To: T10 Technical Committee
From: Robert Sheffield, Intel (robert.l.sheffield@intel.com)
Date: 26 May 2005
Subject: 05-224r0 SAS-1.1: Clarify STP exceptions to SATA link protocol

Revision history

Revision 0 (26 May 2005) First revision

Related documents

sas1.1-r09 - Serial Attached SCSI 1.1 revision 9

Overview

The rules requiring a SAS phy in an STP connection to handle receipt of a SATA_CONT after a continued primitive sequence has already begun are misplaced in the standard and it is not clearly identified that it is an exception to the general stipulation that frame transmission during an STP connection is governed by ATA/ATAPI-7.

This proposal is to move subclause 7.2.2.4 to subclause under 7.17 STP link layer, and to add text to 7.17.1 to clearly identify which elements of STP protocol represent exceptions to the ATA/ATAPI-7 link layer requirements that apply to SAS ports during an STP connection.

Suggested changes

7.2.4 Primitive sequences

7.2.4.1 Primitive sequences overview

Table 1 summarizes the types of primitive sequences.

Table 1 — Primitive sequences

Primitive sequence type	Number of times the transmitter transmits the primitive to transmit the primitive sequence	Number of times the receiver receives the primitive to detect the primitive sequence
Single	1	1
Repeated	1 or more	1
Continued	2 followed by SATA_CONT	1
Triple	3	3
Redundant	6	3

Any number of ALIGNs and NOTIFYs may be sent inside primitive sequences without affecting the count or breaking the consecutiveness requirements. Rate matching ALIGNs and NOTIFYs shall be sent inside primitive sequences inside of connections if rate matching is enabled (see 7.13).

7.2.4.2 Single primitive sequence

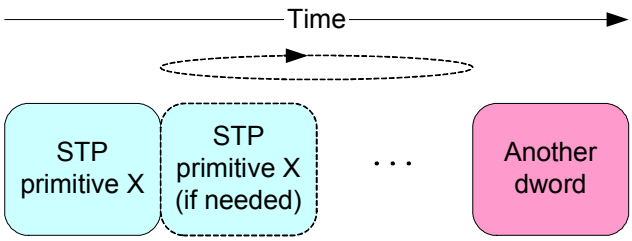
Primitives labeled as single primitive sequences (e.g., RRDY, SATA_SOF) shall be transmitted one time to form a single primitive sequence.

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

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 123 shows an example of transmitting a repeated primitive sequence.



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

Figure 123 — 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).

Figure 124 shows an example of receiving a repeated primitive sequence.

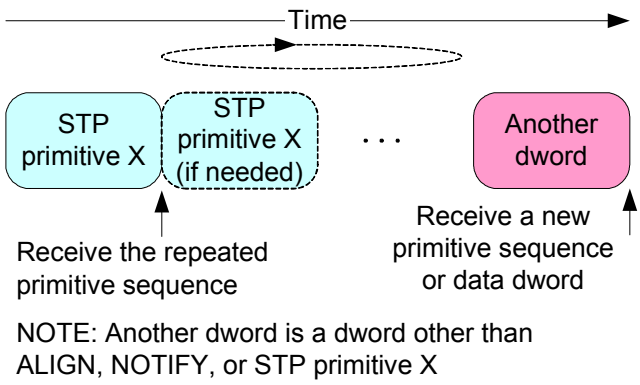


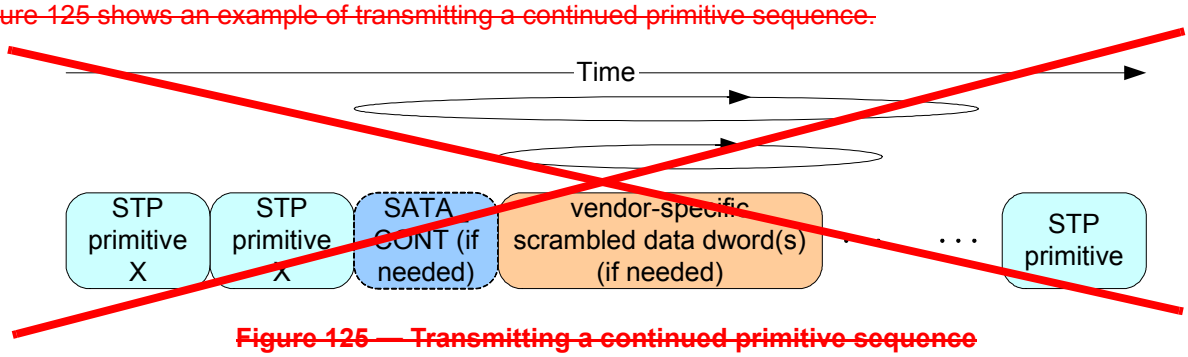
Figure 124 — Receiving a repeated primitive sequence

