To: T10 Technical Committee
From: Timothy Hoglund, LSI Logic
Date: 12 January 2004
Subject: T10/04-036r1 SAS-1.1: AIP clarifications and XL state transition

Revision History

Revision 0 (06 January 2004) first revision
Revision 1 (12 January 2004) incorporate feedback from T10/SAS Protocol Working Group meeting

Related Documents

sas1r02 - Serial Attached SCSI 1.1 revision 2

Overview

This proposal seeks to clarify and correct a few expander-related issues present within the current working draft of the SAS 1.1 specification:

1. Define expected response of a SAS expander when a connection request cannot be completed due to insufficient routing resources.

The current SAS specification requires the Expander Connection Router (ECR) to provide sufficient routing resources to support at least one connection (see 4.6.4). However, what is not specified is the expander's response should it not have enough routing resources to satisfy a a connection request.

This proposal calls for an Arbitrating (Waiting On Connection) response to be sent from the ECM to an expander phy when insufficient routing resources are available to complete a connection request.

2. Correct behavior regarding when AIP (WAITING ON DEVICE) is sent.

In order for the OPEN Timeout mechanism to work properly, an expander phy which has forwarded an OPEN Address frame should only send the Arb Status (Waiting On Device) response once of it's own accord, then one-for-one with any AIP (WAITING ON DEVICE) that may be received. Currently the behavior is incorrectly defined in the XL6:Open_Response_Wait state (see 7.15.9.1) as to require an expander phy to send an Arb Status (Waiting On Device) response whenever an AIP Received message has not been received.

This proposal calls for the sending of the Arb Status (Waiting On Device) response only upon entry to XL6:Open_Response_Wait or whenever an AIP Received (Waiting On Device) message is received.

3. Clarify what it means to repeatedly send Transmit AIP messages when in XL1:Request_Path state.

The XL1:Request_Path state (see 7.15.4.1) uses the term "repeatedly send a Transmit AIP (type) message to the XL transmitter" several times. This verbage conflicts with rules for transmitting AIP define in 7.12.5.1 as follows:

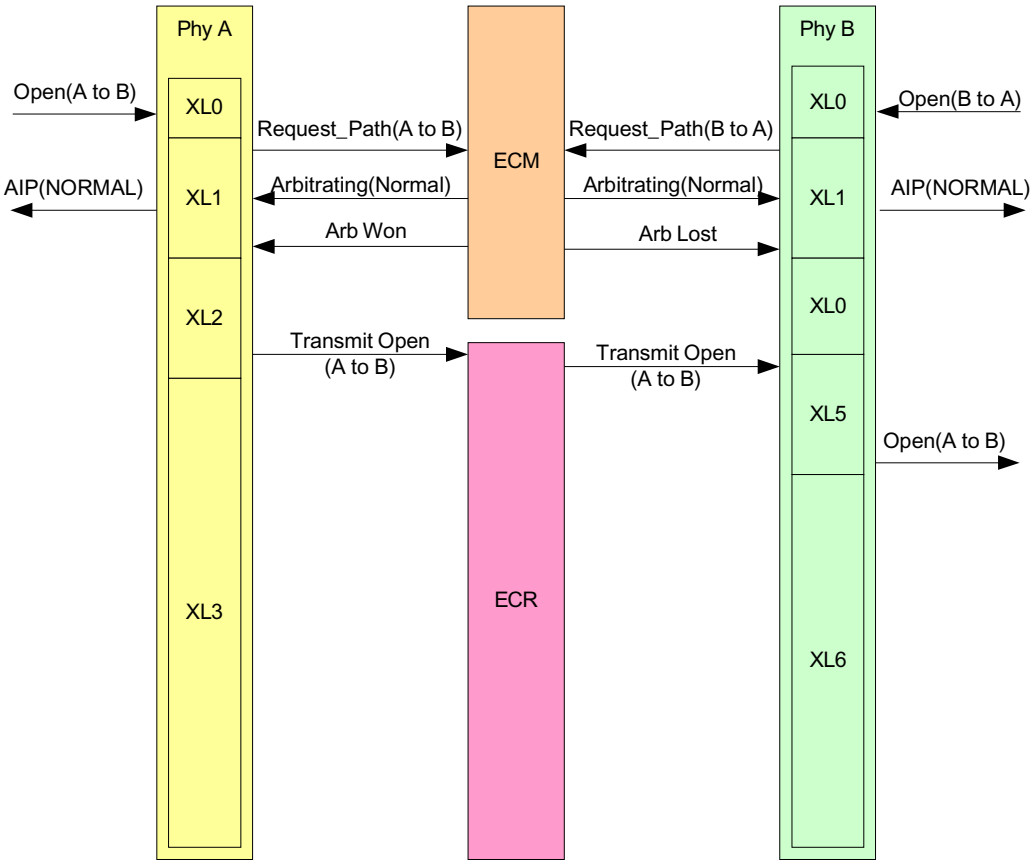
Expander devices shall transmit no more than three consecutive AIPs without transmitting an idle dword. Expander devices shall transmit at least one AIP every 128 dwords.

This proposal specifies that both Transmit AIP and Transmit Idle Dword messages are sent by the XL1:Request_Path state in accordance with 7.12.5.1.

4. Close a timing hole present in the XL state machine.

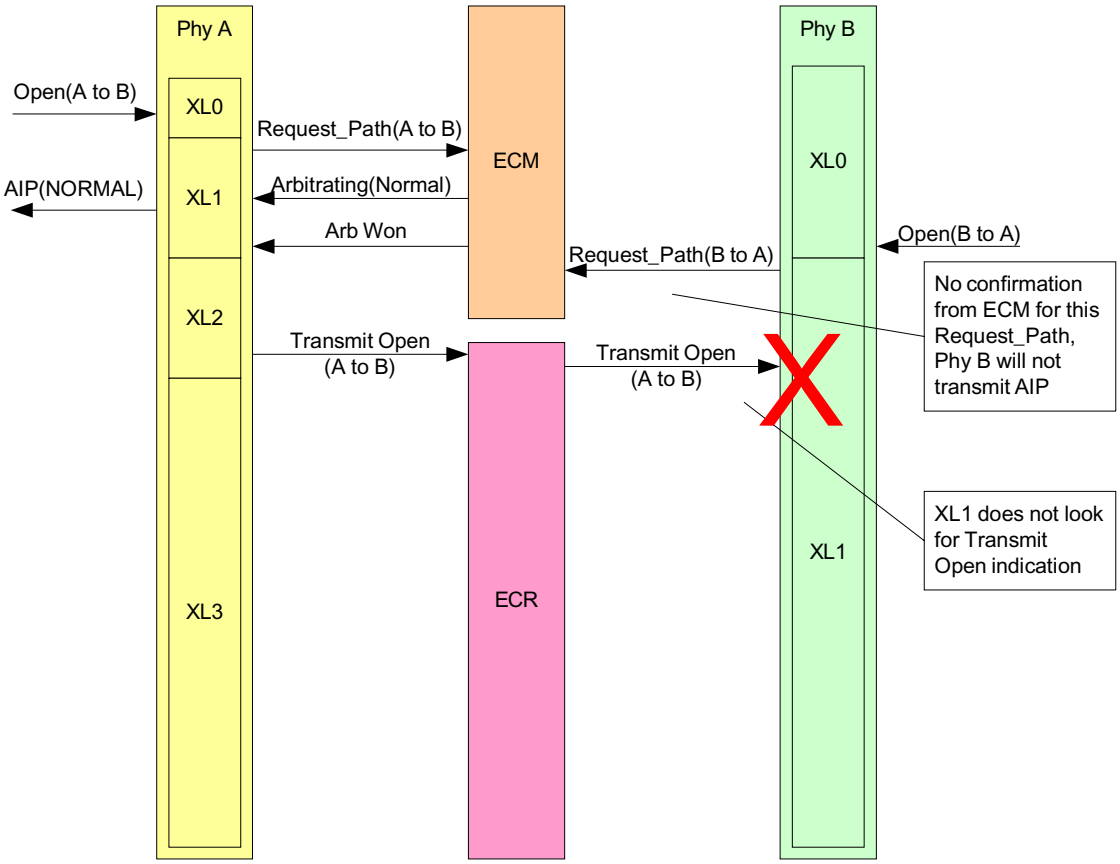
Should two expander phys arbitrate for each other at the same time, the ECM will provide an Arb Won confirmation to one and an Arb Lost confirmation to the other and both phys will properly resolve the situation.

