To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 27 October 2006
Subject: 06-275r0 SAS-2 ALIGN insertion rate during STP connections

Revision history

Revision 0 (27 October 2006) First revision

Related documents

sas2r06 - Serial Attached SCSI - 2 (SAS-2) revision 6

Overview

In some SAS-1.1 expander implementations, there was confusion about expander's obligations to pass through STP initiator phy throttling ALIGNs/NOTIFYs. These are the 1/128 extra ALIGNs, on top of the 1/2048 clock skew management ALIGNs and any rate matching ALIGNs, that STP initiator ports are required to transmit to ensure that an STP/SATA bridge receives enough extra ALIGNs to meet the SATA rule of inserting 2/256 ALIGNs to the SATA device. As the dword stream passes through SAS expanders, the clock skew management and rate matching ALIGNs will occur or not as needed; the STP/SATA bridge still must meet the SATA rule (the bridge must also insert its ALIGNs in pairs, so may have to buffer some data dwords while waiting for the SAS ALIGNs to show up and consume the time).

An expander is supposed to strip off all ALIGNs it receives and regenerate ALIGNs as needed (i.e., on underflows) while transmitting. When the incoming stream includes STP initiator phy throttling ALIGNs, they cause underflows and are expected to be result in an equivalent amount of ALIGNs on the transmitter side.

If the ALIGNs are inside a SATA frame, that rule is likely to be honored; if the expander underflows during data dwords, it has no choice but to insert ALIGNs.

