September 8, 2002

To:	T10 Technical Committee
From:	Timothy Hoglund (LSI Logic)
Subject:	SAS Expander update

This proposal is an update to SAS expander definition including a refinement of the Expander device model (section 4.8) and an update to the XL state machine (section 7.12.3).

Summary changes

1) updated expander device model
2) move PPT timers from Expander Function to XL state machine
3) expander interface signal descriptions
4) updates to XL state machine

To be resolved

1) where should 4.8.4.4.4 Pathway Recovery be incorporated?
2) XL interactions with internal ports
3) SAS address determination (IR for external ports, ?? for internal ports)

4.8 Expander device model

4.8.2 Expander device overview

Figure 1 shows the model of an expander device.

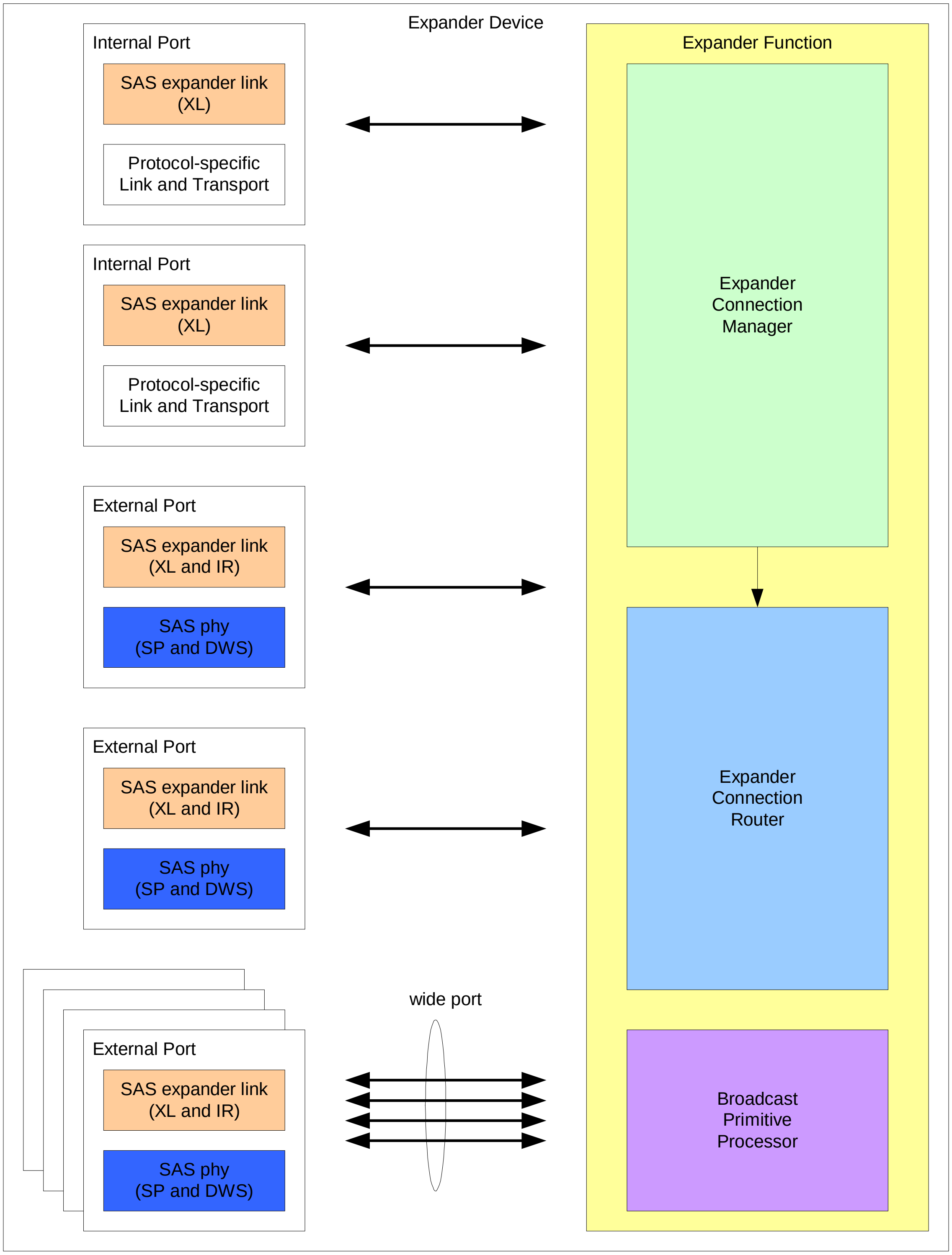
Expander Device
Internal Port
SAS expander link (XL)
Protocol-specific Link and Transport
Internal Port
SAS expander link (XL)
Protocol-specific Link and Transport
External Port
SAS expander link (XL and IR)
SAS phy (SP and DWS)
External Port
SAS expander link (XL and IR)
SAS phy (SP and DWS)
External Port
SAS expander link (XL and IR)
SAS phy (SP and DWS)
wide port
Expander Function
Expander Connection Manager
Expander Connection Router
Broadcast Primitive Processor

Figure 1. Expander device model

An expander device shall contain the following components:

a) two or more external expander ports;
b) expander connection manager;
c) expander connection router;
d) broadcast primitive processor; and
e) SMP target port.