Figure 1 — Arb Lost



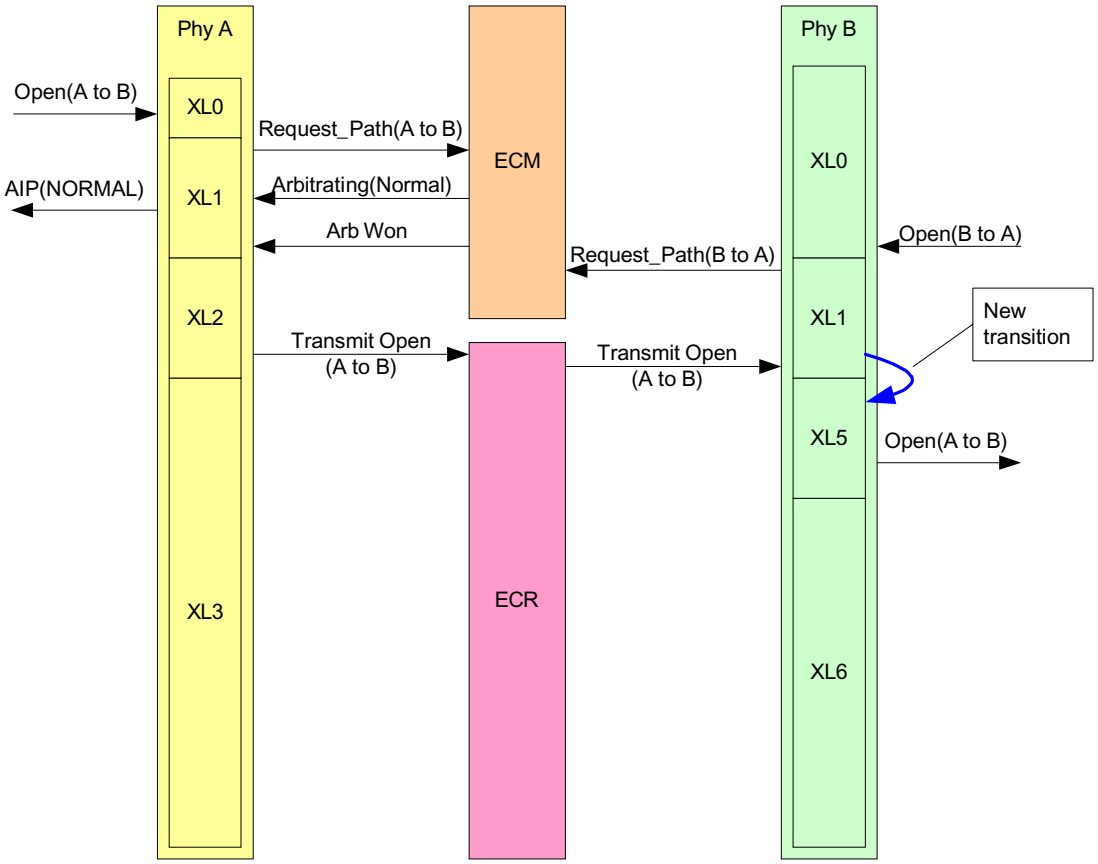
However, a window of misopportunity exists whereby if phy B arbitrates for phy A after phy A has received an Arb Won confirmation, phy B will get stuck in the XL1:Request_Path state and not detect the Transmit Open indication from phy A.

Figure 2 — Missed Transmit Open



This proposal calls for the addition of a state transition from XL1:Request_Path to XL5:Forward_Open and the passing of an OPEN Address Frame Received argument along with this transition to properly resolve the scenario.

Figure 3 — Transmit Open Detected and Processed



5. Update Annex G diagrams based on above changes.

This proposal includes recommended changes to Annex G diagrams in accordance with the suggested normative changes outlined above and detailed below.

Suggested Changes

4.6.6.3 ECM interface

< ... other stuff ... >

Table Table 11 — describes the confirmations from the ECM to an expander phy. These confirmations are sent in confirmation of a Request Path request.

Table 11 — ECM to expander phy confirmations

Message	Description
Arbitrating (Normal)	Confirmation that the ECM has received the Request Path request.
Arbitrating (Waiting On Partial)	Confirmation that the ECM has determined that: a) there is a destination port capable of routing to the requested destination SAS address; b) at least one phy within the destination port supports the requested connection rate; c) each of the phys within the destination port is returning a Phy Status (Partial Pathway) or Phy Status (Blocked Partial Pathway) response; and d) at least one of the phys within the destination port is returning a Phy Status (Partial Pathway) response.
Arbitrating (Blocked On Partial)	Confirmation that the ECM has determined that: a) there is a destination port capable of routing to the requested destination SAS address; b) at least one phy within the destination port supports the requested connection rate; c) each of the phys within the destination port is returning a Phy Status (Blocked Partial Pathway) response.
Arbitrating (Waiting On Connection)	Confirmation that the ECM has determined that the connection request is blocked due to one of the following reasons: a) the connection request is blocked by an active connection; or b) there are insufficient routing resources within the expander to complete the connection request. A connection request shall be considered as blocked by an active connection when: a) there is a destination port capable of routing to the requested destination SAS address; b) at least one phy within the destination port supports the requested connection rate; c) each of the phys within the destination port is returning a Phy Status (Partial Pathway), Phy Status (Blocked Partial Pathway), or Phy Status (Connection) response; and d) at least one of the phys within the destination port is returning a Phy Status (Connection) response.
Arb Won	Confirmation that an expander phy has won path arbitration.
Arb Lost	Confirmation that an expander phy has lost path arbitration.
Arb Reject (No Destination)	Confirmation that the ECM did not find an operational expander phy capable of routing to the requested destination SAS address.
Arb Reject (Bad Destination)	Confirmation that the ECM has determined that the requested destination SAS address maps back to the requesting port.
Arb Reject (Bad Connection Rate)	Confirmation that the ECM has determined that there is a destination port capable of routing to the requested destination SAS address but no phys within the destination port are configured to support the requested connection rate.
Arb Reject (Pathway Blocked)	Confirmation that the ECM has determined that the requesting expander phy shall back off according to SAS pathway recovery rules.

7.2.5 Primitives not specific to type of connections

7.2.5.1 AIP (Arbitration in progress)

AIP is sent by an expander device after a connection request to indicate that the connection request is being processed and indicate the status of the connection request.

The versions of AIP representing different statuses are defined in table 57.

Table 57 — AIP primitives

Primitive	Description
AIP (NORMAL)	Expander device has just accepted the connection request.
AIP (RESERVED 0)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 1)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 2)	Reserved. Processed the same as AIP (NORMAL).
AIP (WAITING ON CONNECTION)	Expander device has determined the routing for the connection request, but either the destination phys are all being used for connections or there are insufficient routing resources to complete the connection request.
AIP (WAITING ON DEVICE)	Expander device has determined the routing for the connection request and forwarded it to the output physical link.
AIP (WAITING ON PARTIAL)	Expander device has determined the routing for the connection request, but the destination phys are all busy with other partial pathways (i.e., connection requests that have not reached the destination phy).
AIP (RESERVED WAITING ON PARTIAL)	Reserved. Processed the same as AIP (WAITING ON PARTIAL).

See 7.12 for details on connections.

7.12.4 Arbitration and resource management in an expander device

7.12.4.1 Arbitration overview

< ... other stuff ... >

The ECM shall generate the Arb Lost confirmation when all of the following conditions are met:

a) the connection request maps to an available expander phy at a supported connection rate; ~~and~~
b) there are sufficient routing resources to route the connection request; and
c) the destination expander phy of this connection request has received a higher priority OPEN address frame with this expander phy as its destination (i.e., when two expander phys both receive an OPEN address frame destined for each other, the ECM shall provide the Arb Lost confirmation to the expander phy that received the lowest priority OPEN address frame).

The ECM shall generate the Arb Won confirmation when all of the following conditions are met:

a) the connection request maps to an available expander phy at a supported connection rate; ~~and~~
b) no higher priority connection requests are present with this expander phy as the destination; and
c) there are sufficient routing resources to route the connection request.

4.6.6.4 ECR interface

< ...other stuff...>

Table Table 13 — describes the responses from an expander phy to the ECR and the corresponding confirmations from the ECR to another expander phy. These responses are sent in response to a Transmit Open indication.

Table 13 — Expander phy to ECR to expander phy responses and confirmations

Message	Description
Arb Status (Normal)	Confirmation/response that AIP (NORMAL) has been received.
Arb Status (Waiting On Partial)	Confirmation/response that AIP (WAITING ON PARTIAL) has been received.
Arb Status (Waiting On Connection)	Confirmation/response that AIP (WAITING ON CONNECTION) has been received.
Arb Status (Waiting On Device)	Confirmation/response that an expander phy has completed the forwarding of an OPEN Address frame and has entered the XL6:Open_Response_Wait state or AIP (WAITING ON DEVICE) has been received.
Open Accept	Confirmation/response that OPEN_ACCEPT has been received.
Open Reject	Confirmation/response that OPEN_REJECT has been received.
Backoff Retry	Confirmation/response that: a) a higher priority OPEN address frame has been received (see 7.12.3); and b) the source SAS address and connection rate of the received OPEN address frame are not equal to the destination SAS address and connection rate of the transmitted OPEN address frame.
Backoff Reverse Path	Confirmation/response that: a) a higher priority OPEN address frame has been received (see 7.12.3); and b) the source SAS address and connection rate of the received OPEN address frame are equal to the destination SAS address and connection rate of the transmitted OPEN address frame.

7.15.9 XL6:Open_Response_Wait state

7.15.9.1 State description

< ...other stuff... >

This state shall send the following responses through the ECR to a source phy, received by the source phy as confirmations:

a) an Arb Status (Waiting On Device) response ~~when an AIP Received message has not been received~~ upon entry to this state;
b) an Arb Status (Normal) response when an AIP Received (Normal) message is received;
c) an Arb Status (Waiting On Partial) response when an AIP Received (Waiting On Partial) message is received;
d) an Arb Status (Waiting On Connection) response when an AIP Received (Waiting On Connection) message is received; and
e) an Arb Status (Waiting On Device) response when an AIP Received (Waiting On Device) message is received.

7.15.4 XL1:Request_Path state

7.15.4.1 State description

This state is used to arbitrate for connection resources and to specify the destination of the connection.