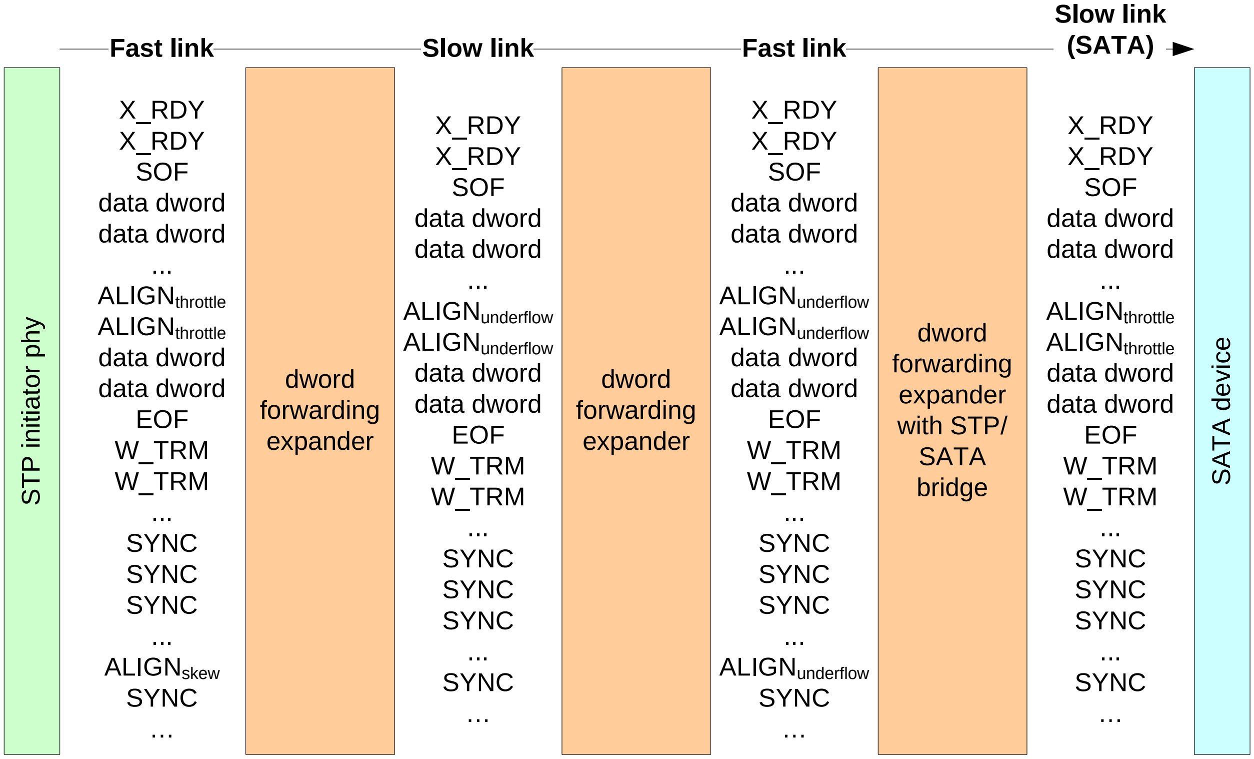

Fast link
Slow link
Fast link
Slow link (SATA)
STP initiator phy
dword forwarding expander
dword forwarding expander
dword forwarding expander with STP/ SATA bridge
SATA device
X_RDY X_RDY SOF data dword data dword ... ALIGN throttle ALIGN throttle data dword data dword EOF W_TRM W_TRM ... SYNC SYNC SYNC ... ALIGN skew SYNC …
X_RDY X_RDY SOF data dword data dword ... ALIGN underflow ALIGN underflow data dword data dword EOF W_TRM W_TRM ... SYNC SYNC SYNC ... SYNC …
X_RDY X_RDY SOF data dword data dword ... ALIGN underflow ALIGN underflow data dword data dword EOF W_TRM W_TRM ... SYNC SYNC SYNC ... ALIGN underflow SYNC …
X_RDY X_RDY SOF data dword data dword ... ALIGN throttle ALIGN throttle data dword data dword EOF W_TRM W_TRM ... SYNC SYNC SYNC ... SYNC …

Figure 1 — Throttling ALIGNs preserved during data frame

If the ALIGNs appear inside a repeated or continued primitive, however, an expander that detects it is in the state of receiving that primitive and places its transmitter in the state of transmitting that primitive could forget that it must preserve those throttling ALIGNS - not necessary 1 for 1, but the rate they occur must be the same. If it does not preserve them, it causes cause overflows in an STP/SATA bridge that expects them (to meet its SATA obligation of inserting 2/256 ALIGNs.

