



# SuperSpeed USB Developers Conference

Tokyo, Japan  
May 20-21, 2009



# Link Layer

**Sue Vining**

Principal USB Architect  
Texas Instruments



# Agenda

- General Terminology
- SuperSpeed System Topology
- Link Layer Functions
- Link Training & Status State Machine (LTSSM)
- Link Layer Building Blocks
- Packet Construction by Link Layer
- Header Packet Exchange
- Link Errors During Bit Transfer: Detection, Response & Count
- Initialization for HP Integrity & FC after Link Training
- Link Power State Transitions
- Disconnect Detection
- Inband Reset
- Summary

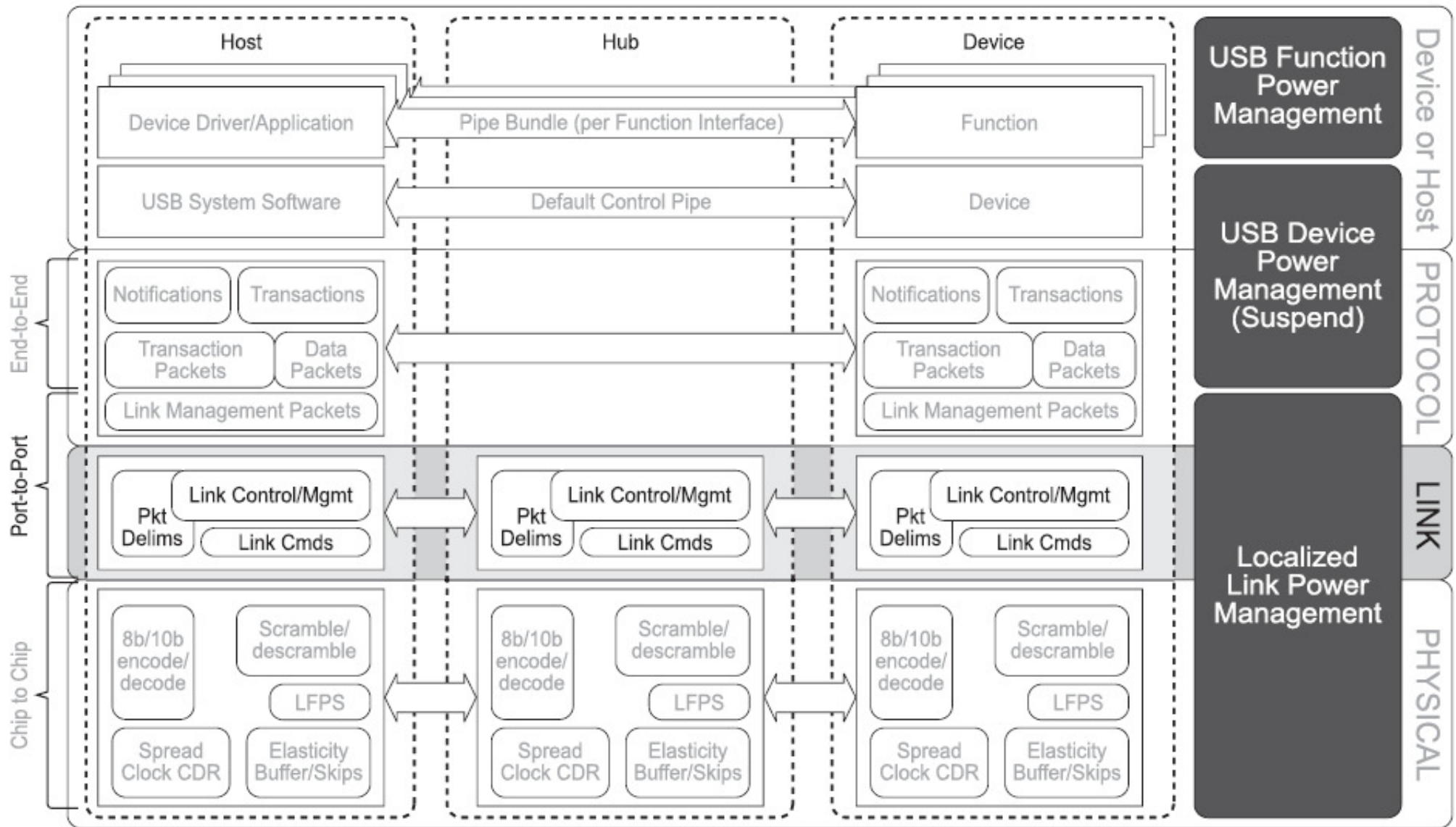


# General Terminology

<b>Symbol</b>	8b/10b encoded control character or data byte
<b>K-symbol</b>	8b/10b encoded control character
<b>D-symbol</b>	8b/10b encoded data byte
<b>Ordered Set</b>	Predefined symbol sequence
<b>Header Packet (HP)</b>	Fixed size packet that contains information consumed by link or host or device
<b>Data Packet (DP)</b>	Compound packet containing header (DPH) and variable size data payload (DPP)
<b>USP</b>	Upstream-facing port
<b>DSP</b>	Downstream-facing port
<b>LFPS</b>	Low frequency period signaling