If an Arbitrating (Normal) confirmation is received, this state shall repeatedly send ~~a~~ Transmit AIP (Normal) and Transmit Idle Dword messages to the XL transmitter in accordance with AIP transmission rules (see 7.12.5.1).

If an Arbitrating (Waiting On Partial) or Arbitrating (Blocked On Partial) confirmation is received, this state shall repeatedly send ~~a~~ Transmit AIP (Waiting On Partial) and Transmit Idle Dword messages to the XL transmitter in accordance with AIP transmission rules (see 7.12.5.1).

If an Arbitrating (Waiting On Partial) confirmation is received, this state shall repeatedly send a Phy Status (Partial Pathway) message to the ECM.

If an Arbitrating (Blocked On Partial) confirmation is received, this state shall repeatedly send a Phy Status (Blocked Partial Pathway) message to the ECM.

If an Arbitrating (Waiting On Connection) confirmation is received, this state shall repeatedly send ~~a~~ Transmit AIP (Waiting On Connection) and Transmit Idle Dword messages to the XL transmitter in accordance with AIP transmission rules (see 7.12.5.1).

If an Arbitrating (Waiting On Connection) confirmation is received, this state shall repeatedly send a Phy Status (Connection) message to the ECM.

< ...other stuff... >

7.15.4.6 Transition XL1:Request_Path to XL5:Forward_Open

This transition shall occur if a Transmit Open indication is received and none of the following confirmations have been received:

a) Arbitrating (Normal);
b) Arbitrating (Waiting On Partial);
c) Arbitrating (Blocked On Partial);
d) Arbitrating (Waiting On Connection);
e) Arb Won;
f) Arb Lost;
g) Arb Reject (No Destination);
h) Arb Reject (Bad Destination);
i) Arb Reject (Bad Connection Rate); or
j) Arb Reject (Pathway Blocked).

This transition shall include an OPEN Address Frame Received argument containing the arguments received in the Transmit Open indication.

Figure Figure 90 — shows several states in the XL state machine.

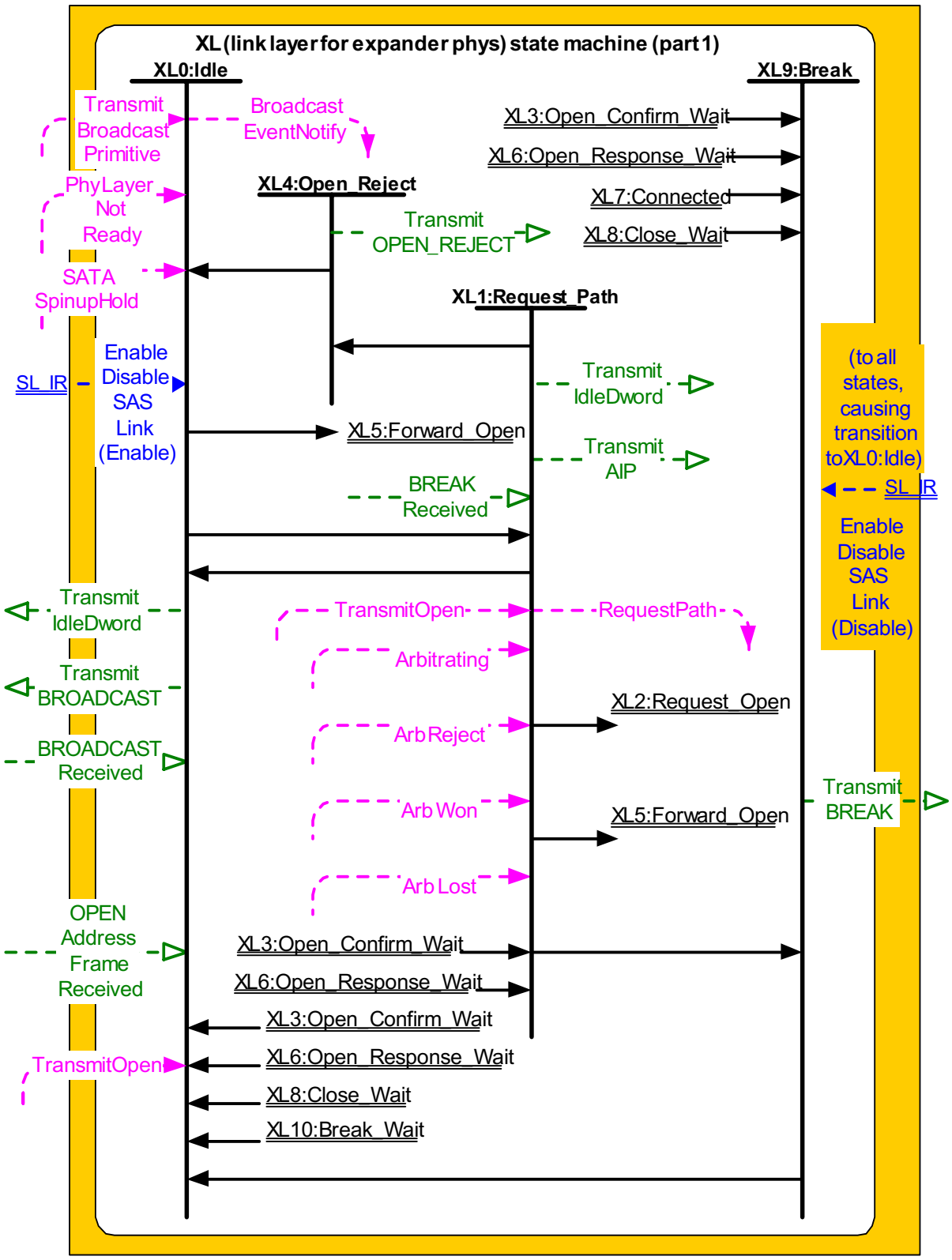


Figure 90 — XL (link layer for expander phys) state machine (part 1)

G.2 Connection request - OPEN_ACCEPT

Figure 2 shows the establishment of a successful connection between two end devices.

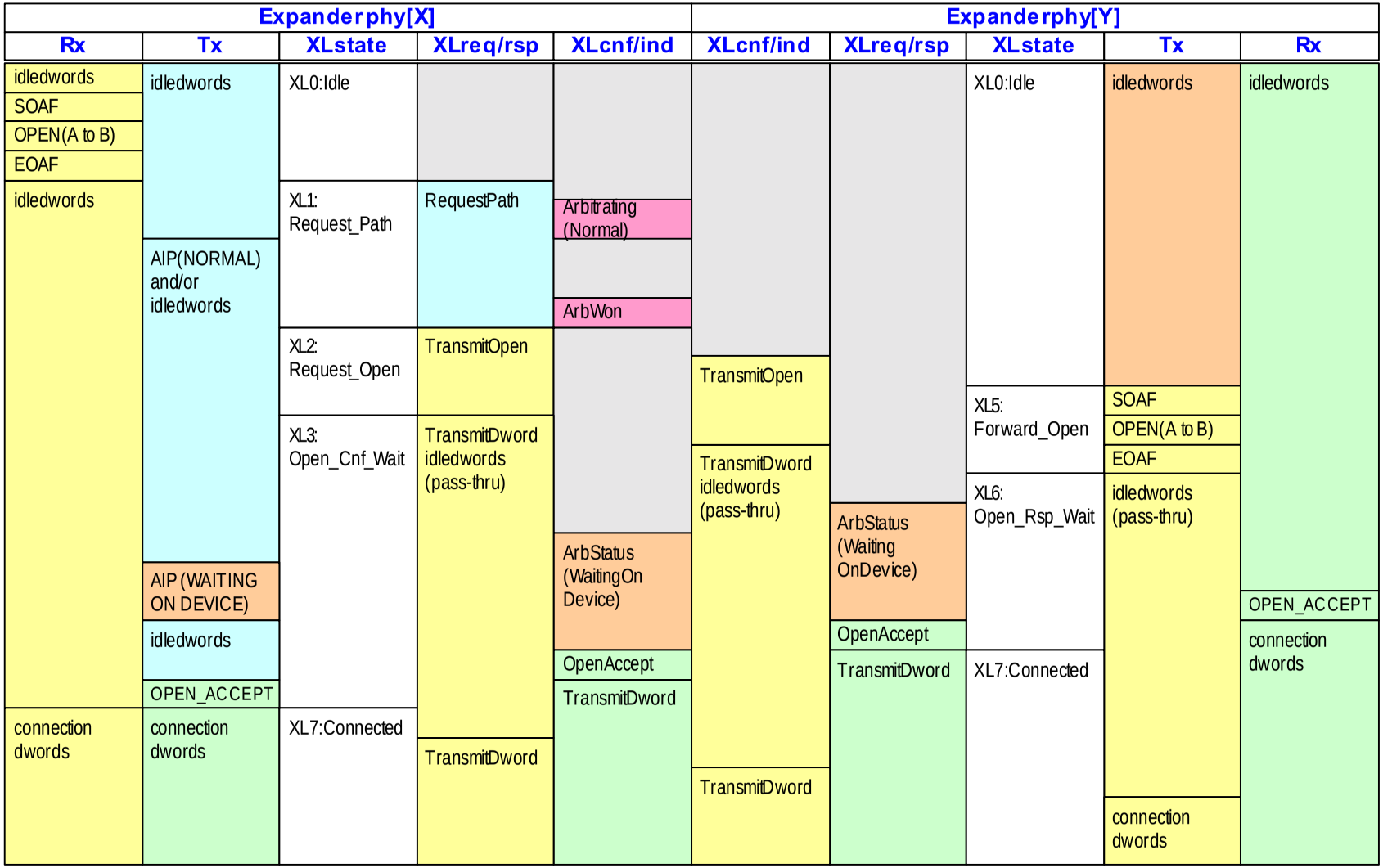


Figure G.2 — Connection request - OPEN_ACCEPT

G.3 Connection request - OPEN_REJECT by end device

Figure 3 shows failure to establish a connection due to rejection of the connection request by an end device.