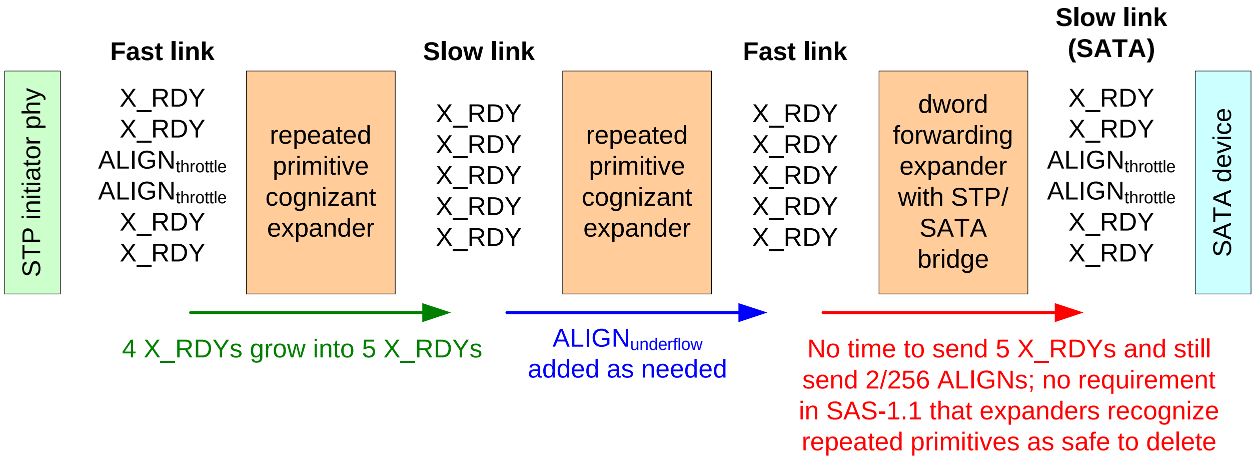

Fast link
Slow link
Fast link
Slow link (SATA)
STP initiator phy
X_RDY
X_RDY
ALIGNthrottle
ALIGNthrottle
X_RDY
X_RDY
repeated primitive cognizant expander
X_RDY
X_RDY
X_RDY
X_RDY
X_RDY
repeated primitive cognizant expander
X_RDY
X_RDY
X_RDY
X_RDY
X_RDY
dword forwarding expander with STP/ SATA bridge
X_RDY
X_RDY
ALIGNthrottle
ALIGNthrottle
X_RDY
X_RDY
SATA device
4 X_RDYs grow into 5 X_RDYs
ALIGNunderflow added as needed
No time to send 5 X_RDYs and still send 2/256 ALIGNs; no requirement in SAS-1.1 that expanders recognize repeated primitives as safe to delete

Figure 2 — Throttling ALIGNs lost during repeated primitive

This type of expander must make a special effort to transmit the throttling ALIGNs. It can simply insert 1/128 ALIGNs back into the data stream to meet the rule. It must not blindly insert 1/128 + 1/2048, however, or it will cause eventual overflow of its own buffers if it is receiving from a fast link (e.g., +100 ppm) and has a slow transmitter (e.g. -100 ppm) (this would only cause a problem if the link was transferring dwords other than repeated or continued primitives - losing a repeated or continued primitive is safe as long as at least one gets through and, for continued primitives, the CONT is not dropped).

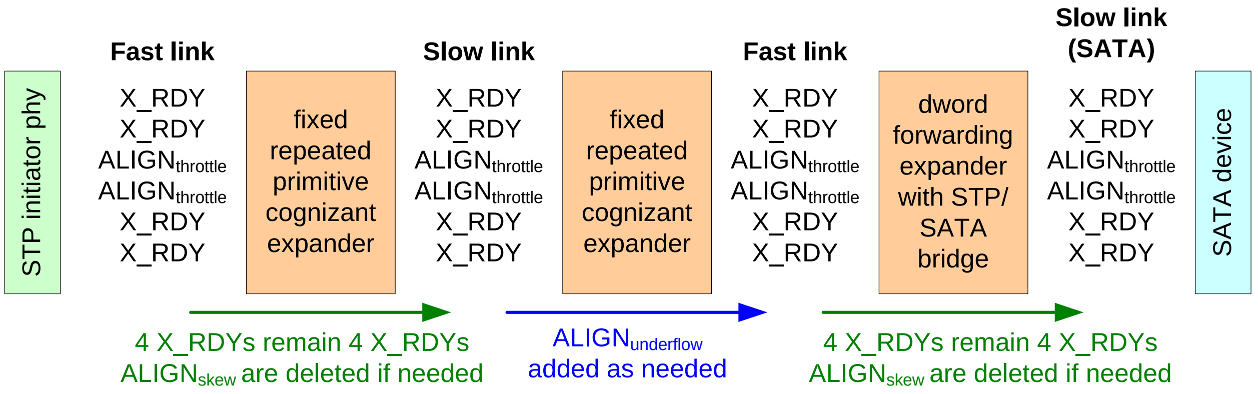


Figure 3 — Throttling ALIGN reconstituted during repeated primitives

SAS-2 needs to state clearly that the outgoing rate of ALIGNs must match the incoming rate of ALIGNs, except for those stripped out because of physical link rate tolerance differences and rate matching.

In SAS-2, the clock skew management (renamed “physical link rate tolerance”) ALIGN frequency has been increased to 1/64 for all protocols to support spread-spectrum clocking. This is more than SAS requires, so the special STP initiator phy throttling ALIGNs are no longer needed. So, this rule cannot be worded in terms like “preserve the rate of STP initiator phy throttling ALIGNs.”