# SuperSpeed System Topology



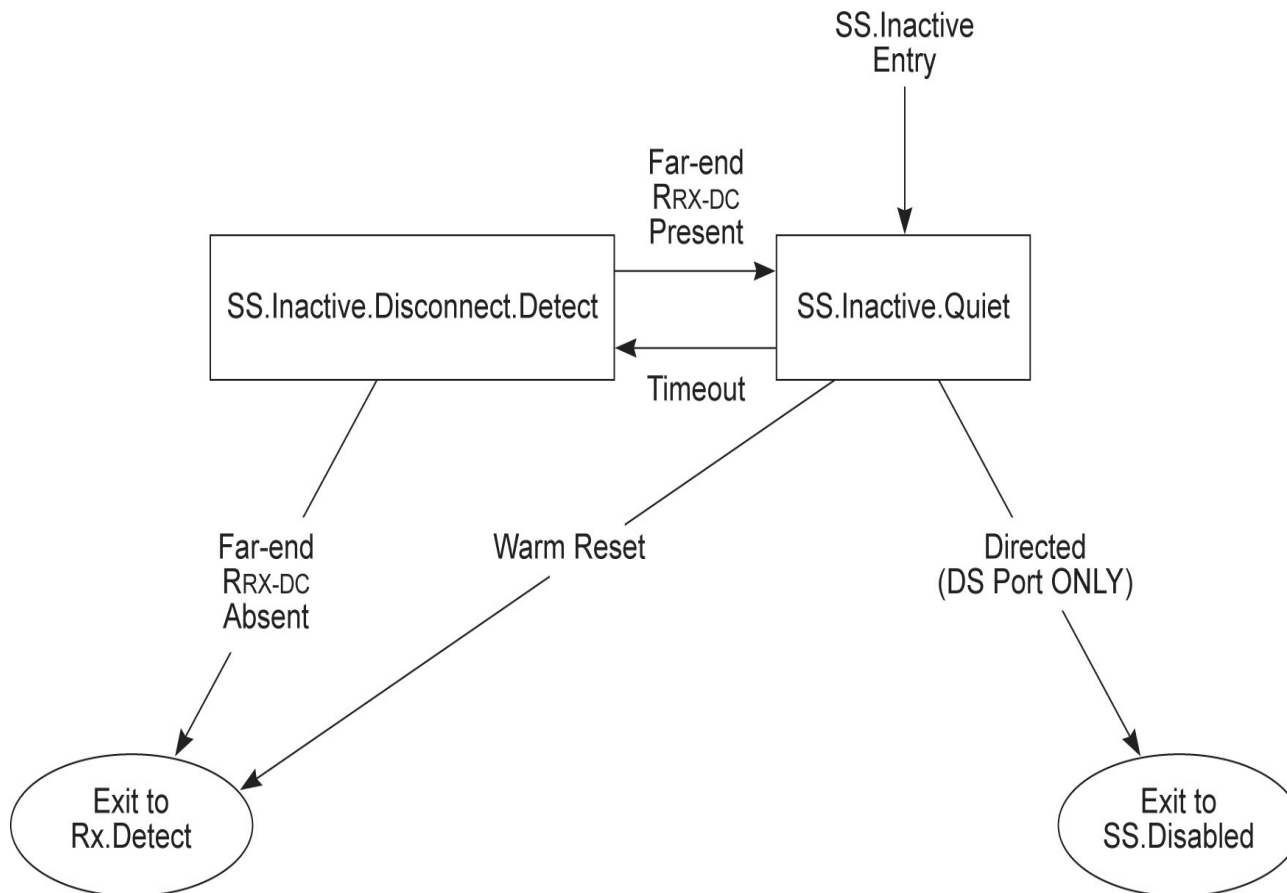


# Link Layer Functions

- Link training control and coordination with link partner
- Build and transmit packets
- Receive and unpack packets
- Ensure reliable header packet exchange
  - Header packet (HP) integrity assurance
  - Header packet flow control (FC)
  - Robust packet framing
- Perform entry & exit sequences for low power states
- Handle link errors during all functions
- Aid disconnect detection by all ports
- Initiate or detect inband reset
- Similar to PCIe, so why USB 3.0 and not cabled PCIe?
  - Class drivers
  - Ubiquity
  - Hot pluggable without sidebands and with relatively small connector



# SS.Inactive



Note: Transition conditions are illustrative only. Not all of the transition conditions are listed.

## Function

- Term resistors present
- DSP or self-powered USP periodically checks for loss of far-end Rx termination