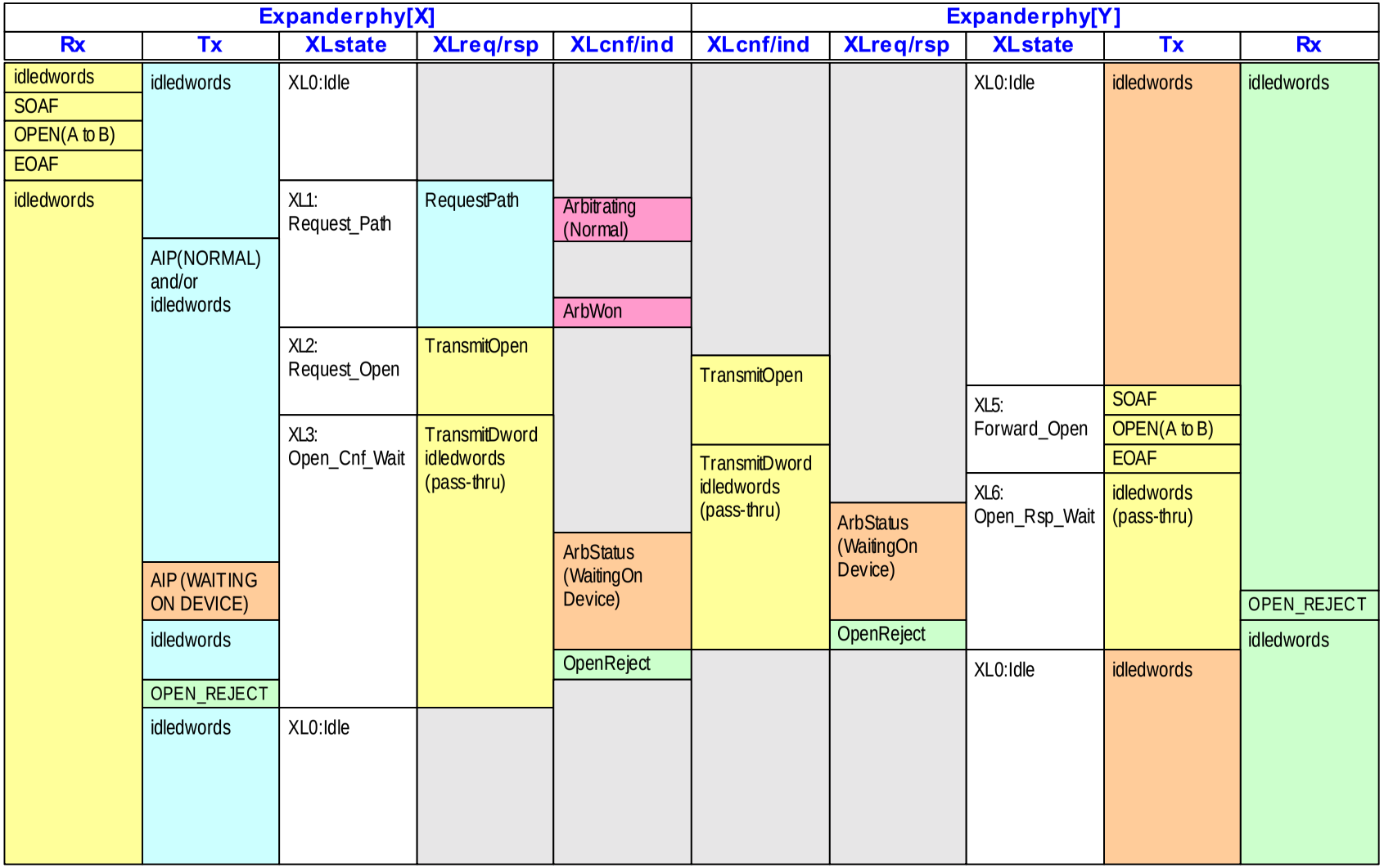


Figure G.3 — Connection request - OPEN_REJECT by end device

G.4 Connection request - OPEN_REJECT by expander device

Figure 4 shows failure to establish a connection due to rejection of the connection request by an expander device.

Expanderphy[X]					Expanderphy[Y]				
Rx	**Tx**	**XLstate**	**XLreq/rsp**	**XLcnf/ind**	**XLcnf/ind**	**XLreq/rsp**	**XLstate**	**Tx**	**Rx**
idledwords	idledwords	XL0:Idle					XL0:Idle	idledwords	idledwords
SOAF									
OPEN(A to B)									
EOAF									
idledwords		XL1: Request_Path	RequestPath						
				Arbitrating (Normal)					
	AIP(NORMAL) and/or idledwords								
				ArbReject					
	OPEN_REJECT	XL4:Open_Rejec							
	idledwords	XL0:Idle							

Figure G.4 — Connection request - OPEN_REJECT by expander device

G.5 Connection request - arbitration lost

Figure 5 shows two end devices attempting to establish a connection at the same time. This example assumes that the OPEN (A to B) address frame has higher priority than the OPEN (B to A) address frame and therefore device A wins arbitration and device B loses arbitration.

Figure G.5 — Connection request - arbitration lost

G.6 Connection request - backoff and retry

Figure 6 shows a higher priority OPEN address frame (B to C) received by a phy which has previously forwarded an OPEN address frame (A to B) whose source (A) differs from the winning destination (C). In this case expander phy [X] is required to back off and retry path arbitration (see 7.15.9).

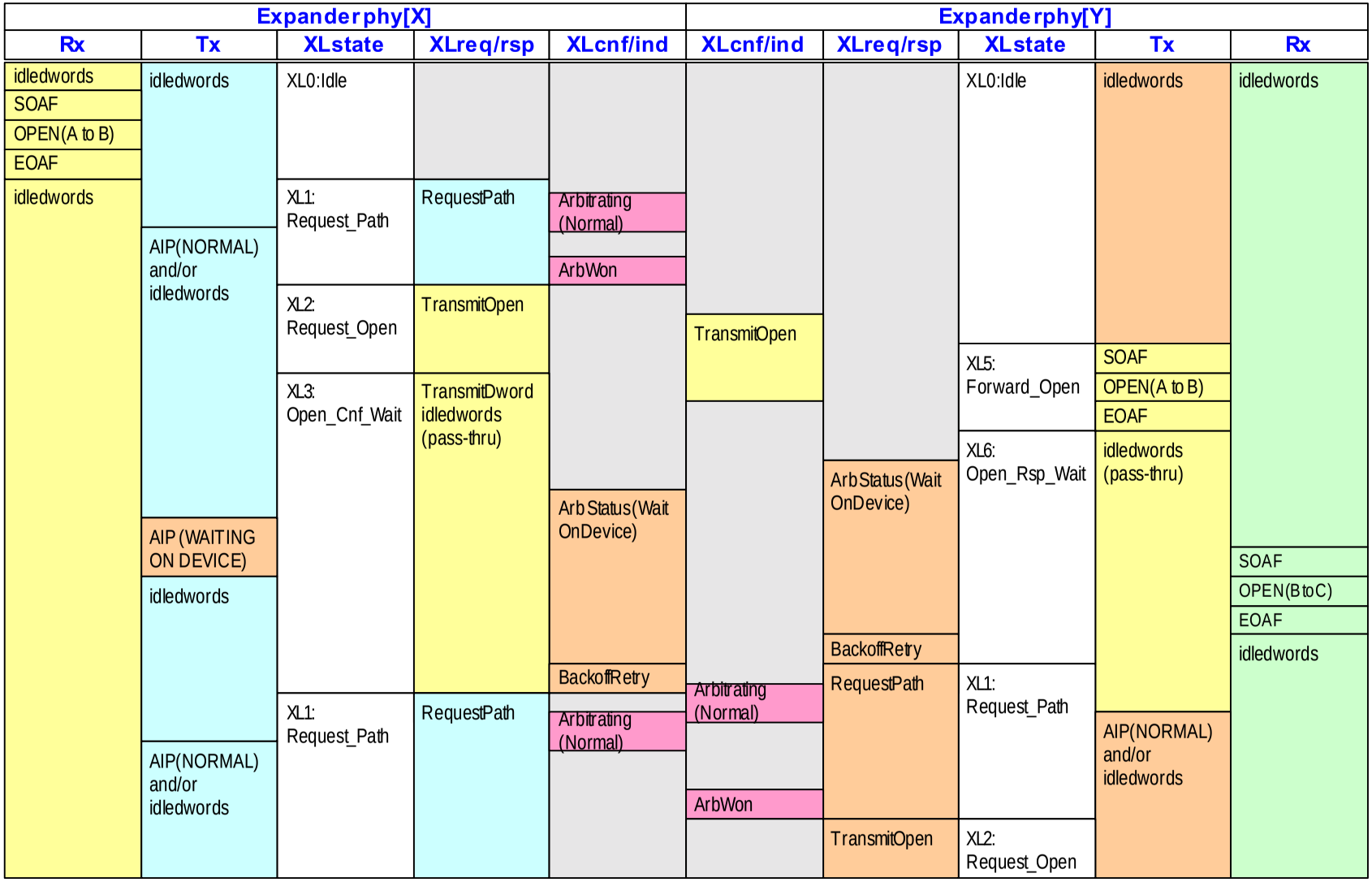


Figure G.6 — Connection request - backoff and retry

G.7 Connection request - backoff and reverse path

Figure 7 shows a higher priority OPEN address frame (B to A) received by a phy which has previously forwarded an OPEN address frame (A to B) whose source (A) matches the winning destination (A). In this case expander phy [Y] forwards the higher priority OPEN to expander phy [X] (see 7.15.9).