Blindly inserting 1/64 ALIGNs will not work, because that would induce overflows of the expander’s own buffers if the dword stream is from a SAS-1.1 initiator or if it is a SAS-2 initiator that is faster than the expander itself. Blindly inserting 1/128 ALIGNs during a repeated or continued primitive should still work even if the dword stream is from a SAS-2 initiator; the SATA device does not care how many primitives it sees, so if the SAS-2 initiator sends more than the SATA device receives, nothing should break.

A few suggested changes to the STP flow control description are also included.

Suggested changes

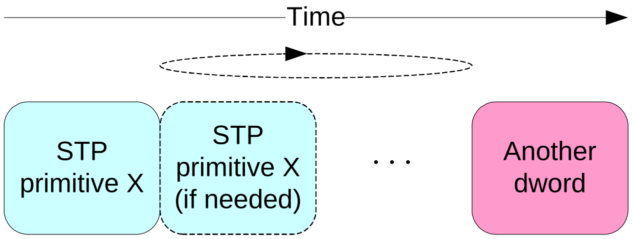
7.2 Primitives

7.2.4 Primitive sequences

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 may be sent inside repeated primitive sequences as described in 7.2.4.1.

Figure 4 shows an example of transmitting a repeated primitive sequence.

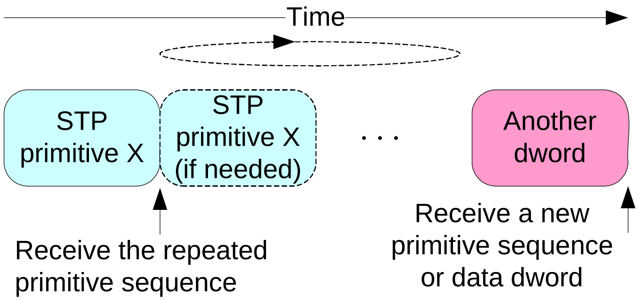


NOTE: Another dword is a dword other than ALIGN, NOTIFY, or STP primitive X

Figure 4 — Transmitting a repeated primitive sequence

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). An expander device forwarding a repeated primitive sequence may transmit more repeated primitives than it receives (i.e., expand) or transmit fewer repeated primitives than it receives (i.e, contract).

Figure 5 shows an example of receiving a repeated primitive sequence.


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

NOTE: Another dword is a dword other than ALIGN, NOTIFY, or STP primitive X

Figure 5 — Receiving a repeated primitive sequence

7.2.4.4 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be transmitted as specified in Figure 7.17.3. ALIGNs and NOTIFYs may be sent inside continued primitive sequences as described in 7.2.4.1.

7.3 Physical link rate tolerance management

7.3.1 Physical link rate tolerance management overview

The internal clock for a device is typically based on a PLL with its own clock generator and is used when transmitting dwords on the physical link. When receiving, however, dwords need to be latched based on a clock derived from the input bit stream itself. Although the input clock is nominally a fixed frequency, it may differ slightly from the internal clock frequency up to the physical link rate tolerance defined in table 51 (see 5.3.3). Over time, if the input clock is faster than the internal clock, the phy receiver 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 phy receiver may not have a dword when needed in an internal buffer; this is called an underrun.

To solve this problem, phy transmitters insert ALIGNs or NOTIFYs in the dword stream. Phy receivers may pass ALIGNs and NOTIFYs through to their internal buffers, or may strip them out when an overrun occurs.

Phy receivers add ALIGNs or NOTIFYs when an underrun occurs. The internal logic shall ignore all ALIGNs and NOTIFYs that arrive in the internal buffers.

Elasticity buffer circuitry, as shown in figure 141, 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 physical link rate tolerance management ALIGN insertion requirements in table 101.