## Purpose

- Software or Warm Reset needed to resolve USB 3.0 error conditions



# SS.Disabled

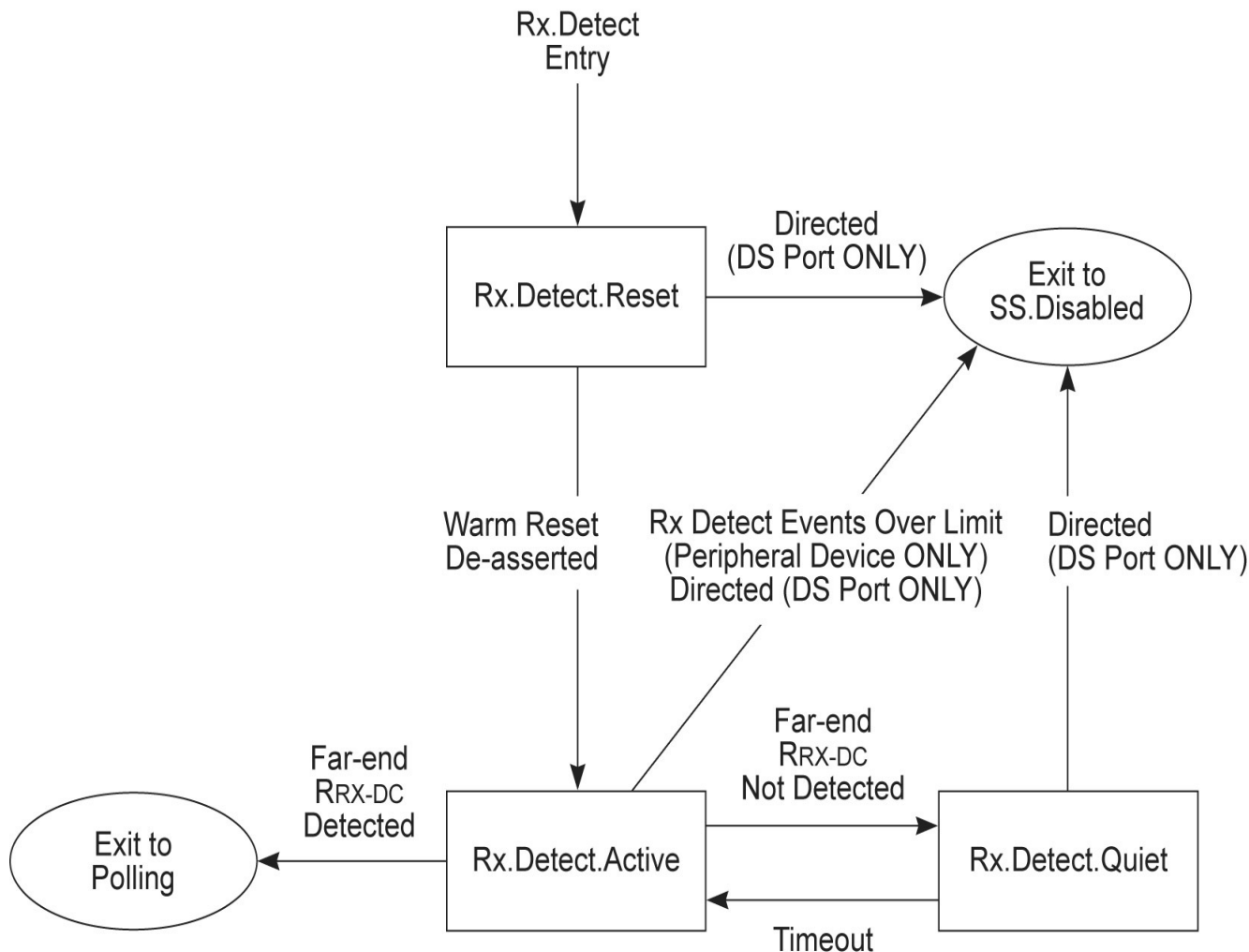
## Function

- Term resistors removed
- Default state for self-powered devices
- DSP exits to Rx.Detect if “directed”
- USP exits to Rx.Detect when
  - VBUS transitions to valid or
  - A USB 2.0 reset is detected

## Purpose

- Allow transition to USB 2.0 for peripheral devices only
- A downstream port will transition to SS.Disabled if the hub’s upstream port does not detect a far-end receiver

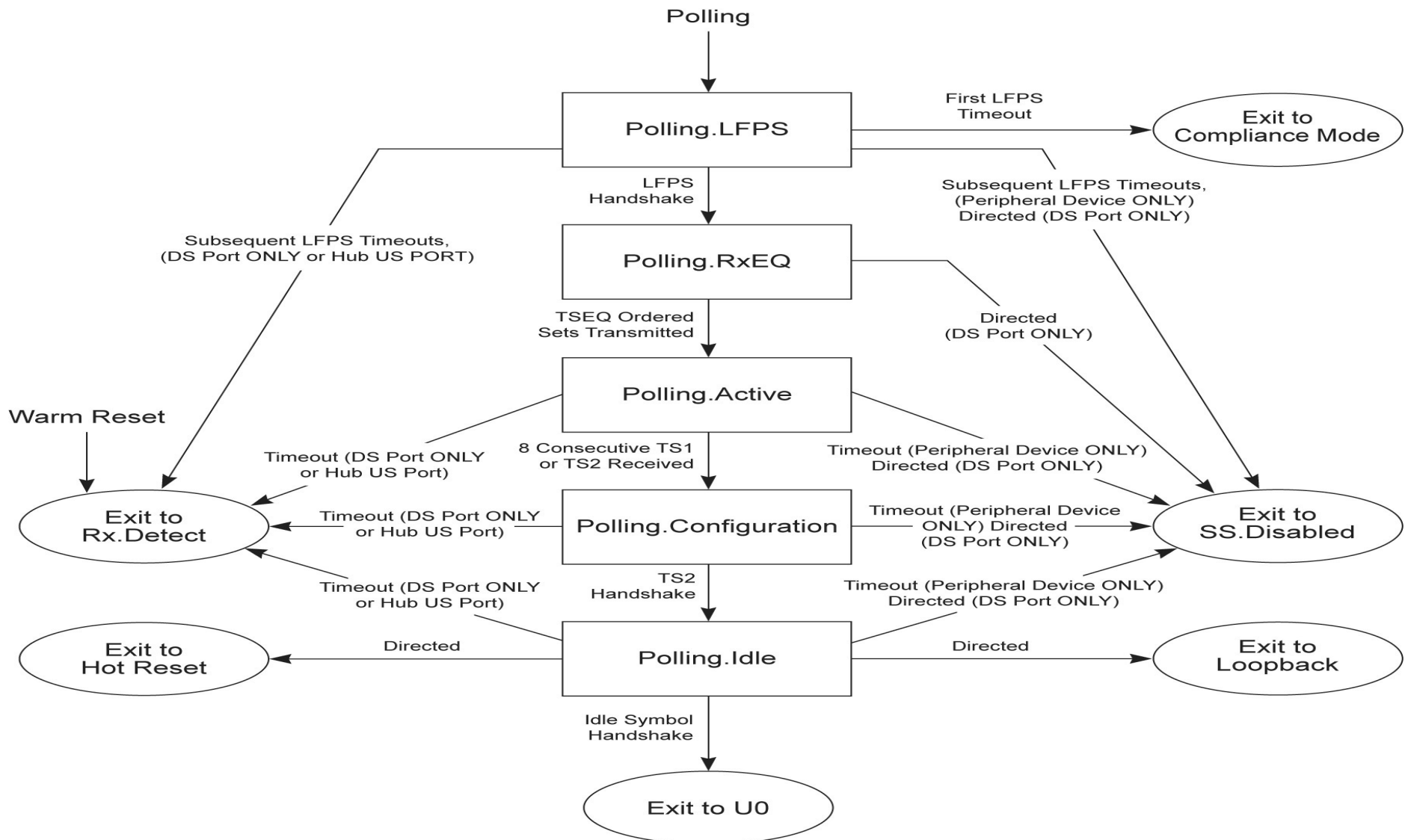
# Rx.Detect



- Initial power-on state for DSP and for USP of bus-powered devices
- Detects far-end termination resistors
- Peripheral USP times out to USB 2.0 (SS.Disabled) after 8 unsuccessful Rx Detection cycles
- Rx.Detect.Reset allows synch of warm reset pulse (USP waits for reset to complete)

Note: Transition conditions are illustrative only. Not all of the transition conditions are listed.

# Polling



Note: Transition conditions are illustrative only. Not all of the transition conditions are listed.