Figure G.7 — Connection request - backoff and reverse path

G.8 Connection close - single step

Figure 8 shows an end device initiating the closing of a connection by transmitting CLOSE, followed by another end device responding with CLOSE at a later time.

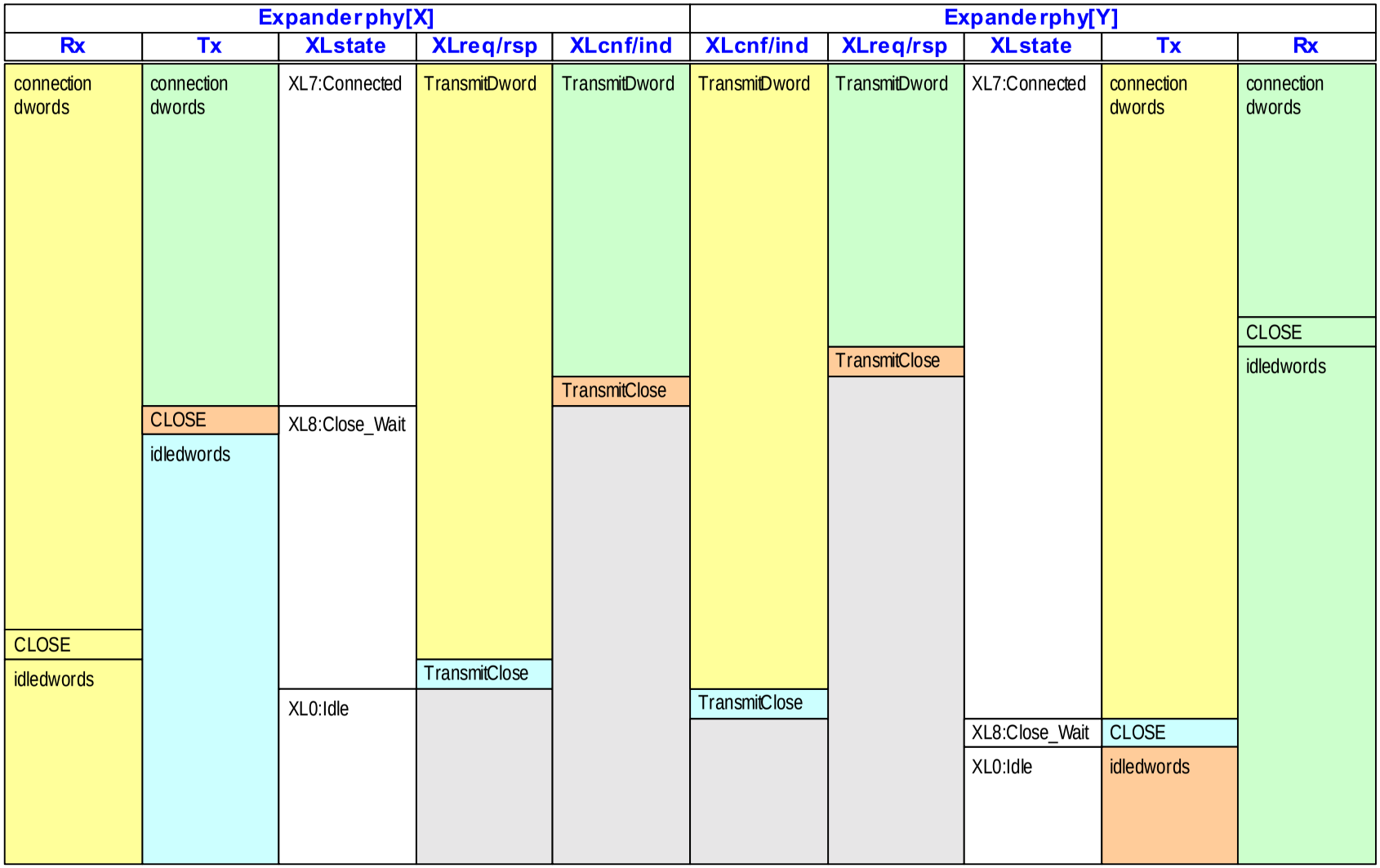


Figure G.8 — Connection close - single step

G.9 Connection close - simultaneous

Figure 9 shows two end devices simultaneously transmitting CLOSE to each other.

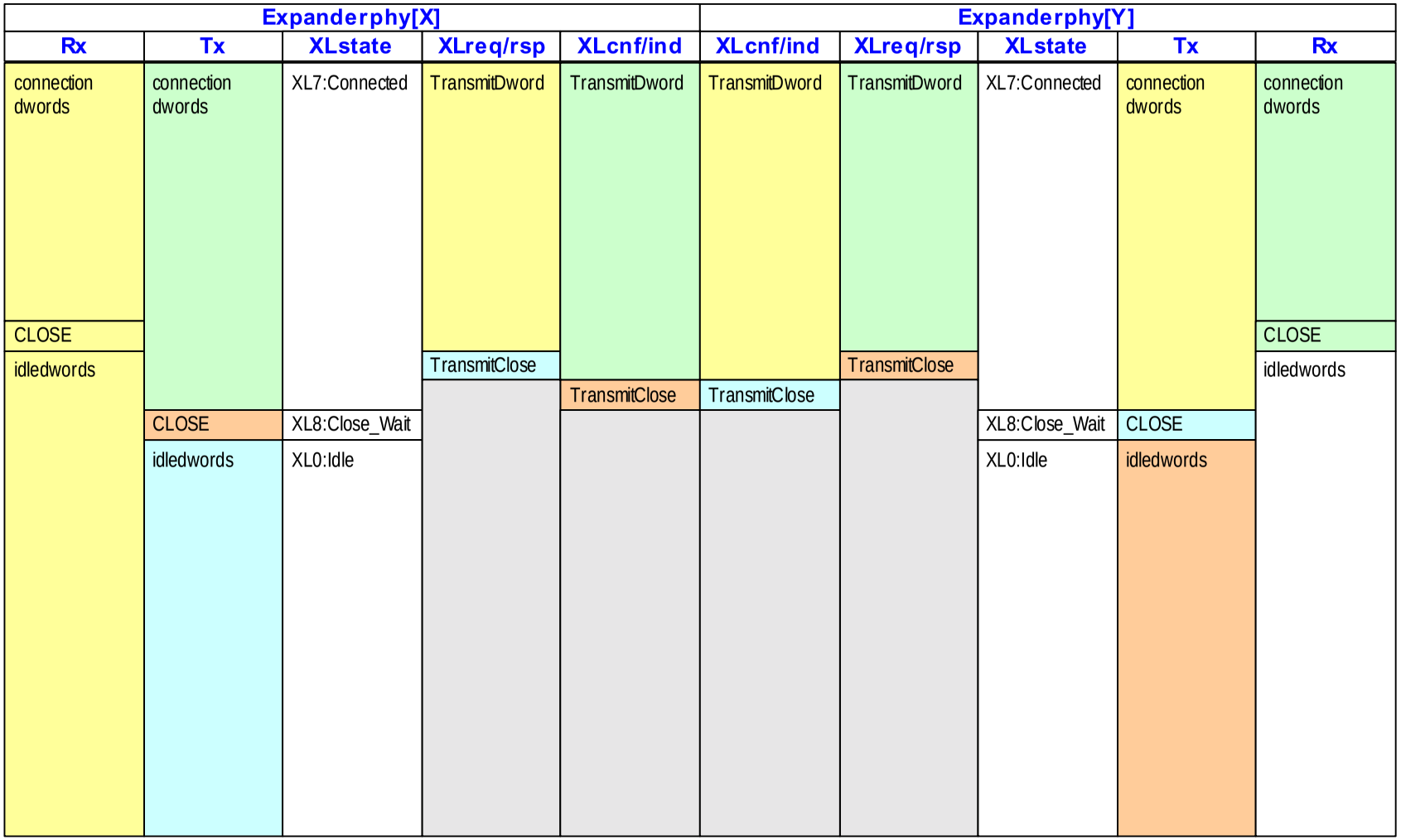


Figure G.9 — Connection close - simultaneous

G.10 BREAK handling during path arbitration

Figure 10 shows an expander device responding to the reception of a BREAK during path arbitration.