An expander device may contain the following components:

a) SMP initiator port;
b) SSP target port;
c) SSP initiator port;
d) STP target port;
e) STP initiator port.

4.8.3 Expander ports

An external expander port consists of one or more expander link / SAS phy pairs according to SAS wide port rules. Each expander link within an expander port requests and responds to connection requests independently.

An internal expander port consists of an expander link and a protocol-specific transport.

4.8.4 Expander connection manager

The expander connection manager performs the following functions:

a) destination SAS address to destination link mapping using direct, subtractive, or table routed addressing methods;
b) arbitrate and assign or deny path resources for expander link connection requests following SAS arbitration and pathway recovery rules; and
c) expander connection router configuration.

4.8.5 Expander connection router

The expander connection router routes signals between pairs of expander links as configured by the expander connection manager. Sufficient routing resources shall be provided to support at least one active connection.

4.8.6 Broadcast primitive processor

The broadcast primitive processor receives broadcast primitive requests from each expander link and requests transmission of those requests on all expander links other than by expander links within the same expander port that the broadcast primitive request was received.

4.8.7 Expander interface overview

Figure 2 shows the interface requests, confirmations, indications, and responses used by an expander device to manage connections.

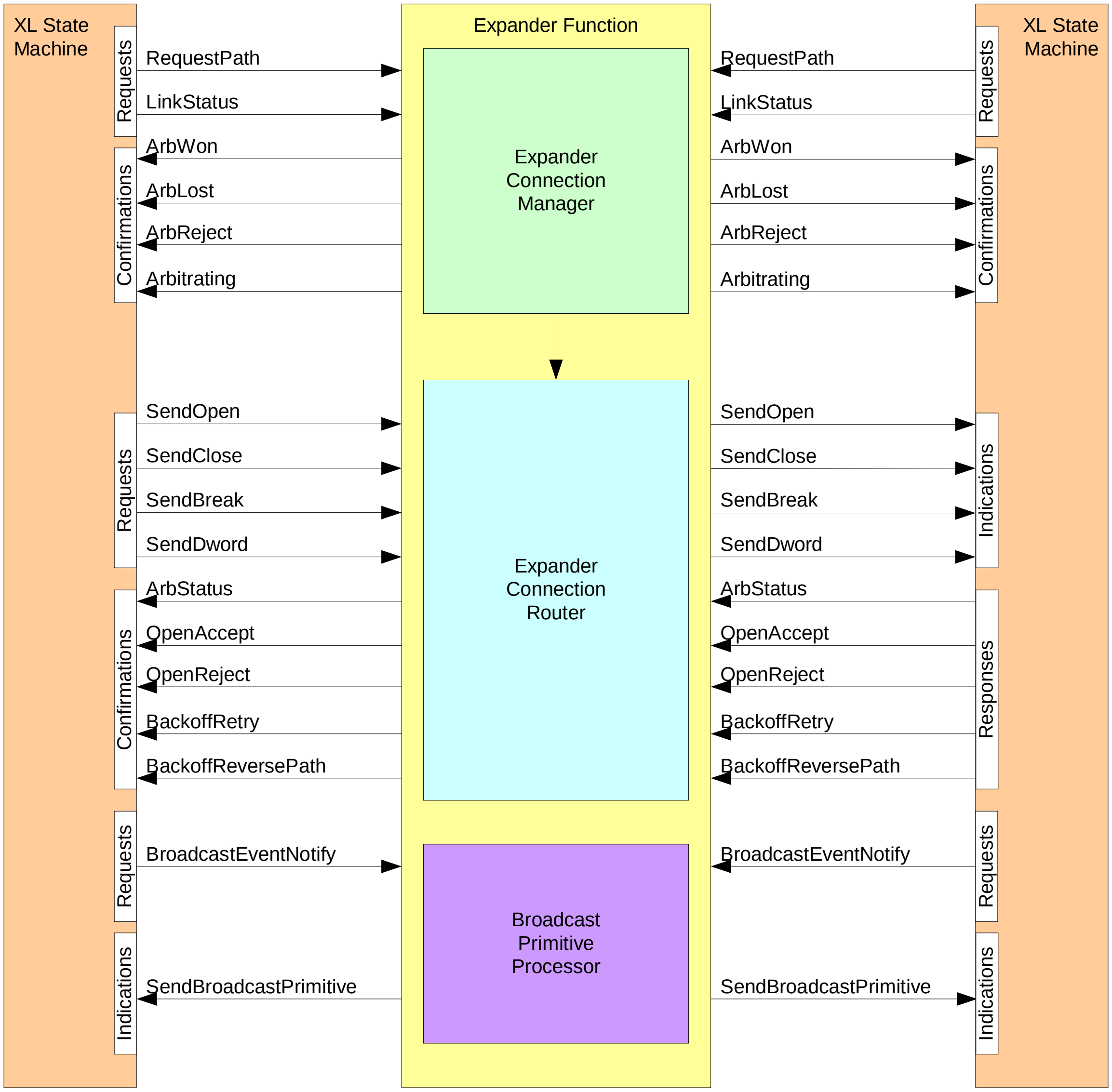

Figure 2. Expander interfaces

4.8.7.1 Expander connection manager interface

Table 1 summarizes the interface between expander links and the expander connection manager.

Primitive	Source	Destination	Description
RequestPath (arguments)	XL state machine	Connection Manager	Request for a connection.
LinkStatus (Partial Pathway)	XL state machine	Connection Manager	Request specifying that an expander link contains a partial pathway.
LinkStatus (Blocked Partial Pathway)	XL state machine	Connection Manager	Request specifying that an expander link contains a partial pathway and the most recent AIP received or indicated is AIP(WAITING ON PARTIAL).
LinkStatus (Connected)	XL state machine	Connection Manager	Request specifying that an expander link contains an active connection.
ArbWon	Connection Manager	XL state machine	Confirmation that an expander link has won path arbitration.
ArbLost	Connection Manager	XL state machine	Confirmation that an expander link has lost path arbitration.
ArbReject (No Destination)	Connection Manager	XL state machine	Confirmation that the expander connection manager did not find an operational phy configured to match the requested destination SAS address.
ArbReject (Bad Destination)	Connection Manager	XL state machine	Confirmation that the expander connection manager has determined that the requested destination SAS address maps back to the requesting port.
ArbReject (Bad Link Rate)	Connection Manager	XL state machine	Confirmation that the expander connection manager has determined that at least one destination link matches the requested destination SAS address but no links within the destination port are configured to support the requested link rate.