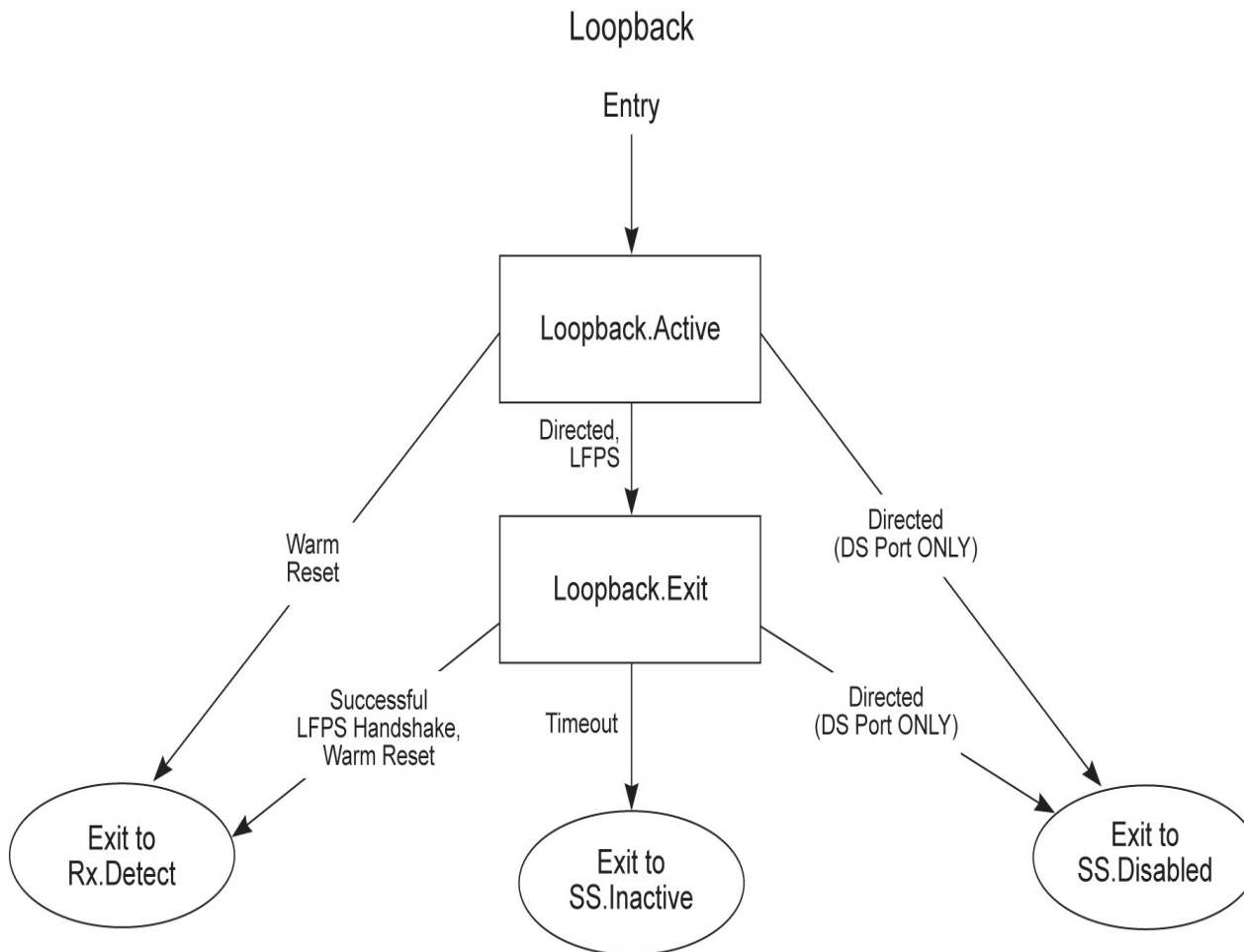
# Polling



## Link training after Rx.Detect

- Polling.LFPS
  - Sync entry to RxEQ by removing up to 12 ms skew from Rx.Detect
  - Establish LFPS DC operating point within 80us,
  - Establish SS DC operating point upon exit
  - Time out to Compliance State when just resistors to ground
- Polling.RxEQ
  - Send TSEQ ordered sets.
  - Train Rx Equalizer
  - Lane Polarity
  - Bit and Symbol Lock
- Polling.Active
  - Send TS1 ordered sets
  - Sync entry to Config from RxEQ
- Polling.Config
  - Send TS2 ordered sets to complete SS Handshake
- Polling.Idle
  - Decode link configuration field & exit to U0

# Loopback



Note: Transition conditions are illustrative only. Not all of the transition conditions are listed.

U-055

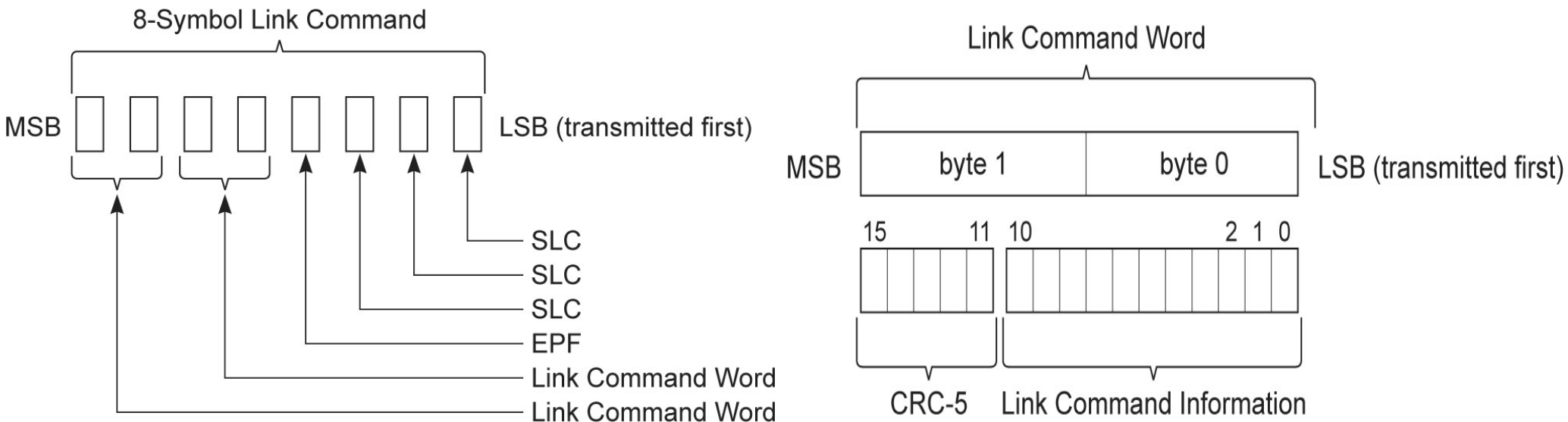
- For bit error rate test (BERT)
- Master initiated “as directed”
- Master function is optional & implementation specific
- Slave function is required
- Two modes
  - Traditional
  - BERT with scrambled logical idle
- Master sets Loopback bit in TS2 to initiate from Recovery or Polling
- Master sends 8b10b with SKPs
- Slave supports the BERT protocol
- Slave responds to BERT commands
- Slave checks the incoming data for the loopback data pattern
- Slave re-transmits the 8b10b but with
  - Lane polarity inversion (if necessary)
  - Clock tolerance compensation (as necessary)
- Exits to Rx.Detect