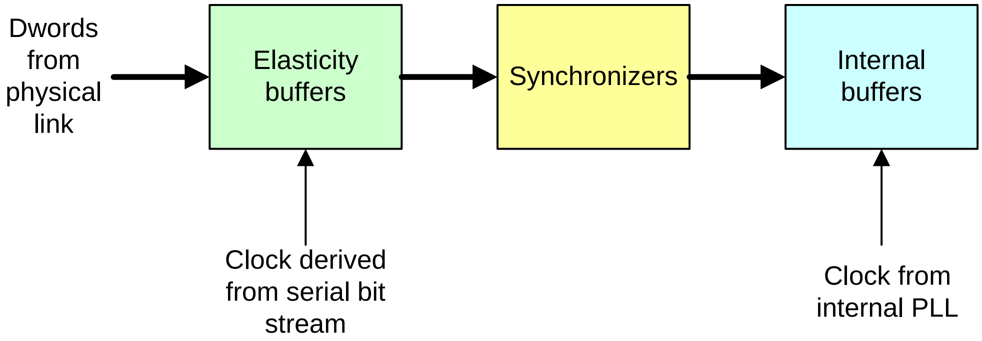


Figure 141 — Elasticity buffers

7.3.2 Phys originating dwords

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 for physical link rate tolerance management as described in table 1.

Table 1 — Physical link rate tolerance management ALIGN insertion requirement

Physical link rate	Requirement
1,5 Gbps	One ALIGN or NOTIFY within every 128 dwords [a]
3 Gbps	Two ALIGNs or NOTIFYs within every 256 dwords [b]
6 Gbps	~~Two~~Four ALIGNs or NOTIFYs within every 512 dwords

[a] Phys compliant with previous versions of this standard were required to insert one ALIGN or NOTIFY within every 2 048 dwords at 1,5 Gbps.

[b] Phys compliant with previous versions of this standard were required to insert two ALIGNs or NOTIFYs within every 4 096 dwords at 3 Gbps.

NOTE 1 - These numbers account for the worst case clock frequency differences between the fastest phy transmitter and the slowest phy receiver (e.g., a center-spreading expander phy originating dwords in an STP connection at +2 400 ppm that are forwarded to a down-spreading SATA device with an internal clock at

-5 350 ppm). The difference of 7 750 ppm (i.e., 0,775 % or 1/129) is less than the ALIGN insertion rate of 1/128 (i.e.,7 813 ppm or 0,78125 %), ensuring there are enough deletable primitives for the phy receiver to delete without having to buffer dwords.

ALIGNs and NOTIFYs inserted for physical link rate tolerance management are in addition to ALIGNs and NOTIFYs inserted for rate matching (see 7.13). See Annex H for a summary of their combined requirements.

See 7.2.5.2 for details on rotating through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3). NOTIFYs may also be used in place of ALIGNs (see 7.2.5.10) on SAS physical links.

7.3.3 Expander phys forwarding dwords

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 as required to match the transmit and receive connection rates. It is not required to transmit the number of ALIGNs and/or NOTIFYs for physical link rate tolerance management described in table 1 when forwarding to a SAS physical link. It ~~may~~shall 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 (e.g., if receiving at -100 ppm and transmitting at +100 ppm, it transmits fewer ALIGNs and/or NOTIFYs that it receives).

The expander device is also required to insert ALIGNs and NOTIFYs for rate matching (see 7.xx). During an STP connection, the expander device shall:

a) preserve the incoming rate of any additional ALIGNs and NOTIFYs that it receives that are not discarded because of physical link rate tolerance management or rate matching (e.g. the 1/128 ALIGNs and/or NOTIFYs received from an originating STP initiator phy compliant with previous versions of this standard for STP initiator phy throttling); or

b) transmit one ALIGN or NOTIFY within every 128 dwords,

without discarding any data dwords or primitives. It may reduce the length of repeated primitive sequences (i.e., primitive, CONT, and data dword sequences).

NOTE 2 - One possible implementation for expander devices forwarding dwords is for the expander device to delete all ALIGNs and NOTIFYs received and to insert ALIGNs and/or NOTIFYs 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 as required to match the transmit and receive connection rates. It is not required to transmit any particular number of ALIGNs and/or NOTIFYs for physical link rate tolerance management when forwarding to a SAS physical link and is not required to ensure that any ALIGNs and/or NOTIFYs it transmits are in pairs.