ArbReject (Pathway Blocked)	Connection Manager	XL state machine	Confirmation that the expander connection manager has determined that the requesting expander link must back off according to SAS pathway recovery rules.
Arbitrating (Waiting On Partial)	Connection Manager	XL state machine	Confirmation that the expander connection manager has determined that: a) at least one destination link matches the requested destination SAS address; and b) all of the links within the destination port contain a LinkStatus of Partial Pathway.
Arbitrating (Blocked On Partial)	Connection Manager	XL state machine	Confirmation that the expander connection manager has determined that: a) at least one destination link matches the requested destination SAS address;

			b) all of the links within the destination port contain a LinkStatus of Blocked Partial Pathway.
Arbitrating (Waiting On Connection)	Connection Manager	XL state machine	Confirmation that the expander connection manager has determined that: a) at least one destination link matches the requested destination SAS address; b) no links within the destination port are available; and c) at least one of the links within the destination port contain a LinkStatus of Connected.

Table 1: Expander connection manager interface

4.8.7.2 Expander connection router interface

Table 2 summarizes the interface between expander links via the expander connection router.

Primitive	Source	Destination	Description
SendOpen (arguments)	XL state machine	XL state machine	Request/indication to send an OPEN address frame.
SendClose	XL state machine	XL state machine	Request/indication to send a CLOSE primitive sequence.
SendBreak	XL state machine	XL state machine	Request/indication to send a BREAK primitive sequence.
SendDword	XL state machine	XL state machine	Request/indication to send a dword.
ArbStatus (Normal)	XL state machine	XL state machine	Confirmation/response that AIP(NORMAL) has been received.
ArbStatus (Waiting On Partial)	XL state machine	XL state machine	Confirmation/response that AIP(WAITING ON PARTIAL) has been received.
ArbStatus (Waiting On Connection)	XL state machine	XL state machine	Confirmation/response that AIP(WAITING ON CONNECTION) has been received.
ArbStatus (Waiting On Device)	XL state machine	XL state machine	Confirmation/response that AIP(WAITING ON DEVICE) has been received.
OpenAccept	XL state machine	XL state machine	Confirmation/response that OPEN_ACCEPT has been received.
OpenReject	XL state machine	XL state machine	Confirmation/response that OPEN_REJECT has been received.
Backoff Retry	XL state machine	XL state machine	Confirmation/response that: a) a higher priority OPEN address frame has been received; and b) the source SAS address of the received

			OPEN address frame is not equal to the destination SAS address of the transmitted OPEN address frame.
Backoff Reverse Path	XL state machine	XL state machine	Confirmation/response that: a) a higher priority OPEN address frame has been received; and b) the source SAS address of the received OPEN address frame is equal to the destination SAS address of the transmitted OPEN address frame.