7.2.4.4 Continued primitive sequence

Primitives that form continued primitive sequences (e.g., SATA_HOLD) shall be transmitted two times, then be followed by SATA_CONT, if needed, 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. Specific requirements for SATA_CONT are described in 7.17.2. ~~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 125 shows an example of transmitting a continued primitive sequence.~~

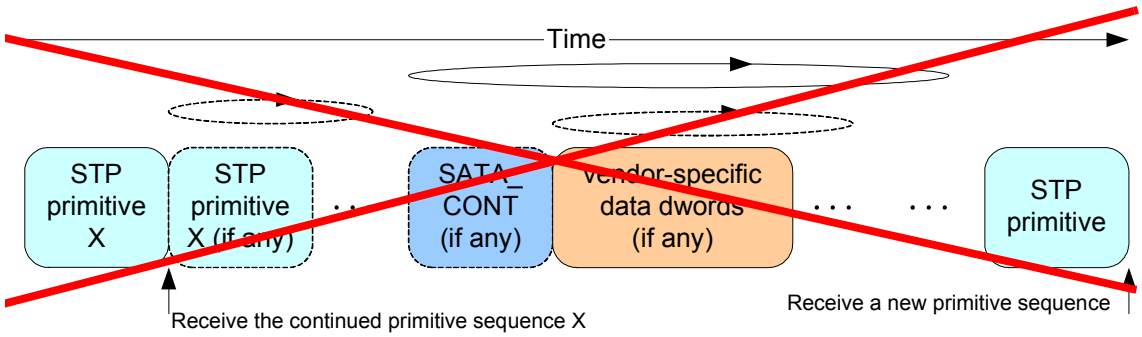


~~Figure 125 — Transmitting a continued primitive sequence~~

~~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 126 shows an example of receiving a continued primitive sequence.~~



~~**Figure 126 — Receiving a continued primitive sequence**~~

The remainder of subclause 7.2.4 remains unchanged.

7.17 STP link layer

7.17.1 STP frame transmission and reception

STP frame transmission is defined by SATA (see ATA/ATAPI-7 V3) except as described in subclauses 7.17.2, 7.17.3, and 7.17.4 (see table 104). During an STP connection, frames are preceded by SATA_SOF and followed by SATA_EOF as shown in figure 151.

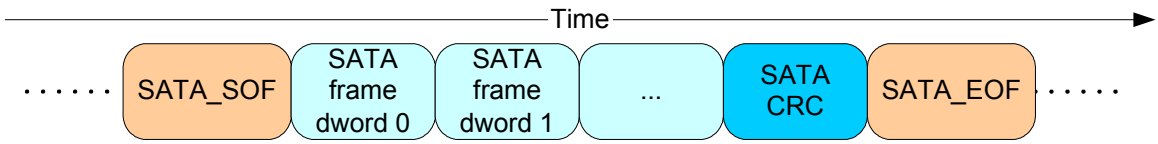


Figure 151 — STP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5).

Other primitives may be interspersed during the connection as defined by SATA.

STP encapsulates SATA with connection management.

Table 104 lists elements of STP protocol that are exceptions or additional qualifications to the link layer protocol defined by ATA/ATAPI-7 for transmission and reception of SATA FISes during the context of an STP connection.

Table 104 — Exceptions to ATA/ATAPI-7 link layer protocol during an STP connection

STP prototol element	Description	Reference
Continued primitive sequence	Sustain the continued primitive sequence if a SATA_CONT appears after the continued primitive sequence has begun.	7.17.2
STP initiator phy throttling	Limit the number of dwords transmitted to make room for more ALIGN primitives which may be deleted en-route from the STP initiator to the STP/SATA bridge in elasticity buffers as the dword stream crosses different clock domains.	7.17.3
STP flow control	Flow control through an STP connection is point-to-point, not end-to-end. Expander devices accept up to 28 dwords in a temporary holding buffer after transmitting SATA HOLD to avoid losing data en-route before the transmitting device acklowledges the SATA_HOLD with a SATA_HOLDA.	7.17.4

7.17.2 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 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 152 shows an example of transmitting a continued primitive sequence.