NOTE 3 - Due to physical link rate tolerance management ALIGN and NOTIFY removal, the STP target port may not receive a pair of ALIGNs and/or NOTIFYs every 256 dwords, even if the STP initiator port transmitted them in pairs. However, the rate of the dword stream allows for ALIGN or NOTIFY insertion by the STP/SATA bridge. One possible implementation is for the STP/SATA bridge to delete all ALIGNs and NOTIFYs received by the STP target port and to insert two consecutive ALIGNs at the SATA host port when its elasticity buffer is empty or when 254 non-ALIGN dwords have been transmitted. It may need to buffer up to 2 dwords concurrently being received by the STP target port while it does so.

7.15 XL state machine

7.15.2 XL transmitter and receiver

...

The XL transmitter shall ensure physical link rate tolerance management requirements are met (see 7.3) while originating dwords.

The XL transmitter shall ensure physical link rate tolerance management requirements are met while forwarding dwords (i.e., during a connection) by inserting or deleting as many ALIGNs and/or NOTIFYs as required to match the transmit and receive connection rates (see 7.3.2).

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

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

NOTE 4 - The XL transmitter may always insert an ALIGN or NOTIFY before transmitting a BREAK, BREAK_REPLY, CLOSE, or SATA_HOLDA to meet physical link rate tolerance management requirements.

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

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

NOTE 5 - This ensures that physical link rate tolerance management requirements are met, even if the forwarded dword stream does not include an ALIGN or NOTIFY until the last possible dword.

The XL transmitter shall ensure rate matching requirements are met during a connection (see 7.13).

...

7.15.9 XL6:Open_Response_Wait state

7.15.9.1 State description

This state waits for a response to a transmitted OPEN address frame and determines the appropriate action to take based on the response.

This state shall either:

a) request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter, honoring ALIGN insertion rules for rate matching and physical link rate ~~tolernace~~tolerance management; or
b) send Transmit Dword messages to the XL transmitter to transmit all dwords received with Forward Dword indications. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence.

Editor's Note 1: add Forward Dword going into XL6 in the state machine figure, where it is missing

...

7.15.10 XL7:Connected state

7.15.10.1 State description

This state provides a full-duplex circuit between two phys within an expander device.

This state shall send Transmit Dword messages to the XL transmitter to transmit all dwords received with Forward Dword indications. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence..

If this state has not sent a Forward Close request to the ECR, this state shall send Forward Dword requests to the ECR containing each valid dword except BREAK and CLOSE primitives received with Dword Received messages. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence.

If:

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

the expander phy shall:

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

If:

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

the expander phy shall:

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

If a CLOSE Received message is received, this state shall send a Forward Close request to the ECR with the argument from the CLOSE Received message.

If a BREAK Received message is received, this state shall send a Forward Break request to the ECR (see 7.15.10.3).

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

7.15.11 XL8:Close_Wait state

7.15.11.1 State description

This state closes a connection and releases path resources.

Upon entry into this state, this state shall send a Transmit CLOSE message to the XL transmitter with the argument from the Forward Close indication, then shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

> NOTE 49 - Possible livelock scenarios occur if the BREAK_REPLY method of responding to received BREAK primitive sequences is disabled and a phy transmits BREAK to break a connection (e.g., if its Close Timeout timer expires). Phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

If a Dword Received message is received containing a valid dword except a BREAK or CLOSE primitive, this state shall send Forward Dword requests to the ECR containing that dword. During an STP connection, the expander device may expand or contract a repeated or continued primitive sequence.

If:

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

the expander phy shall:

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

If:

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

the expander phy shall:

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

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

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

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

7.17 STP link layer

7.17.1 STP frame transmission and reception

...

STP encapsulates SATA with connection management. Table 129 summarizes STP link layer differences from the SATA link layer (see ATA/ATAPI-7 V3) that affect behavior during an STP connection.