Table 2: Expander connection router interface

4.8.7.3 Broadcast primitive processor interface

Table 3 summarizes the interface between expander links and the broadcast primitive processor.

Primitive	Source	Destination	Description
Broadcast Event Notify (arguments)	XL state machine	Broadcast primitive processor	Request/indication to send a broadcast primitive sequence on all other ports.
Send Broadcast Primitive	Broadcast primitive processor	XL state machine	Request/indication to send a broadcast primitive sequence on an expander link.

Table 3: Broadcast primitive processor interface

7.12.3 SAS link layer for expander phys

7.12.3.1 SAS Expander Link state overview

The Expander Link (XL) state machine controls the flow of dwords on the link and requests and maintains connections with another Expander Link state machine as facilitated by the expander connection manager and expander connection router.

The XL state machine contains the following states:

a) XL0:Idle;
b) XL1:Request_Path;
c) XL2:Request_Open;
d) XL3:Open_Confirm_Wait;
~~e) XL4:Open_Backoff;~~
f) XL5:Open_Reject;
g) XL6:Forward_Open;
h) XL7:Open_Response_Wait;
i) XL8:Connected;
j) XL9:Close_Wait;
k) XL10:Break; and
l) XL11:Break_Wait.

The XL state machine sends the following requests to the expander connection manager:

a) RequestPath;
b) LinkStatus(Link Ready);
c) LinkStatus(Partial Pathway);
d) LinkStatus(Blocked Partial Pathway); and
e) LinkStatus(Connected).

The XL state machine receives the following confirmations from the expander connection manager:

a) ArbReject(No Destination);
b) ArbReject(Bad Destination);
c) ArbReject(Bad Link Rate);
d) ArbReject(Pathway Blocked);
e) Arbitrating(Waiting On Connection);
f) Arbitrating(Waiting On Partial);
g) Arbitrating(Blocked On Partial);
h) ArbWon; and
i) ArbLost.

A pair of XL state machines communicate with each other using the following requests/indications which are routed through the expander connection router:

a) SendOpen;
b) SendBreak;
c) SendClose; and
d) SendDword.

A pair of XL state machines communicate with each other using the following confirmations/responses which are routed through the expander connection router:

a) ArbStatus(Normal);
b) ArbStatus(Waiting On Partial);
c) ArbStatus(Waiting On Connection);
d) ArbStatus(Waiting On Device);
e) OpenAccept;
f) OpenReject;
g) BackoffRetry; and
h) BackoffReversePath.

The XL state machine sends the following requests to the broadcast primitive processor:

a) BroadcastEventNotify.

The XL state machine receives the following requests from the broadcast primitive processor:

a) SendBroadcastPrimitive.

The XL state machine receives the following parameters from the IR state machine:

a) EnableSASLink.

The XL state machine shall be activated after the completion of the phy reset sequence by receiving an EnableSASLink parameter from the IR state machine.

Unless otherwise stated within a state description, all disparity errors, illegal characters, and unexpected primitives received within any XL state shall be ignored.

7.12.3.2 SAS Expander Link state diagram

Figure 3 defines the expander link layer connection state machine.