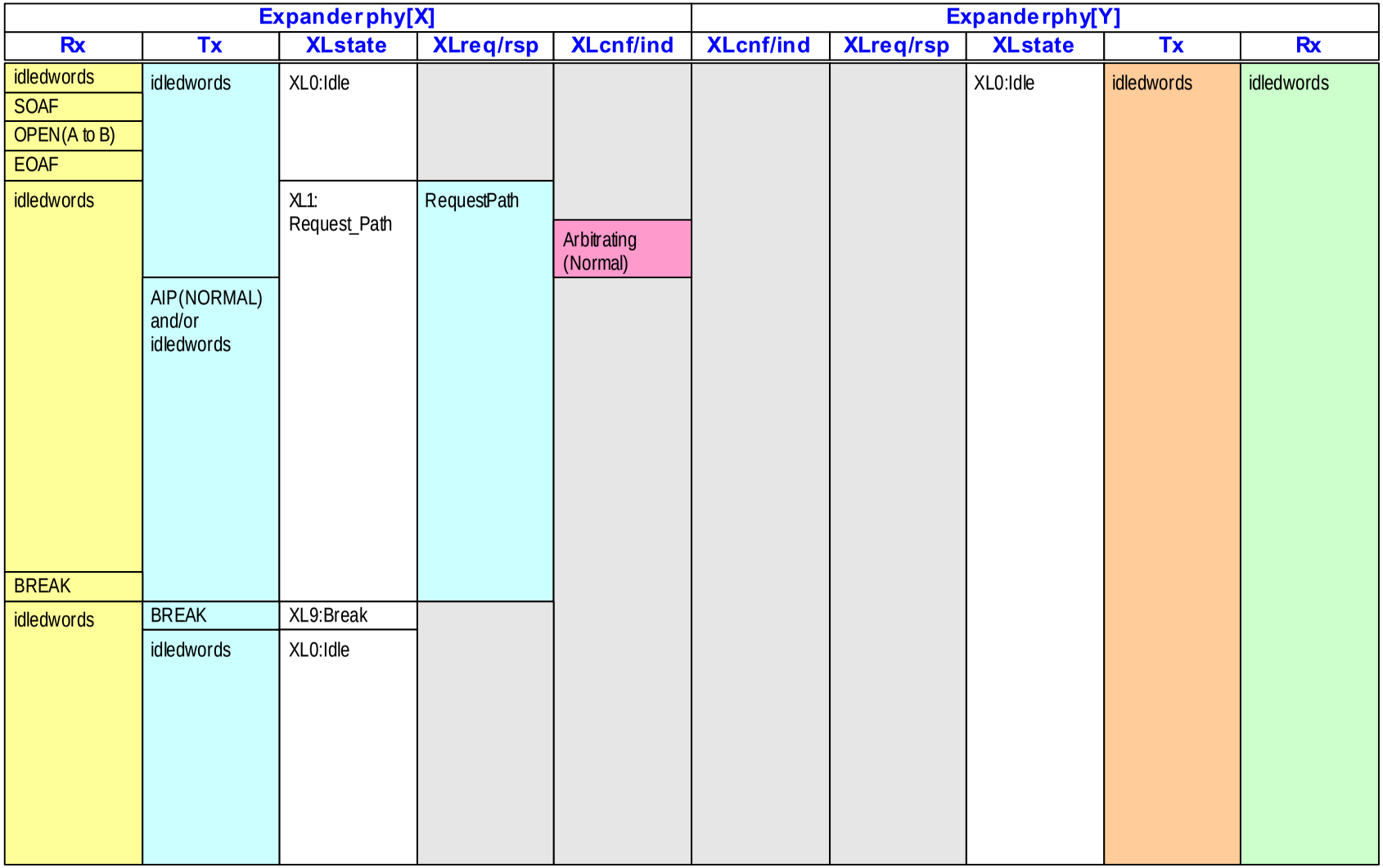


Figure G.10 — BREAK handling during path arbitration

G.11 BREAK handling during connection

Figure 11 shows an expander device responding to the reception of a BREAK during a connection.

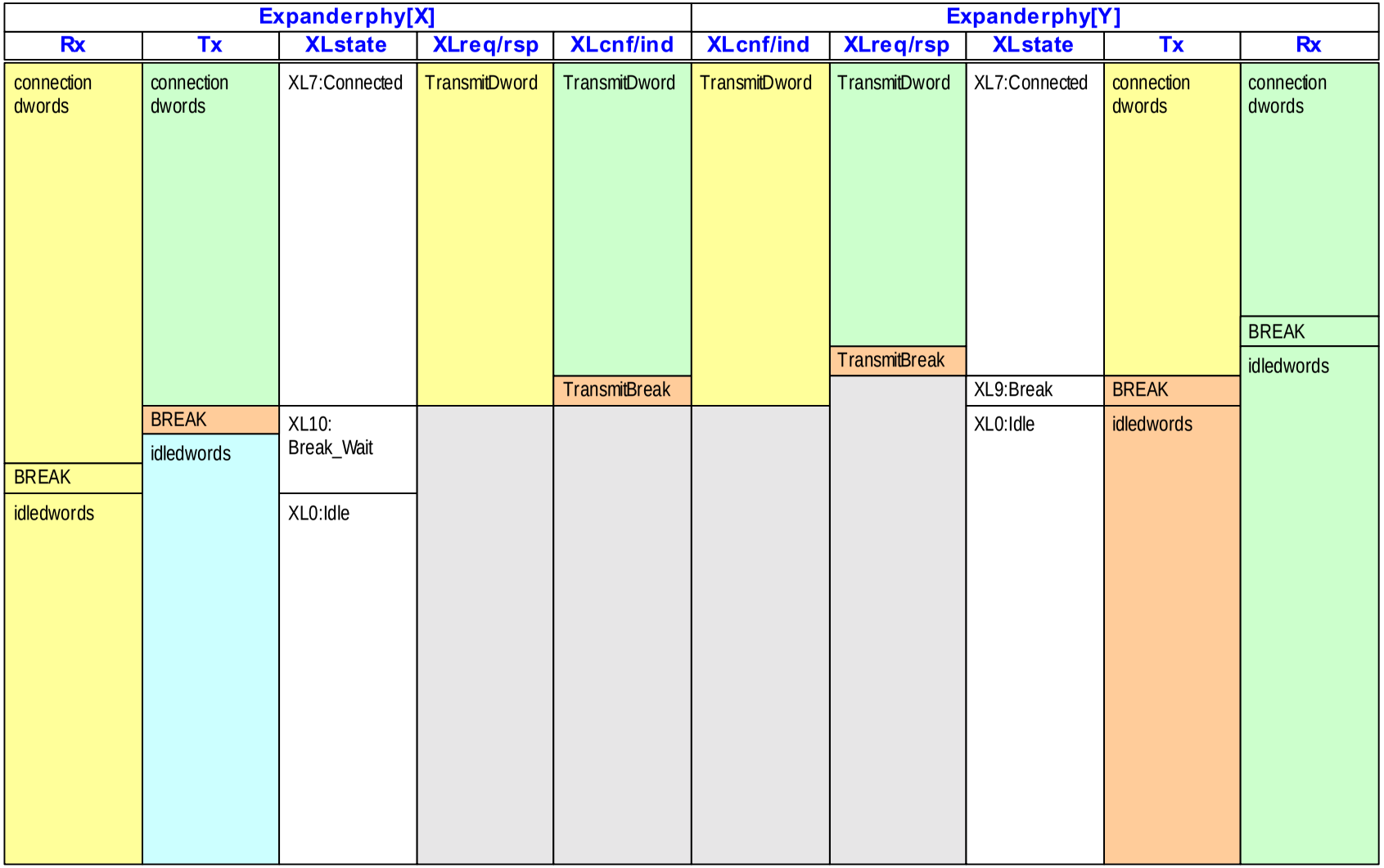
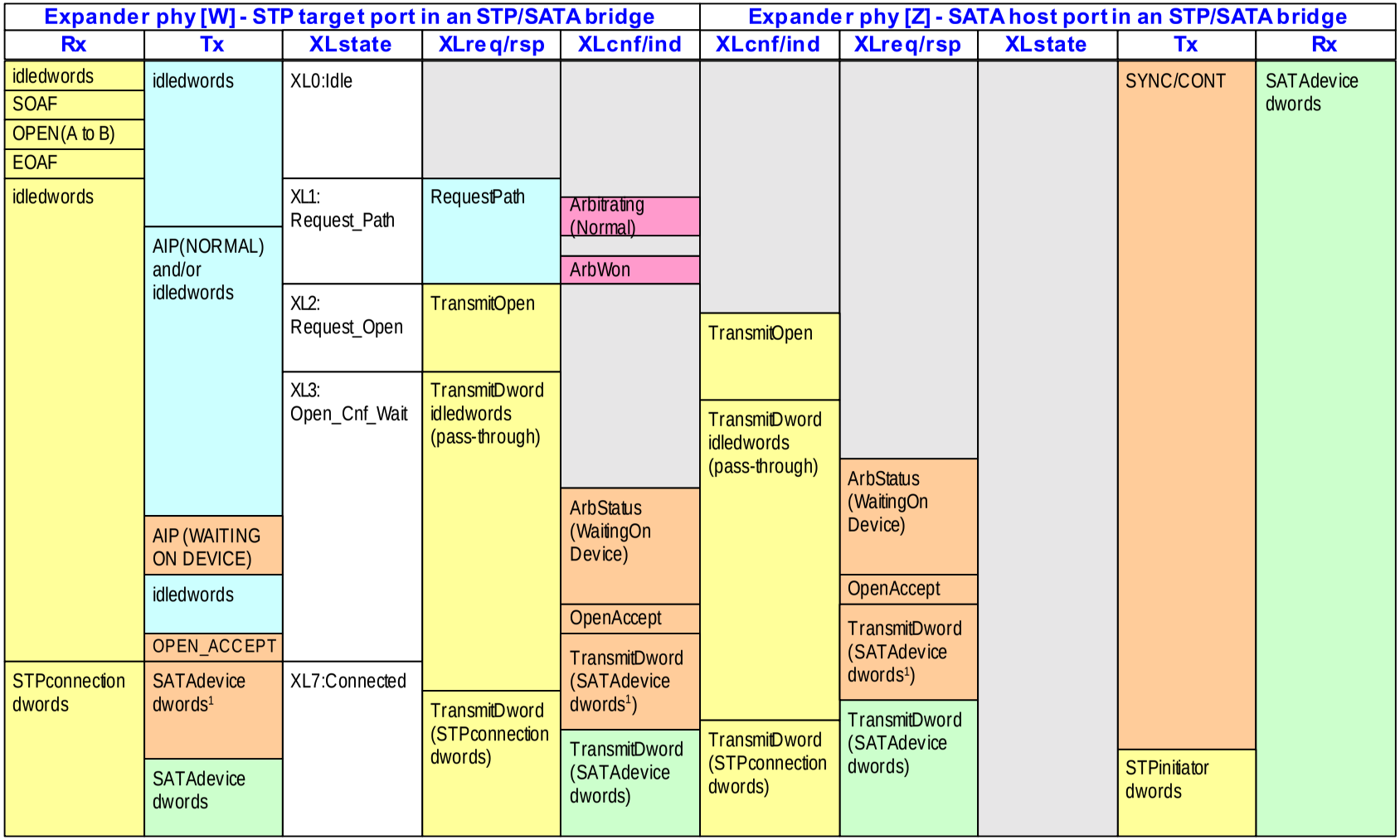