# Link Layer Building Blocks: Link Commands



- Link commands (LCs) enable all link layer functions other than link training control and inband reset
- Robust design of link commands
  - LCSTART is a 4-symbol ordered set
    - Start of link command word is deterministic with any 3 symbols
    - No end delimiter is needed
    - Bogus LCs not created due to even 2 corrupted symbols
  - The link command word is sent twice
- Detection of missing LCs is ensured by
  - Timers, with fixed timeout values defined in standard
  - Sequence checks

# Link Layer Building Blocks: Link Command Structure



U-045

U-046

- Fixed 8-Symbol size:
  - 4 K-Symbols at start followed by two-byte Link Command Word(LCMD) repeated
  - Valid receipt: 3 of 4 K-Symbols & either both LCMDs are valid & identical or one LCMD is valid & the other is invalid
- Link Command Word:
  - Total content: 2 scrambled bytes
  - 5-bit CRC-5 protects 11-bit link command information

# Link Layer Building Blocks: List of Link Commands



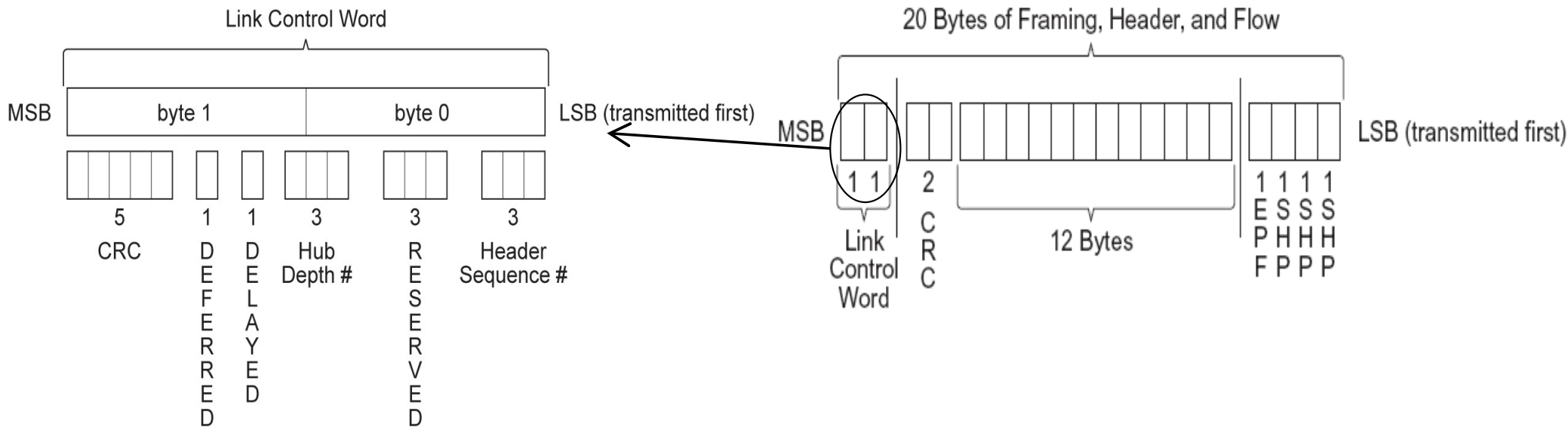
- **HP Integrity** – LGOOD\_n (n=0 to 7), LBAD, LRTY
- **HP Flow Control** – LCRD\_x (x=A,B,C,D)
- **Link Power Management** – LGO\_U1, LGO\_U2, LGO\_U3, LAU, LXU, LPMA
- **Announce Presence in U0** – LUP, LDN
  - DSP: detect disconnect by loss of LUP
  - USP: [ ECN] detect disabled by loss of LDN



# Link Layer Building Blocks: HP Structure for Integrity Assurance

- Robust HP framing
  - 4-symbol ordered set, HPSTART, similar to LCSTART
  - No end framing needed or used
- HPs have 2 CRCs & a header sequence number (HSEQ#)
  - HSEQ# is independent of the transaction layer data sequence number and ranges from 0 to 7
  - Correct receipt of HPs is checked using the 2 CRCs
  - Correct receipt of HP is *acknowledged* using the LGOOD\_n link command, where n=Rx HSEQ#
  - Incorrect receipt of HP is *acknowledged* using LBAD link command
  - Every HP must be *acknowledged* with LGOOD\_n or LBAD
- Missing HPs detectable by a skipped HSEQ#
- Every receiver must advertise its expected HSEQ# at entry to U0
  - Advertised HSEQ# must be accepted by link even during Polling.Idle or Recovery.Idle

# Link Layer Building Blocks: HP Structure for Integrity Assurance



U-040

## Link Control Word:

- Integrity assurance
  - Header Sequence #
  - CRC-5: for other 11 bits
- Hub forwarding support
  - Hub Depth, Delayed, Deferred