Table 129 — STP link layer differences from SATA link layer during an STP connection

Feature	Description	Reference
STP flow control	Flow control through an STP connection is point-to-point, not end-to-end. Expander devices accept dwords in ~~a temporary holding~~into an STP flow control buffer after transmitting SATA_HOLD to avoid losing data en-route before the transmitting phy acknowledges the SATA_HOLD with SATA_HOLDA.	7.17.3
Continued primitive sequence	Sustain the continued primitive sequence if a SATA_CONT appears after the continued primitive sequence has begun.	7.17.3

7.17.2 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 or expander phy during an STP connection is receiving a SATA frame and its STP flow control buffer begins to fill up, it shall transmit SATA_HOLD. After transmitting SATA_HOLD, it shall accept at least the following number of data dwords for the SATA frame into the STP flow control buffer:

a) 24 data dwords at 1,5 Gbps; or
b) 28 data dwords at 3 Gbps~~.~~,

and shall expect to receive SATA_HOLDA within that number of data dwords. While receiving SATA_HOLDA, it does not place any data dwords into the STP flow control buffer. When the STP flow control buffer empties enough to hold at least that number of data dwords, it shall stop transmitting SATA_HOLD.

When an STP phy or expander phy in an STP connection is transmitting a SATA frame and receives SATA_HOLD, it shall transmit no more than 20 data dwords for the SATA frame and respond with SATA_HOLDA.

NOTE 50 - The ~~receive~~STP flow control buffer requirements are based on $(20 + (4 \times 2^n))$ where n is 0 for 1,5 Gbps and 1 for 3 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see ATA/ATAPI-7 V3). The (4×2^n) portion of this equation is based on:

a) One-way propagation time on a 10 m cable = (5 ns/m propagation delay) × (10 m cable) = 50 ns;
b) Round-trip propagation time on a 10 m cable = 100 ns (e.g., time to send SATA_HOLD and receive SATA_HOLDA);
c) Time to transmit a 1,5 Gbps dword = (0,667 ns/bit unit interval) × (40 bits/dword) = 26,667 ns; and
d) Number of 1,5 Gbps dwords on the wire during round-trip propagation time = (100 ns / 26,667 ns) = 3,75.

Receivers may support longer cables by providing larger STP flow control buffer sizes.

When a SATA host phy in an STP/SATA bridge is receiving a SATA frame from a SATA physical link, it shall transmit ~~a~~ SATA_HOLD when it is only capable of receiving 21 more data dwords. It shall stop transmitting SATA_HOLD (e.g., return to transmitting SATA_R_IP) when it is capable of receiving at least 21 more data dwords.

NOTE 51 - SATA requires that frame transmission cease and SATA_HOLDA be transmitted within 20 data dwords of receiving SATA_HOLD. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

When a SATA host phy in an STP/SATA bridge is transmitting a SATA frame to a SATA physical link, it shall transmit no more than 19 data dwords after receiving SATA_HOLD.

NOTE 52 - SATA assumes that once a SATA_HOLD is transmitted, frame transmission ceases and SATA_HOLDA arrives within 20 dwords. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

Figure 168 shows STP flow control between:

a) an STP initiator phy receiving a frame;
b) an expander device (the first expander device);
c) an expander device with an STP/SATA bridge (the second expander device); and
d) a SATA device phy transmitting a frame.

...

Figure 168 — STP flow control

After the STP initiator phy transmits SATA_HOLD, it receives a SATA_HOLDA reply from the first expander device within 24 dwords. The first expander device transmits SATA_HOLD to the second expander device and receives SATA_HOLDA within 24 dwords, buffering data dwords in the STP flow control buffer that it is no longer able to forward to the STP initiator phy. The second expander device transmits SATA_HOLD to the SATA device phy and receives SATA_HOLDA within 21 dwords, buffering data dwords in the STP flow control buffer that it is no longer able to forward to the first expander device. When the SATA device phy stops transmitting data dwords, its previous data dwords are stored in the STP flow control buffers in both expander devices and the STP initiator phy.

