a HARD_RESET Received message after an Identify Received message is received, the HARD_RESET Received message shall be ignored.

7.9.5.5.3.2 Transition SL_IRM_IRC2:Wait to SL_IRM_IRC3:Completed

This transition shall occur after sending a HARD_RESET Received confirmation, Identify Timeout confirmation, or an Identification Sequence Complete and an Phy Enabled confirmation.

Editor's Note 9: Add state(s) to handle the multiplexing sequence.

7.9.5.5.4 SL_IRM_IRC3:Completed state

This state waits for a Phy Layer Not Ready confirmation.

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 16.

Table 16 — 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 23 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.

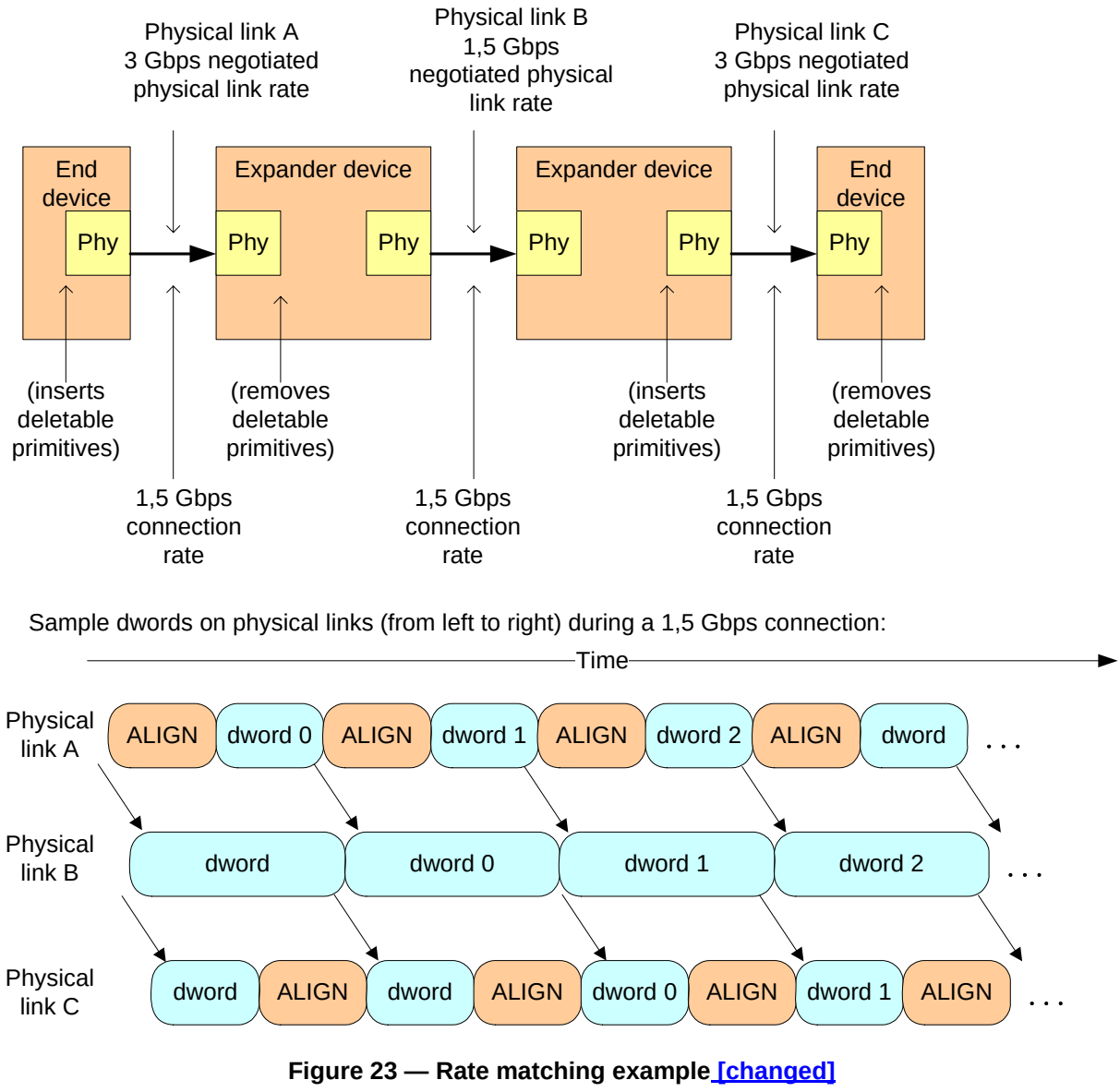


Figure 23 — Rate matching example [changed]

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 10.4.3.12) 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 phys) state machines

7.14.1 SL state machines overview

...