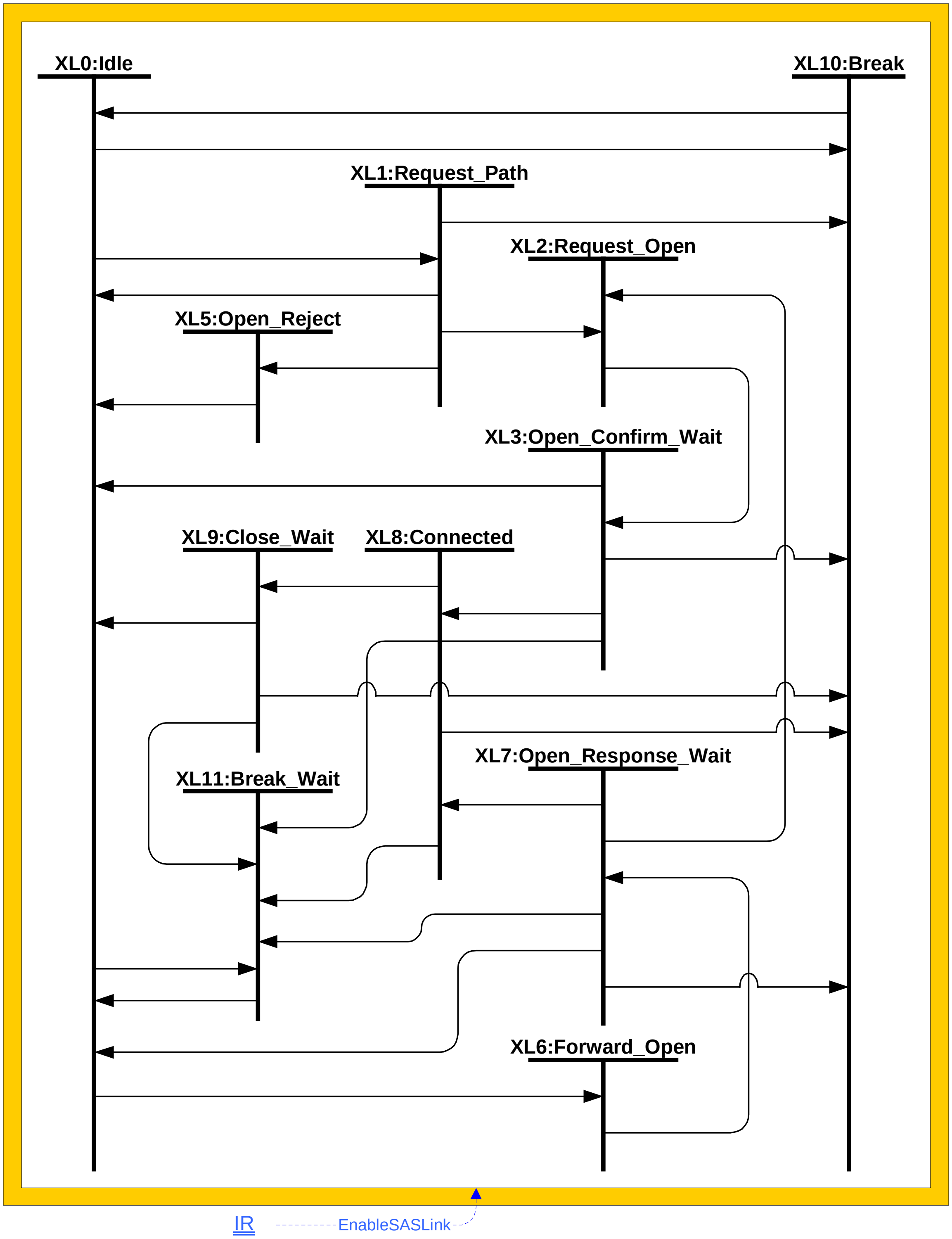

Figure 3. SAS Expander Connection state diagram

7.12.3.3 XL0:Idle state

7.12.3.3.1 XL0:Idle state description

The XL0:Idle state is the initial state and is the state that occurs when there is no pending or active connection.

The XL0:Idle state shall transmit idle dwords.

7.12.3.3.2 Transition XL0:Idle to XL1:Request_Path

The XL0:XL1 transition shall occur when the following conditions are met:

a) SendOpen indication is not being received from another link via the expander connection router;

b) SendBreak indication is not being received from another link via the expander connection router; and

c) a valid OPEN address frame is received.

7.12.3.3.3 Transition XL0:Idle to XL6:Forward_Open

The XL0:XL6 transition shall occur when a SendOpen indication is received from another XL state machine via the expander connection router.

7.12.3.3.4 Transition XL0:Idle to XL10:Break

The XL0:XL10 transition shall occur when a BREAK primitive sequence is received.

7.12.3.3.5 Transition XL0:Idle to XL11:Break_Wait

The XL0:XL11 transition shall occur when a SendBreak indication is received from another XL state machine via the expander connection router.

[TEH: this transition is to cover BREAK received immediately after ArbLost case]

7.12.3.4 XL1:Request_Path state

7.12.3.4.1 XL1:Request_Path state description

The XL1:Request_Path state is used to arbitrate for connection resources and to specify the destination of the connection.

The XL1:Request_Path state shall transmit the following:

a) AIP(WAITING ON PARTIAL) when an Arbitrating(WaitingOnPartial) confirmation is received from the expander connection manager;

b) AIP(WAITING ON PARTIAL) when an Arbitrating(BlockedOnPartial) confirmation is received from the expander connection manager;

c) AIP(WAITING ON CONNECTION) when an Arbitrating(WaitingOnConnection) confirmation is received from the expander connection manager; otherwise

d) AIP(NORMAL).