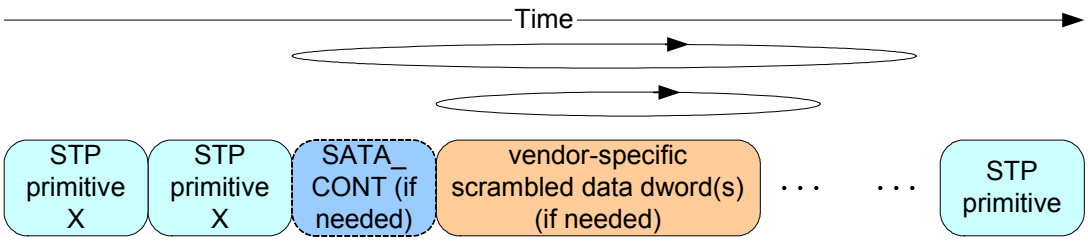


Figure 152 — Transmitting a continued primitive sequence

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 153 shows an example of receiving a continued primitive sequence.

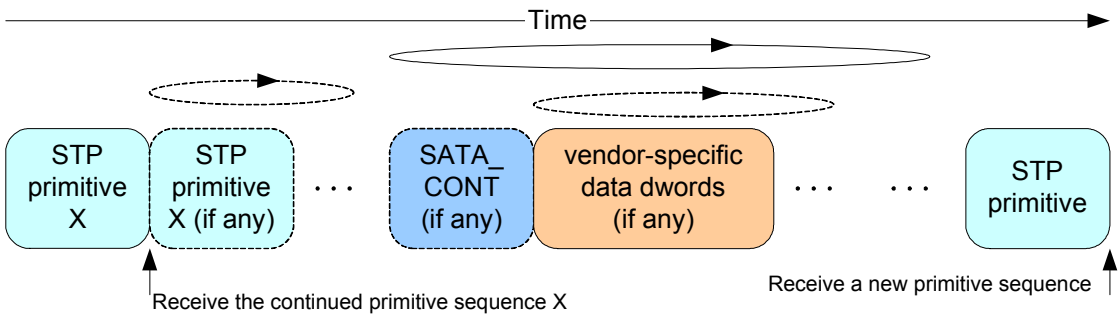


Figure 153 — Receiving a continued primitive sequence

7.17.3 STP initiator phy throttling

On a SATA physical link, phys are required to transmit two consecutive ALIGN (0)s 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 within every 256 dwords (i.e., within every overlapping window of 256 dwords) that are not ALIGNs or NOTIFYs 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 are indistinguishable from clock skew management ALIGNs and NOTIFYs.

STP target phys are not required to insert extra ALIGNs and/or NOTIFYs, 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.

ALIGNs and NOTIFYs inserted for STP initiator phy throttling are in addition to ALIGNs and NOTIFYs 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 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 for STP initiator phy throttling after:

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

7.17.4 STP flow control

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

When an STP port is receiving a frame and its buffer begins to fill up, it shall transmit SATA_HOLD. After transmitting SATA_HOLD, it shall accept the following number of data dwords for the frame:

a) 24 dwords at 1,5 Gbps; or
b) 28 dwords at 3,0 Gbps.

When an STP port is transmitting a frame and receives SATA_HOLD, it shall transmit no more than 20 data dwords for the frame and respond with SATA_HOLDA.

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

a) One-way propagation time on a 10 m cable = (5 ns/m propagation delay) $\times$ (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) $\times$ (40 bits/dword) = 26,667 ns; and
d) Number of 1,5 Gbps dwords on the wire during round-trip propagation time = (100 ns / 26,667 ns) = 3,75.

Receivers may support longer cables by providing larger buffer sizes.

When a SATA host port in an STP/SATA bridge is receiving a frame from a SATA physical link, it shall transmit a SATA_HOLD when it is only capable of receiving 21 more dwords.

NOTE 2 - SATA requires that frame transmission cease and SATA_HOLDA be transmitted within 20 dwords of receiving SATA_HOLD. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

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

NOTE 3 - 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 154 shows STP flow control between:

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

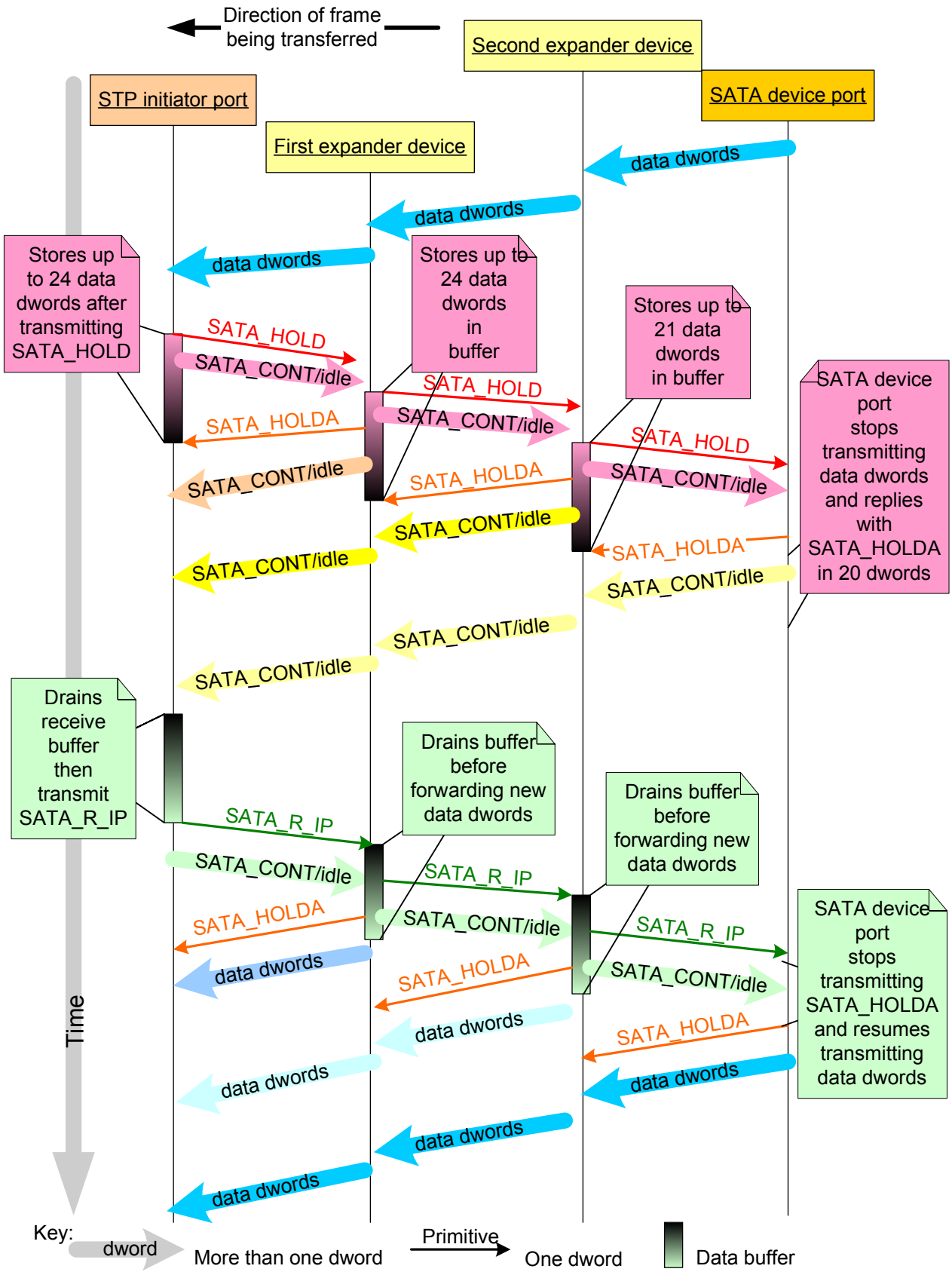


Figure 154 — STP flow control

After the STP initiator port transmits SATA_HOLD, it receives a SATA_HOLDA reply from the first expander device within 24 dwords (for a 1,5 Gbps physical link). The first expander device transmits SATA_HOLD to the second expander device and receives SATA_HOLDA within 24 dwords (for a 1,5 Gbps physical link), buffering data dwords it is no longer able to forward to the STP initiator port. The second expander device transmits SATA_HOLD to the SATA device port and receives SATA_HOLDA within 21 dwords (for a SATA physical link), buffering data dwords it is no longer able to forward to the first expander device. When the SATA device port stops transmitting data dwords, its previous data dwords are stored in the buffers in both expander devices and the STP initiator port.

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

The remainder of subclause 7.17 remains unchanged except for updating the subclause numbers:

~~7.17.4~~ 7.17.5 **Affiliations**

~~7.17.5~~ 7.17.6 **Opening an STP connection**

~~7.17.6~~ 7.17.7 **Closing an STP connection**

~~7.17.7~~ 7.17.8 **STP connection management examples**

~~7.17.8~~ 7.17.9 **STP (link layer for STP phys) state machines**

~~7.17.9~~ 7.17.10 **SMP target port support**