Figure G.11 — BREAK handling during a connection

G.12 STP connection - originated by STP initiator port

Figure 12 shows an STP initiator port originating a connection to an STP target port in an STP/SATA bridge.

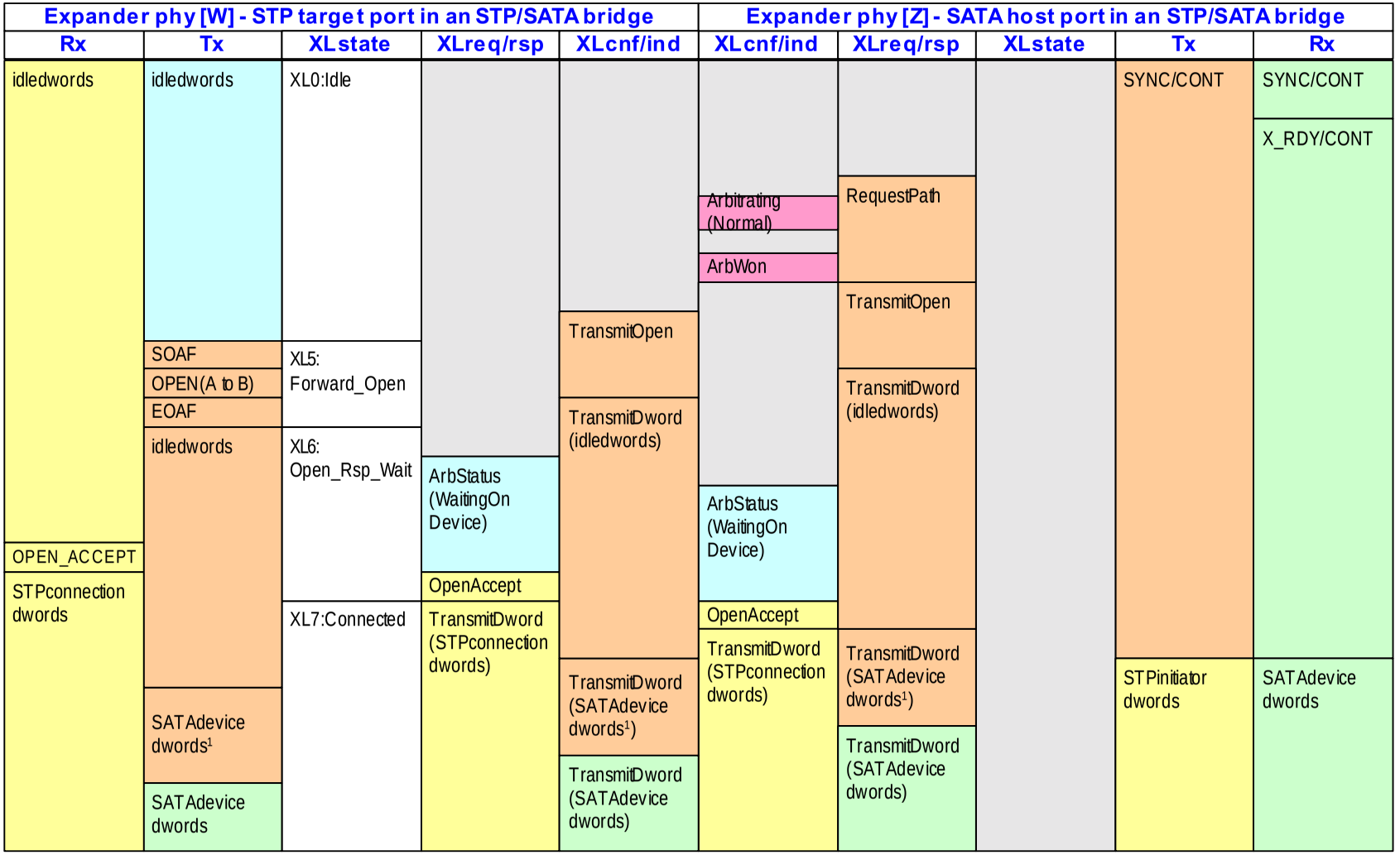


[1] STP/SATA bridge duplicates the dword stream which is being received from the SATA device before forwarding dwords - this ensures that a continued SATA primitive is correctly forwarded to the STP initiator port.

Figure G.12 — STP connection - originated by STP initiator port

G.13 STP connection - originated by STP target port in an STP/SATA bridge

Figure 13 shows an STP target port in an STP/SATA bridge originating a connection on behalf of a SATA device which is requesting to transmit a frame.



[1] STP/SATA bridge duplicates the dword stream which is being received from the SATA device before forwarding dwords - this ensures that a continued SATA primitive is correctly forwarded to the STP initiator port.

Figure G.13 — STP connection - originated by STP target port in an STP/SATA bridge

G.14 STP connection close - originated by STP initiator port

Figure 14 shows an STP initiator port closing a connection to an STP target port in an STP/SATA bridge.

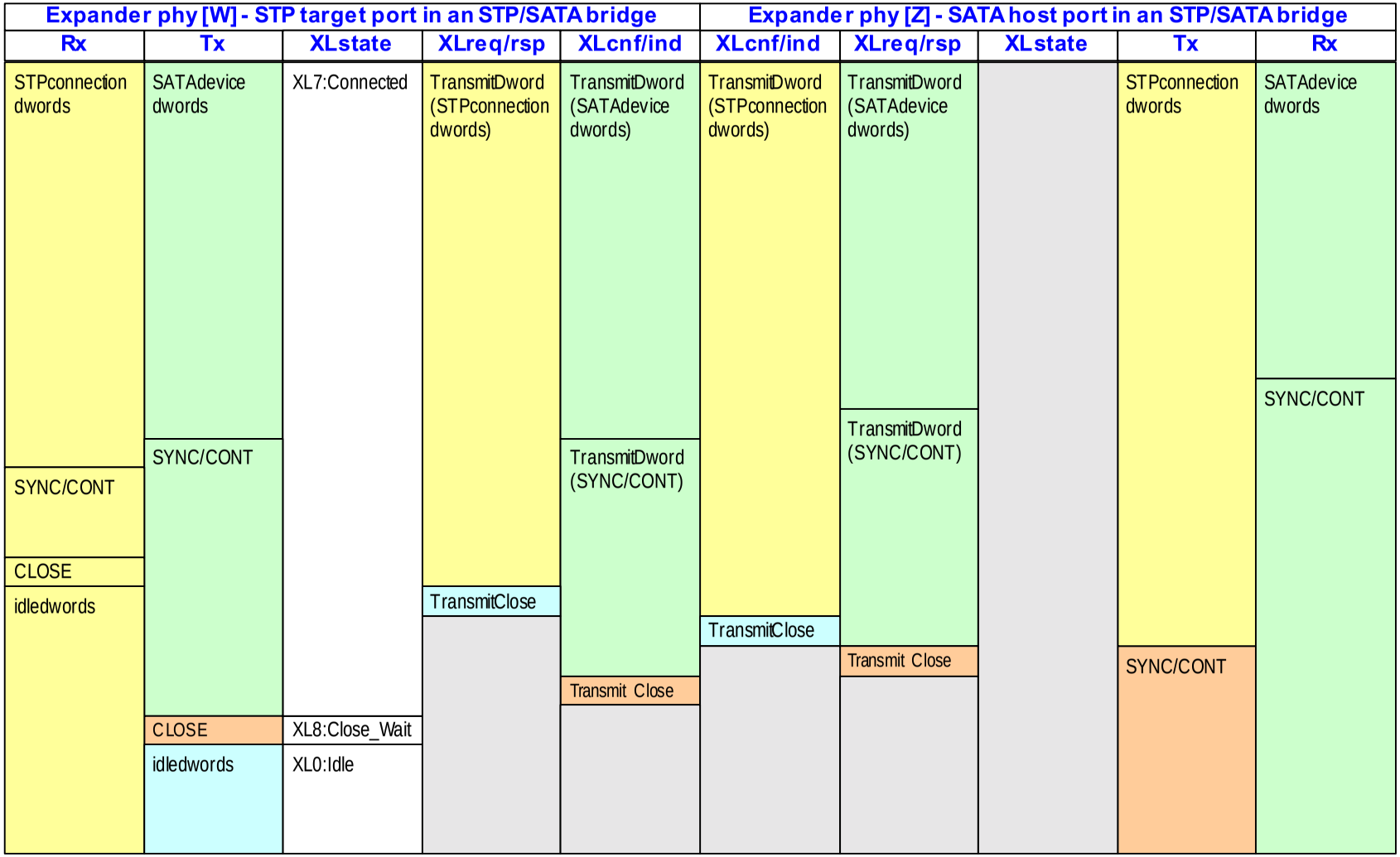


Figure G.14 — STP connection close - originated by STP initiator port

G.15 STP connection close - originated by STP target port in an STP/SATA bridge

Figure 15 shows an STP target port in an STP/SATA bridge closing an STP connection.