The XL1:Request_Path state shall send a RequestPath request to the expander connection manager with the following arguments:

a) destination SAS address;

b) source SAS address;

c) protocol;

d) link rate;

e) arbitration wait time;

f) pathway blocked count; and

g) partial pathway timeout status.

The XL1:Request_Path state shall initialize a partial pathway timeout timer to PPTOV microseconds (see X.Y) and shall decrement the timer once per microsecond, until reaching zero, while the Arbitrating(Blocked On Partial) confirmation is being received from the expander connection manager.

When the Arbitrating(Blocked On Partial) confirmation is not received, the partial pathway timeout timer shall be held at an initial value of PPTOV.

When the partial pathway timeout timer expires, i.e. reaches zero while receiving the Arbitrating(Blocked On Partial) confirmation, timeout status is conveyed to the expander connection manager via the partial pathway timeout status argument in the RequestPath request.

7.12.3.4.2 Transition XL1:Request_Path to XL2:Request_Open

The XL1:XL2 transition shall occur when an ArbWon confirmation is received from the expander connection manager.

7.12.3.4.3 Transition XL1:Request_Path to XL5:Open_Reject

The XL1:XL5 transition shall occur when an ArbReject confirmation is received from the expander connection manager.

7.12.3.4.4 Transition XL1:Request_Path to XL0:Idle

The XL1:XL0 transition shall occur when an ArbLost confirmation is received from the expander connection manager.

7.12.3.4.5 Transition XL1:Request_Path to XL10:Break

The XL1:XL10 transition shall occur when a BREAK primitive sequence is received.

7.12.3.5 XL2:Request_Open state

7.12.3.5.1 XL2:Request_Open state description

The XL2:Request_Open state is used to forward an OPEN address frame through the expander connection router to a destination link.

The XL2:Request_Open state shall transmit idle dwords.

The XL2:Request_Open state shall send a SendOpen request through the expander connection router to a destination link, received by the destination link as a SendOpen indication. The arguments to the SendOpen request/indication are:

a) destination SAS address;
b) source SAS address;
c) protocol;
d) link rate;
e) arbitration wait time;
f) initiator bit;
g) initiator connection tag;
h) features; and
i) pathway blocked count.

7.12.3.5.2 Transition XL2:Request_Open to XL3:Open_Confirm_Wait

The XL2:XL3 transition shall occur after the OPEN address frame has been forwarded to a destination link.

7.12.3.6 XL3:Open_Confirm_Wait state

7.12.3.6.1 XL3:Open_Confirm_Wait state description

The XL3:Open_Confirm_Wait state waits for confirmation to an OPEN address frame sent on a destination link.

The XL3:Open_Confirm_Wait state shall transmit the following:

a) AIP(NORMAL) when an ArbStatus(Normal) confirmation is received from a destination link;
b) AIP(WAITING ON PARTIAL) when an ArbStatus(Waiting On Partial) confirmation is received from a destination link;
c) AIP(WAITING ON CONNECTION) when an ArbStatus(Waiting On Connection) confirmation is received from a destination link;
d) AIP(WAITING ON DEVICE) when an ArbStatus(Waiting On Device) confirmation is received from a destination link;
e) OPEN_ACCEPT when an OpenAccept confirmation is received from a destination link;
f) OPEN_REJECT when an OpenReject confirmation is received from a destination link; or [TEH: enumerate OPEN_REJECTS and OpenReject confirmations here?]
g) idle dwords when none of the previous conditions are present.

[TEH: should this be a 1,2,3 list with an else/otherwise for idle dwords?]

The XL3:Open_Confirm_Wait state shall send a SendBreak request to a destination link when a BREAK primitive sequence is received.

7.12.3.6.2 Transition XL3:Open_Confirm_Wait to XL0:Idle

The XL3:XL0 transition shall occur after:

a) An OpenReject confirmation is received from a destination link and path resources have been released;
b) A BackoffRetry confirmation is received from a destination link and path resources have been released; or
c) A BackoffReversePath confirmation is received from a destination link.