## Header Packet:

- 20 symbol structure
  - 4 K-symbol start framing
  - 12 Byte end-to-end header content
  - CRC-16 over the 12 Byte header
  - 2 Byte *Link Control Word*

# Packet Construction by Link Layer



- Link Layer constructs HPs by adding
  - Link Control Word (may be updated by link layer)
    - Header sequence # is unique to each link
    - Hub forwarding support values must be maintained, if set
  - CRC-16, if not added by higher layer
  - HPSTART (framing)
- Link Layer constructs data packets (DPs) by
  - Constructing header (DPH) as defined above
  - Constructing data payload (DPP) by adding start (DPPSTART) and end (DPPEND or DPPABORT) framing  
CRC is added by higher layer
  - Combining DPH and DPP back-to-back (no gap)
- No gap is required between packets sent by hubs or peripherals

# HP Exchange: Link Flow Control Background



- Link flow control (FC) is only for HPs, not data payload
- Every receiver must advertise its 4 initial credits (Local Rx Header Buffer Credit) to its partner's transmitter using LCRD\_x link commands
- Transmitter needs at least 1 credit from partner (Remote Rx Header Buffer Credit) to send a header packet
- Credit is returned whenever a packet is processed, independently of when it was received relative to other packets
  - LCRD\_x index(x:A,B,C,D) is solely to detect missing LCRD\_x
  - Delayed processing of any packet does not prevent credit return for subsequent packets
- Each LCRD\_x always returns exactly 1 header packet credit

# HP Exchange: Terminology



## **Link partner**

Device at one end of link

## **HP Rx**

Link partner receiving an HP

## **Rx Header Buffer**

Received & unprocessed HP storage

## **Local Rx Header Buffer Credit**

Available space to receive HP's in Rx Header Buffer

## **Rx Header Sequence Number**

Expected in next received HP

## **Rx LCRD\_x index**

Assigned to next LCRD\_x sent

## **HP Tx**

Link partner transmitting a header packet

## **Tx Header Buffer**

Transmitted & unacknowledged HP storage

## **Remote Rx Header Buffer Credit**

Credit to transmit HP's

## **Tx Header Sequence Number**

Assigned to next new HP

## **ACK Tx Header Sequence Number**

Points to oldest HP in Tx Header Buffer

## **Remote Rx LCRD\_x index**

Expected index of next received LCRD\_x

## **PENDING\_HP\_TIMER**

3 us timer to check for missing acknowledgment

## **CREDIT\_HP\_TIMER**

5 ms timer to check for missing credit return



# HP Exchange: Initial Conditions

- The initial conditions for the following illustration of HP Exchange happen to match the reset state but could be random
- Initial and final exchange conditions of an idle state were chosen to allow a simpler sequence
  - All buffers are empty
  - No timers are running



# HP Exchange: No Errors

## Link Partner HP Tx

### Tx Header Buffer

Empty
Empty
Empty
Empty

Tx Header Sequence Number = 0

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 4

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = 0, IDLE

## Link Partner HP Rx

### Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: No Errors

## Link Partner HP Tx

Tx Header Buffer

<b>HP, HSEQ#=0</b>
Empty
Empty
Empty

Tx Header Sequence Number = **1**

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 4

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = 0, IDLE

## Link Partner HP Rx

Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: No Errors

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = **3**

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = **running**

CREDIT\_HP\_TIMER = **running**

Framed HP  
HSEQ#=0

## Link Partner HP Rx

Rx Header Buffer

<b>HP, HSEQ#=0</b>
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: No Errors

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = running

CREDIT\_HP\_TIMER = running

Framed HP  
HSEQ#=0

## Link Partner HP Rx

Rx Header Buffer

HP, HSEQ#=0, <b>good CRC's</b>
Empty
Empty
Empty

Rx Header Sequence Number = **1**

Local Rx Header Buffer Credit = **3**

Rx LCRD\_x index = A



# HP Exchange: No Errors

## Link Partner HP Tx

Tx Header Buffer

<b>Empty</b>
Empty
Empty
Empty

Tx Header Sequence Number = 1

ACK Tx Header Sequence Number = **1**

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = **0, IDLE**

CREDIT\_HP\_TIMER = running

Framed HP  
HSEQ#=0

**LG00D\_0**

## Link Partner HP Rx

Rx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Rx Header Sequence Number = 1

Local Rx Header Buffer Credit = 3

Rx LCRD\_x index = A



# HP Exchange: No Errors

## Link Partner HP Tx

Tx Header Buffer

Empty
Empty
Empty
Empty

Tx Header Sequence Number = 1

ACK Tx Header Sequence Number = 1

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = running

Framed HP  
HSEQ#=0

LG00D\_0

## Link Partner HP Rx

Rx Header Buffer

<b>Processing complete, Empty</b>
Empty
Empty
Empty

Rx Header Sequence Number = 1

Local Rx Header Buffer Credit = **4**

Rx LCRD\_x index = A



# HP Exchange: No Errors

## Link Partner HP Tx

Tx Header Buffer

Empty
Empty
Empty
Empty

Tx Header Sequence Number = 1

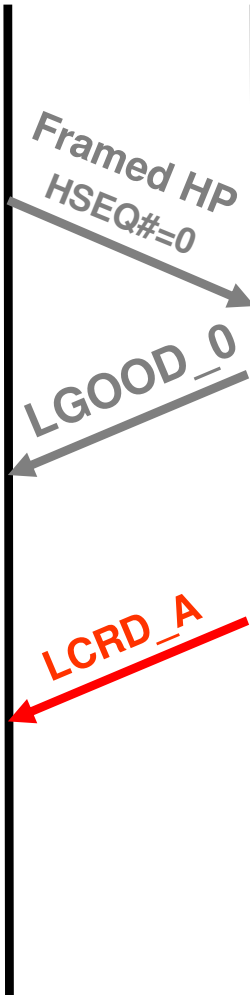
ACK Tx Header Sequence Number = 1

Remote Rx Header Buffer Credit = **4**

Remote Rx LCRD\_x index = **B**

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = **0, IDLE**



## Link Partner HP Rx

Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 1

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = **B**



# HP Exchange: Retry

- The following sequence has both
  - An initial header packet receive error and
  - A successful retry using a sequence of
    - Header packet transmit,
    - LBAD,
    - LRTY,
    - Header packet retransmit,
    - LGOOD\_n and
    - LCRD\_x