All the SL state machines shall begin after receiving an Enable Disable SAS Link (Enable) message from the SL_IRM state machines.

Editor's Note 10: change SL_IR to SL_IRM 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_IRM state machines (see 7.9.5).

...

The SL_CC state machine receives the following messages from the SL_IRM state machines (see 7.9.5):

...

7.15 XL (link layer for expander phys) state machine

7.15.1 XL state machine overview

...

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_IRM state machines (see 7.9.5).

The XL state machine receives the following messages from the SL_IRM state machines:

Editor's Note 11: change SL_IR to SL_IRM in figure 147, figure 148, and figure 149

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.

7.17.2 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 phy through which the STP connection is routed shall implement the SATA flow control protocol on each physical 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 dwords at 1,5 Gbps; ~~or~~
b) 28 dwords at 3~~,0~~ Gbps; and
c) 36 dwords at 6 Gbps.

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 4 - The receive buffer requirements are based on $(20 + (4 \times 2^{(n-1)}))$ where n is 1 for 1,5 Gbps ~~and~~, 2 for 3,0 Gbps, and 3 for 6 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see ATA/ATAPI-7 V3). The $(4 \times n)$ 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 the 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 the 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 17. 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 17 — 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

...

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 18 defines the request format.

Table 18 — 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 19 defines the response format.

Table 19 — DISCOVER response (part 1 of 2)

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

Table 19 — DISCOVER response (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
48	Reserved							
49								
50	Vendor specific							
51								
52	Reserved							
53								
54	Reserved				REQUESTED LOGICAL LINK RATE			
55	Reserved		HARDWARE MAXIMUM LOGICAL LINKS		REQUESTED LOGICAL LINKS		ATTACHED REQUESTED LOGICAL LINKS	
~~52~~ 56	(MSB)							
~~55~~ 59	CRC							(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 contains the number of dwords that follow, not including the CRC field (i.e., 13). A RESPONSE LENGTH field set to 00h indicates there are 12 additional dwords (i.e., 48 additional bytes) before the CRC field in the response frame.

The PHY IDENTIFIER field indicates the phy for which physical configuration link information is being returned.

The ATTACHED DEVICE TYPE field indicates the DEVICE TYPE value received during the link reset sequence and is defined in table 11.

Table 20 — 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 NEGOTIATED PHYSICAL LINK RATE field is defined in table 21 and indicates the physical link rate negotiated during the link reset sequence. The negotiated physical link rate may be less than the programmed minimum physical link rate or greater than the programmed maximum physical link rate if the programmed physical link rates have been changed since the last link reset sequence.

Table 21 — NEGOTIATED PHYSICAL LINK RATE **field**

Code	Name	Description
0h	UNKNOWN	Phy is enabled; unknown physical 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 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.
8h	G1	Phy is enabled; 1,5 Gbps physical link rate. This field shall be updated to this value after the speed negotiation sequence completes.
9h	G2	Phy is enabled; 3~~,0~~ Gbps physical link rate. This field shall be updated to this value after the speed negotiation sequence completes.
Ah	G3	Phy is enabled; 6 Gbps physical link rate. This field shall be updated to this value after the speed negotiation sequence completes.
All others	Reserved.	

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

Table 22 describes the ATTACHED SATA PORT SELECTOR bit and the ATTACHED SATA DEVICE bit.

Table 22 — 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 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 5 - 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 23. 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 24.

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 23. The default value shall be the value of the HARDWARE MAXIMUM PHYSICAL LINK RATE field.

Table 23 — 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
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 24.

Table 24 — 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
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 6 - 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 7 - 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 25.