~~7.12.3.6.3 Transition XL3:XL4 (Open_Confirm_Wait:Open_Backoff)~~

~~The XL3:XL4 transition shall occur when the Expander Function confirms via the Confirm.Backoff.Retry expander primitive that this link must release path resources and retry the connection in order to avoid deadlock.~~

~~7.12.3.6.4 Transition XL3:XL6 (Open_Confirm_Wait:Forward_Open)~~

~~The XL3:XL6 transition shall occur when the Expander Function confirms via the Confirm.Backoff.FwdOpen expander primitive that this link must forward the other link’s received OPEN address frame and switch roles to connection responder in order to avoid deadlock.~~

7.12.3.6.5 Transition XL3:Open_Confirm_Wait to XL8:Connected

The XL3:XL8 transition shall occur when an OpenAccept confirmation is received from a destination link.

7.12.3.6.6 Transition XL3:Open_Confirm_Wait to XL10:Break

The XL3:XL10 transition shall occur when a BREAK primitive sequence is received.

7.12.3.6.7 Transition XL3:Open_Confirm_Wait to XL11:Break_Wait

The XL3:XL11 transition shall occur when a SendBreak indication is received from a destination link.

~~7.12.3.7 XL4:Open_Backoff state~~

~~7.12.3.7.1 XL4:Open_Backoff state description~~

~~The Open_Backoff state is used to release path resources in order to avoid deadlock.~~

~~The Open_Backoff state shall transmit AIP to the SAS phy.~~

~~7.12.3.7.2 Transition XL4:XL1 (Open_Backoff:Request_Path)~~

~~The XL4:XL1 transition shall occur after path resources have been released for this connection request.~~

7.12.3.8 XL5:Open_Reject state

7.12.3.8.1 XL5:Open_Reject state description

The XL5:Open_Reject state is used to reject a connection request.

The XL5:Open_Reject state shall transmit the following:

a) OPEN_REJECT(NO DESTINATION) when an ArbReject(No Destination) confirmation is received from the expander connection manager;
b) OPEN_REJECT(BAD DESTINATION) when an ArbReject(Bad Destination) confirmation is received from the expander connection manager;
c) OPEN_REJECT(LINK RATE NOT SUPPORTED) when an ArbReject(Bad Link Rate) confirmation is received from the expander connection manager;
d) OPEN_REJECT(PATHWAY BLOCKED) when an ArbReject(Pathway Blocked) confirmation is received from the expander connection manager.

7.12.3.8.2 Transition XL5:Open_Reject to XL0:Idle

The XL5:XL0 transition shall occur after OPEN_REJECT has been transmitted.

7.12.3.9 XL6:Forward_Open state

7.12.3.9.1 XL6:Forward_Open state description

The XL6:Forward_Open state is used to transmit an OPEN address frame indicated by the SendOpen indication received from a source link via the expander connection router.

The XL6:Forward_Open state shall transmit an OPEN address frame with the fields set to the values specified by the SendOpen indication.

After sending the OPEN address frame, the XL6:Forward_Open state shall transmit idle dwords.

7.12.3.9.2 Transition XL6:Forward_Open to XL7:Open_Response_Wait

The XL6:XL7 transition shall occur after the OPEN address frame has been transmitted.

7.12.3.10 XL7:Open_Response_Wait state

7.12.3.10.1 XL7:Open_Response_Wait state description

The XL7:Open_Response_Wait state waits for a response to a transmitted OPEN address frame and determines the appropriate action to take based on the response.

The XL7:Open_Response_Wait state shall transmit idle dwords.

The XL7:Open_Response_Wait state shall send the following responses through the expander connection router to a source link, received by the source link as confirmations:

a) OpenAccept when OPEN_ACCEPT is received;
b) OpenReject when OPEN_REJECT is received;
c) BackoffRetry when a higher priority OPEN address frame is received and the source SAS address of the received OPEN address frame is not equal to the destination SAS address of the transmitted OPEN address frame;

d) BackoffReversePath when a higher priority OPEN address frame is received and the source SAS address of the received OPEN address frame is equal to the destination SAS address of the transmitted OPEN address frame.

[TEH: enumerate OpenRejects ?]

The XL7:Open_Response_Wait state shall send the following arbitration responses through the expander connection router to a source link, received by the source link as confirmations:

a) ArbStatus(Waiting On Device) when AIP has not been received;
b) ArbStatus(Normal) when AIP(NORMAL) is received;
c) ArbStatus(Waiting On Partial) when AIP(WAITING ON PARTIAL) is received;
d) ArbStatus(Waiting On Connection) when AIP(WAITING ON CONNECTION) is received;
e) ArbStatus(Waiting On Device) when AIP(WAITING ON DEVICE) is received.

The XL7:Open_Response_Wait state shall send a SendBreak request to a source link when a BREAK primitive sequence is received.