After the STP initiator phy drains its buffer and transmits SATA_R_IP, it receives data dwords from the first expander device's STP flow control buffer, followed by data dwords from the second expander device's STP flow control buffer, followed by data dwords from the SATA device phy.

7.17.3 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be:

1) transmitted two times;
2) ~~then~~ be followed by SATA_CONT, if needed;
3) ~~then~~ be followed by vendor-specific scrambled data dwords, if needed.

ALIGNs and NOTIFYs may be sent inside continued primitive sequences as described in 7.2.4.1.

After the SATA_CONT, during the vendor-specific scrambled data dwords:

a) a SATA_CONT continues the continued primitive sequence; and
b) any other STP primitive, including the primitive that is being continued, ends the continued primitive sequence.

Figure 169 shows an example of transmitting a continued primitive sequence.

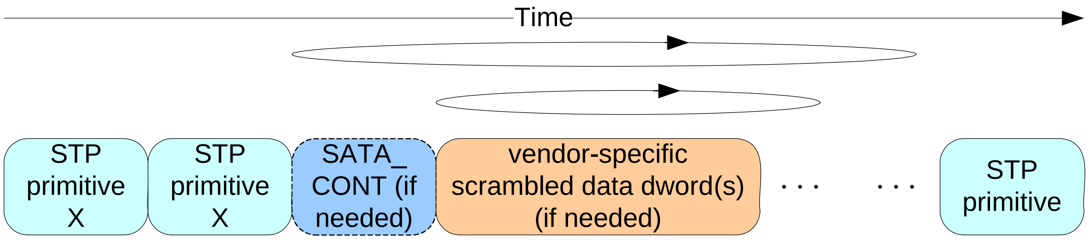


Figure 169 — Transmitting a continued primitive sequence

Receivers shall detect a continued primitive sequence after at least one primitive is received. The primitive may be followed by one or more of the same primitive. The primitive may be followed by one or more SATA_CONTs, each of which may be followed by vendor-specific data dwords. Receivers shall ignore invalid dwords before, during, or after the SATA_CONT(s). Receivers do not count the number of times the continued primitive, the SATA_CONTs, or the vendor-specific data dwords are received (i.e., receivers are simply in the state of receiving the primitive).

Expanders forwarding dwords may or may not detect an incoming sequence of the same primitive and convert it into a continued primitive sequence.

Figure 170 shows an example of receiving a continued primitive sequence.

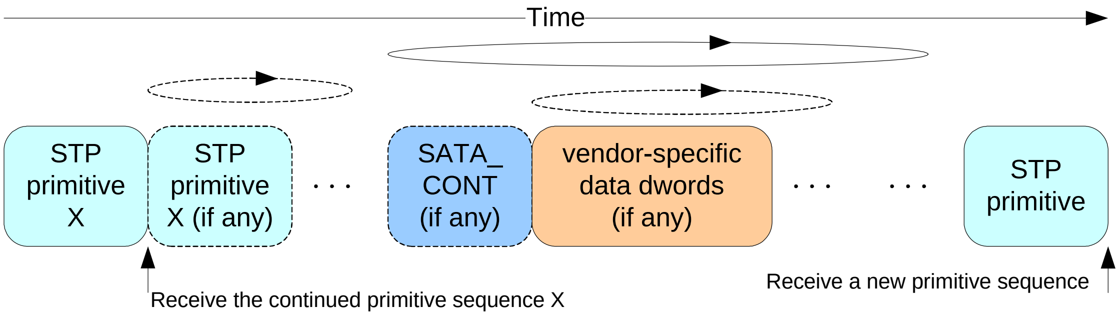

Time
STP primitive X
STP primitive X (if any)
. . .
SATA_ CONT (if any)
vendor-specific data dwords (if any)
. . .
. . .
STP primitive
Receive the continued primitive sequence X
Receive a new primitive sequence

Figure 170 — Receiving a continued primitive sequence

An expander device forwarding a continued primitive sequence may transmit more dwords in the continued primitive sequence than it receives (i.e., expand) or transmit fewer dwords in the continued primitive sequence than it receives (i.e, contract).