Table 25 — 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).

The REQUESTED LOGICAL LINK RATE field indicates the value of the REQUESTED LOGICAL LINK RATE field set by the PHY CONTROL function and is defined in table 12 (see 7.xx).

Table 26 — REQUESTED LOGICAL LINK RATE field

Code	Description
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
All others	Reserved

The HARDWARE MAXIMUM LOGICAL LINKS field indicates the maximum value supported by the phy for the REQUESTED LOGICAL LINKS field in the IDENTIFY address frame and is defined in table 12 (see 7.xx). This value is not adjusted based on the current physical link rate.

The REQUESTED LOGICAL LINKS field indicates the value of the REQUESTED LOGICAL LINKS field transmitted during the link reset sequence and is defined in table 12 (see 7.xx).

The ATTACHED REQUESTED LOGICAL LINKS field indicates the value of the REQUESTED LOGICAL LINKS field received during the link reset sequence and is defined in table 12 (see 7.xx).

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 27 defines the request format.

Table 27 — 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 (09h)							
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				REQUESTED LOGICAL LINK RATE			
38	Reserved							
39								
40	(MSB) CRC							
43								(LSB)

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

The FUNCTION field shall be set to 91h.

The REQUEST LENGTH field contains the number of dwords that follow, not including the CRC field (i.e., 9). A REQUEST LENGTH field set to 00h indicates there are 9 additional dwords (i.e., 36 additional bytes) before the CRC field in the request frame.

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

Table 28 defines the PHY OPERATION field.

Table 28 — 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 28 — 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 29 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 29 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 29 — 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
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 REQUESTED LOGICAL LINK RATE field specifies the logical link rate the phy should attempt to enable via multiplexing and is defined in table 30. 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 30 — REQUESTED LOGICAL LINK RATE field

Code	Description
0h	Do not change current value
8h	1,5 Gbps
9h	3 Gbps
Ah	6 Gbps
All others	Reserved

The CRC field is defined in 10.4.3.1.

Table 31 defines the response format.

Table 31 — 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 32 defines the values for this field.

Table 32 — PHY TEST PATTERN PHYSICAL LINK RATE **field**

Code	Description
8h	1,5 Gbps
9h	3~~,0~~ Gbps
Ah	6 Gbps
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 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
K28.5	D01.4	D16.7	D07.3	NAK (RESERVED 2)
K28.5	D01.4	D24.0	D16.7	RRDY (NORMAL)

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

1st	2nd	3rd	4th	Assignment
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 (0)
K28.5	D02.0	D24.0	D07.3	BREAK
K28.5	D02.0	D29.7	D16.7	
K28.5	D02.0	D30.0	D27.4	CLOSE (NORMAL)
K28.5	D02.0	D31.4	D30.0	CLOSE (RESERVED 0)
K28.5	D04.7	D01.4	D24.0	BROADCAST (RESERVED 1)
K28.5	D04.7	D02.0	D01.4	BROADCAST (CHANGE)
K28.5	D04.7	D04.7	D04.7	BROADCAST (RESERVED 2)
K28.5	D04.7	D07.3	D29.7	BROADCAST (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	MUX (1)
K28.5	D07.3	D07.3	D07.3	
K28.5	D07.3	D24.0	D29.7	
K28.5	D07.3	D27.4	D16.7	
K28.5	D07.3	D29.7	D27.4	
K28.5	D07.3	D30.0	D24.0	
K28.5	D07.3	D31.4	D02.0	
K28.5	D10.2	D10.2	D27.3	ALIGN (0)
K28.5	D16.7	D01.4	D02.0	
K28.5	D16.7	D02.0	D07.3	
K28.5	D16.7	D04.7	D31.4	
K28.5	D16.7	D16.7	D16.7	OPEN_ACCEPT
K28.5	D16.7	D24.0	D27.4	MUX (2)
K28.5	D16.7	D27.4	D30.0	

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

1st	2nd	3rd	4th	Assignment
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	MUX (3)
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)
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)

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

1st	2nd	3rd	4th	Assignment
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)

Annex K Messsages between state machines

Editor's Note 12: Change SL_IR to SL_IRM throughout this annex