7.12.3.10.2 Transition XL7:Open_Response_Wait to XL0:Idle

The XL7:XL0 transition shall occur when one of the following conditions are met:

a) OPEN_REJECT is received and path resources have been released; or
b) a higher priority OPEN address frame is received and the source SAS address of the received OPEN address frame is not equal to the destination SAS address of the transmitted OPEN address frame and path resources have been released.

~~7.12.3.10.3 Transition XL7:XL1 (Open_Response_Wait:Request_Path)~~

~~The XL7:XL1 transition causes a role change from connection responder to connection requestor and shall occur when any of the following conditions are met:~~

~~c) AIP has not been received and a higher priority OPEN address frame is received which requires path resources not currently allocated to this pending connection~~

~~AIP has been received and any OPEN address frame is received which requires path resources not currently allocated to this pending connection~~

7.12.3.10.4 Transition XL7:Open_Response_Wait to XL2:Request_Open

The XL7:XL2 transition shall occur when a higher priority OPEN address frame is received and the source SAS address of the received OPEN address frame is equal to the destination SAS address of the transmitted OPEN address frame.

7.12.3.10.5 Transition XL7:Open_Response_Wait to XL8:Connected

The XL7:XL8 transition shall occur when OPEN_ACCEPT is received.

7.12.3.10.6 Transition XL7:Open_Response_Wait to XL10:Break

The XL7:XL10 transition shall occur when a BREAK primitive sequence is received.

7.12.3.10.7 Transition XL7:Open_Response_Wait to XL11:Break_Wait

The XL7:XL11 transition shall occur when a SendBreak indication is received from a source link.

7.12.3.11 XL8:Connected state

7.12.3.11.1 XL8:Connected state description

The XL8:Connected state provides a full-duplex circuit between two links within an expander.

The XL8:Connected state shall transmit all dwords received by the SendDword indication from a connected link via the expander connection router.

The XL8:Connected state shall send all valid dwords received by the SAS phy through the expander connection router to a connected link using the SendDword request with the exception of the BREAK and CLOSE primitive sequences.

The XL8:Connected state shall replace invalid dwords received by the SAS phy as specified in section 7.12.4

The XL8:Connected state shall send a SendClose request to a connected link when a CLOSE primitive sequence is received.

The XL8:Connected state shall send a SendBreak request to a connected link when a BREAK primitive sequence is received.

7.12.3.11.2 Transition XL8:Connected to XL9:Close_Wait

The XL8:XL9 transition shall occur when a SendClose indication is received from a connected link via the expander connection router.

7.12.3.11.3 Transition XL8:Connected to XL10:Break

The XL8:XL10 transition shall occur when a BREAK primitive sequence is received.

7.12.3.11.4 Transition XL8:Connected to XL11:Break_Wait

The XL8:XL11 transition shall occur when a SendBreak indication is received from a connected link via the expander connection router.

7.12.3.12 XL9:Close_Wait state

7.12.3.12.1 XL9:Close_Wait state description

The XL9:Close_Wait state closes a connection and releases path resources.

The XL9:Close_Wait state shall transmit a CLOSE primitive sequence.

The XL9:Close_Wait state shall send a SendClose request to a connected link when a CLOSE primitive sequence is received.

The XL9:Close_Wait state shall send a SendBreak request to a connected link when a BREAK primitive sequence is received.

7.12.3.12.2 Transition XL9:Close_Wait to XL0:Idle

The XL9:XL0 transition shall occur after a CLOSE primitive sequence has been both transmitted and received and after path resources have been released for this connection.

7.12.3.12.3 Transition XL9:Close_Wait to XL10:Break

The XL9:XL10 transition shall occur when a BREAK primitive sequence is received.

7.12.3.12.4 Transition XL9:Close_Wait to XL11:Break_Wait

The XL9:XL10 transition shall occur when a SendBreak indication is received from a connected link via the expander connection router.

7.12.3.13 XL10:Break state

7.12.3.13.1 XL10:Break state description

The XL10:Break state closes any connection and releases any path resources.

The XL10:Break state shall transmit a BREAK primitive sequence.

7.12.3.13.2 Transition XL10:Break to XL0:Idle

The XL10:XL0 transition shall occur after transmitting a BREAK primitive sequence.

7.12.3.14 XL11:Break_Wait state

7.12.3.14.1 XL11:Break_Wait state description

The XL11:Break_Wait state closes any connection and releases any path resources.

The XL11:Break_Wait state shall transmit a BREAK primitive sequence. After transmitting the BREAK primitive sequence the XL11:Break_Wait state shall initialize a break timeout timer to 1 millisecond and start the timer.

7.12.3.14.2 Transition XL11:Break_Wait to XL0:Idle

The XL11:XL0 transition shall occur after a BREAK primitive sequence is received or after the break timeout timer expires, whichever occurs first.