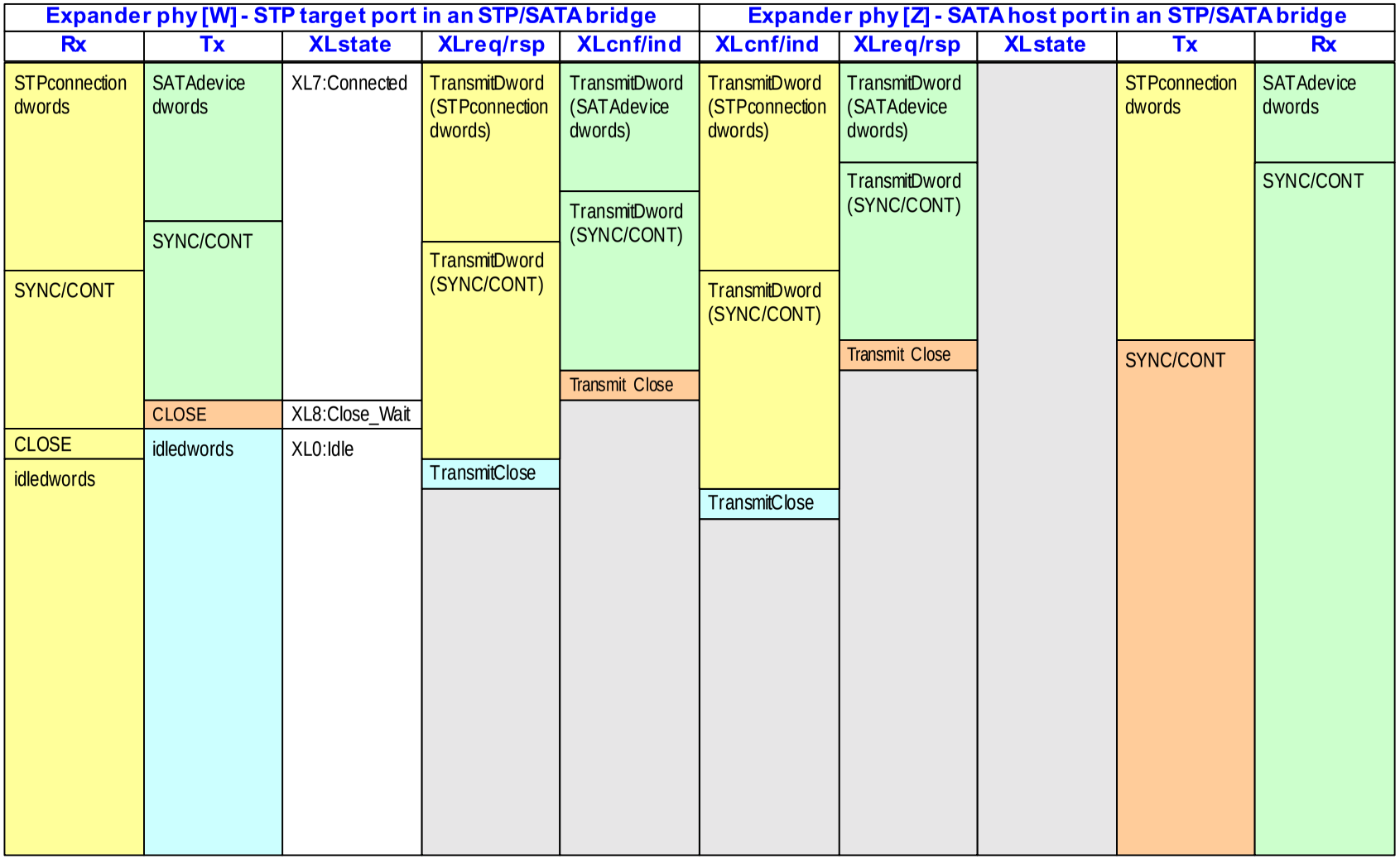


Figure G.15 — STP connection close - originated by STP target port in an STP/SATA bridge

G.X Connection request - XL1:Request_Path to XL5:Forward_Open transition

Figure G.X shows the establishment of a successful connection between two end devices following a XL1:Request_Path to XL5:Foward_Open transition by Expanderphy[Y].

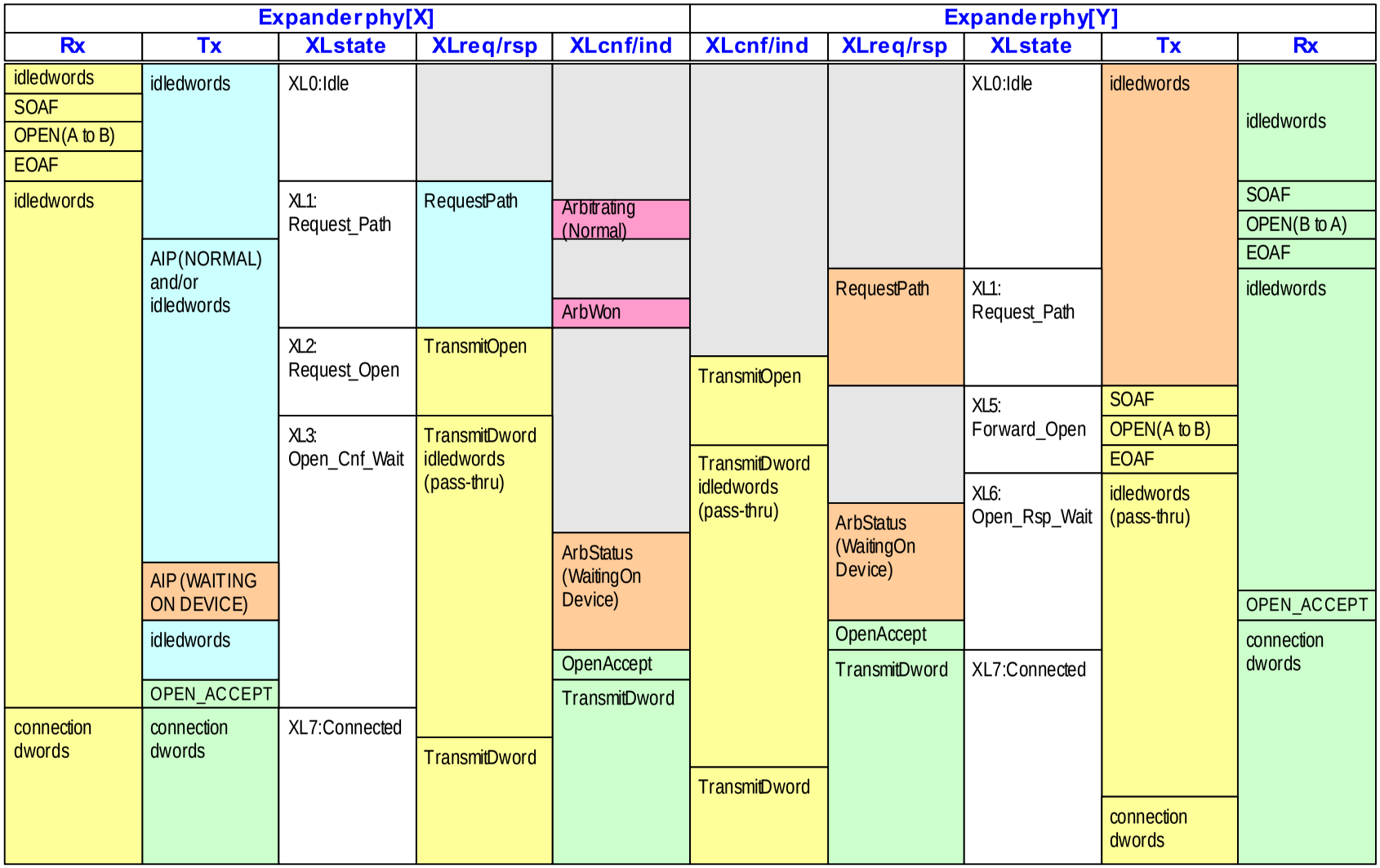


Figure G.X — Connection request - XL1:Request_Path to XL5:Forward_Open transition