# HP Exchange: Retry

## Link Partner HP Tx

### Tx Header Buffer

Empty
Empty
Empty
Empty

Tx Header Sequence Number = 0

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 4

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = 0, IDLE

## Link Partner HP Rx

### Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

<b>HP, HSEQ#=0</b>
Empty
Empty
Empty

Tx Header Sequence Number = **1**

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 4

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = 0, IDLE

## Link Partner HP Rx

Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = **3**

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = **running**

CREDIT\_HP\_TIMER = **running**

Framed HP  
HSEQ#=0

## Link Partner HP Rx

Rx Header Buffer

<b>HP, HSEQ#=0</b>
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = running

CREDIT\_HP\_TIMER = running

Framed HP  
HSEQ#=0

## Link Partner HP Rx

Rx Header Buffer

HP, HSEQ#=0, <b>bad CRC</b>
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

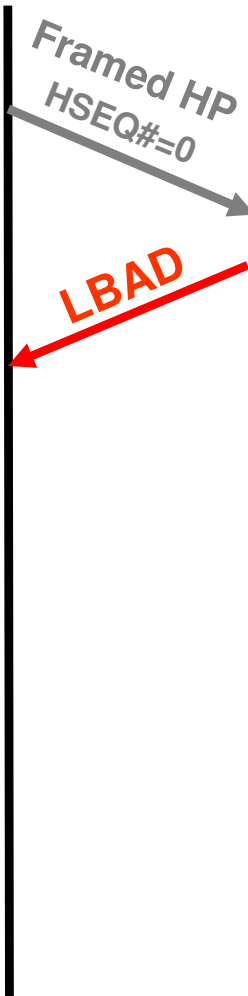
ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = 0, IDLE



## Link Partner HP Rx

Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

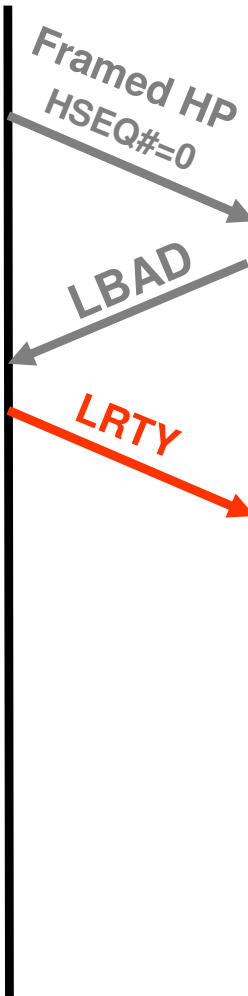
ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = 0, IDLE



## Link Partner HP Rx

Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

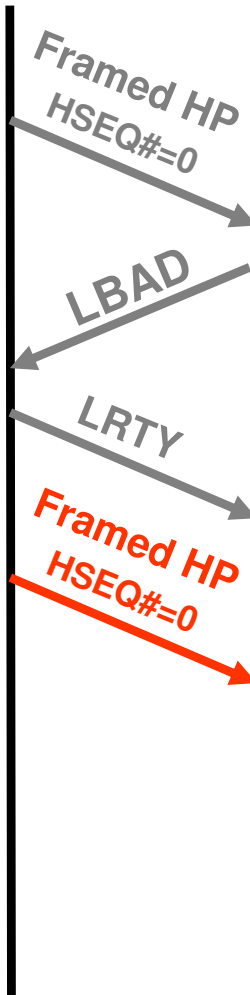
ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = **running**

CREDIT\_HP\_TIMER = **running**



## Link Partner HP Rx

Rx Header Buffer

<b>HP, HSEQ#=0</b>
Empty
Empty
Empty

Rx Header Sequence Number = 0

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Tx Header Sequence Number = 1

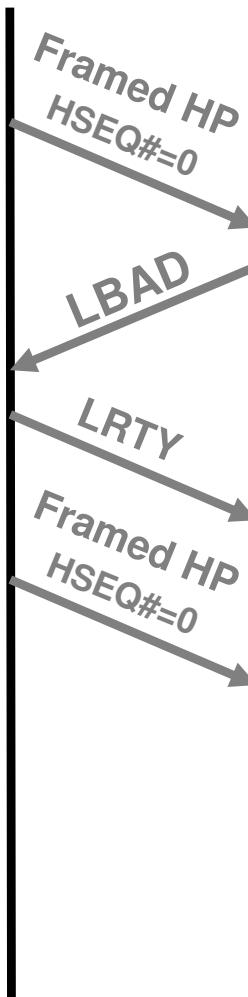
ACK Tx Header Sequence Number = 0

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = running

CREDIT\_HP\_TIMER = running



## Link Partner HP Rx

Rx Header Buffer

HP, HSEQ#=0, <b>good CRC's</b>
Empty
Empty
Empty

Rx Header Sequence Number = **1**

Local Rx Header Buffer Credit = **3**

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

<b>Empty</b>
Empty
Empty
Empty

Tx Header Sequence Number = 1

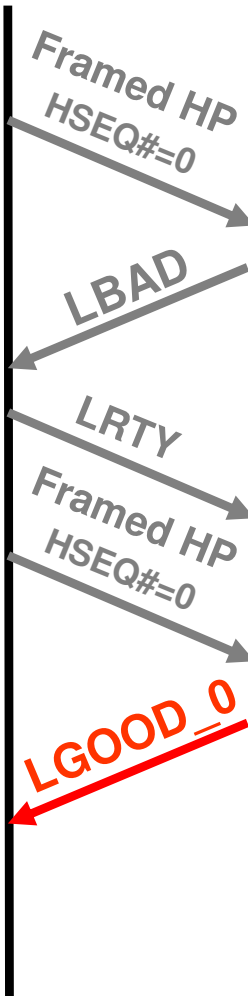
ACK Tx Header Sequence Number = **1**

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = **0, IDLE**

CREDIT\_HP\_TIMER = running



## Link Partner HP Rx

Rx Header Buffer

HP, HSEQ#=0
Empty
Empty
Empty

Rx Header Sequence Number = 1

Local Rx Header Buffer Credit = 3

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

Empty
Empty
Empty
Empty

Tx Header Sequence Number = 1

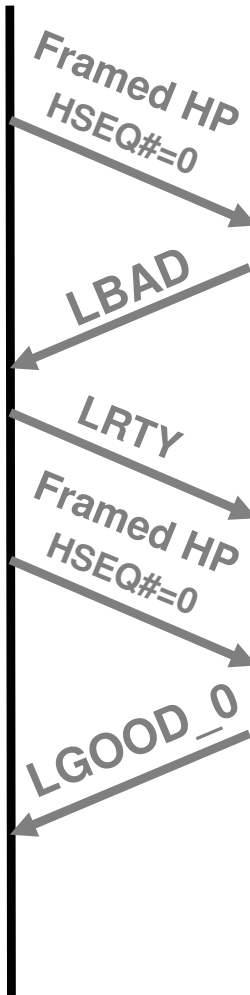
ACK Tx Header Sequence Number = 1

Remote Rx Header Buffer Credit = 3

Remote Rx LCRD\_x index = A

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = running



## Link Partner HP Rx

Rx Header Buffer

<b>Processing complete, Empty</b>
Empty
Empty
Empty

Rx Header Sequence Number = 1

Local Rx Header Buffer Credit = **4**

Rx LCRD\_x index = A



# HP Exchange: Retry

## Link Partner HP Tx

Tx Header Buffer

Empty
Empty
Empty
Empty

Tx Header Sequence Number = 1

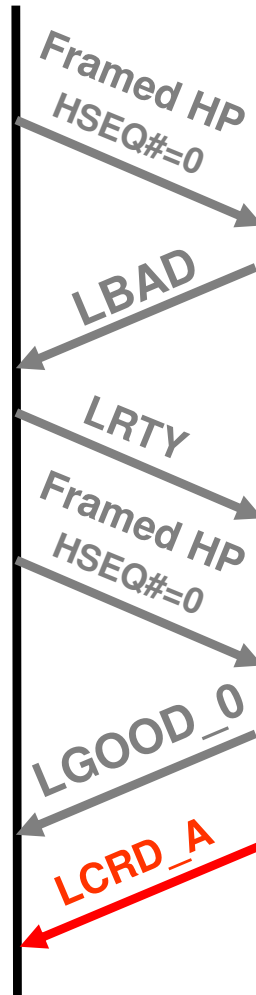
ACK Tx Header Sequence Number = 1

Remote Rx Header Buffer Credit = **4**

Remote Rx LCRD\_x index = **B**

PENDING\_HP\_TIMER = 0, IDLE

CREDIT\_HP\_TIMER = **0, IDLE**



## Link Partner HP Rx

Rx Header Buffer

Empty
Empty
Empty
Empty

Rx Header Sequence Number = 1

Local Rx Header Buffer Credit = 4

Rx LCRD\_x index = **B**

# Link Errors During Bit Transfer: Detection, Response & Count



- Packet Framing Errors:
  - Receiver ignores insufficiently framed packets
  - Detected by Pending\_HP\_Timer – Go to Recovery
- Header Sequence Number Error:
  - Get a Header Sequence Number that is not expected – Go to Recovery
  - Possible consequence of packet framing error observed on subsequent HP
- Packet CRC Errors:
  - Header Packet is Retried up to 2 times
  - Header Packet has CRC Error on 3rd attempt – Go to Recovery
- Training Sequence Error:
  - Timeout from Polling or Recovery
  - Timeout from Recovery is a serious error that requires SW intervention to fix
- Missing Link Command:
  - Go to Recovery if not LPMA
- Corrupt Link Command:
  - Ignore
- Link Error Count is incremented by the downstream port each time link enters Recovery due to an error

# Initialization for HP Integrity & FC After Link Training



Advertise expected Rx Header Sequence Number

- Send LGOOD\_n, where
  - n = 7 after reset or
  - n = HSEQ# of last good received HP otherwise
- Receive LGOOD\_n from link partner
  - Flush the HPs from Tx Header Buffer with HSEQ#  $\leq$  n (considering rollover)
  - Set ACK Tx Header Sequence Number to n+1 (with rollover)

Advertise Local Rx Header Buffer Credit

- Send credit for all available buffers using 1 LCRD\_x for each
  - First credit is always returned using LCRD\_A
  - Must advertise all 4 credits after reset.
- Receive first credit (LCRD\_A)
  - May now send a packet
- Receive remaining credits (LCRD\_B-D)
  - May now initiate power state transitions

# Link Power State Transitions: Background



- Higher layer conditions for initiating, accepting or rejecting U1 or U2 are mainly defined elsewhere
- U1 and U2 can be requested by either side of a link
  - U2 can also be entered by timeout from U1
- U3 is initiated only by software request to a downstream port
  - U3 must be accepted
- Entry requests can be sent and received concurrently
  - Request by DSP takes precedence
- Entry sequence completed through link commands
  - LGO\_U1, LGO\_U2, or LGO\_U3 is sent to request a low power link state
  - LAU or LXU is sent to accept or reject a request respectively
  - LPMA is sent to confirm receipt of LAU and resulting entry to U1, U2 or U3
  - Defined to ensure that both ends of the link are in the same power state
- Exit from U1/U2 and wake from U3 by LFPS handshake

# Link Power State Transitions: Rules and Terms



## Rules for a port to request or accept a low power link state

- It has transmitted an LGOOD\_n, LCRD\_x sequence for all packets received
- It has received an LGOOD\_n, LCRD\_x sequence for all packets transmitted
- It has no pending packets for transmission
- It is permitted to request or accept by the higher layer
- It must be directed by a higher layer to request a low power link state

## Terms

**PM\_LC\_TIMER:** Runs while waiting 3us for response (LAU or LXU)

**PM\_ENTRY\_TIMER:** Runs while waiting 6us for LPMA or TS1 for Recover

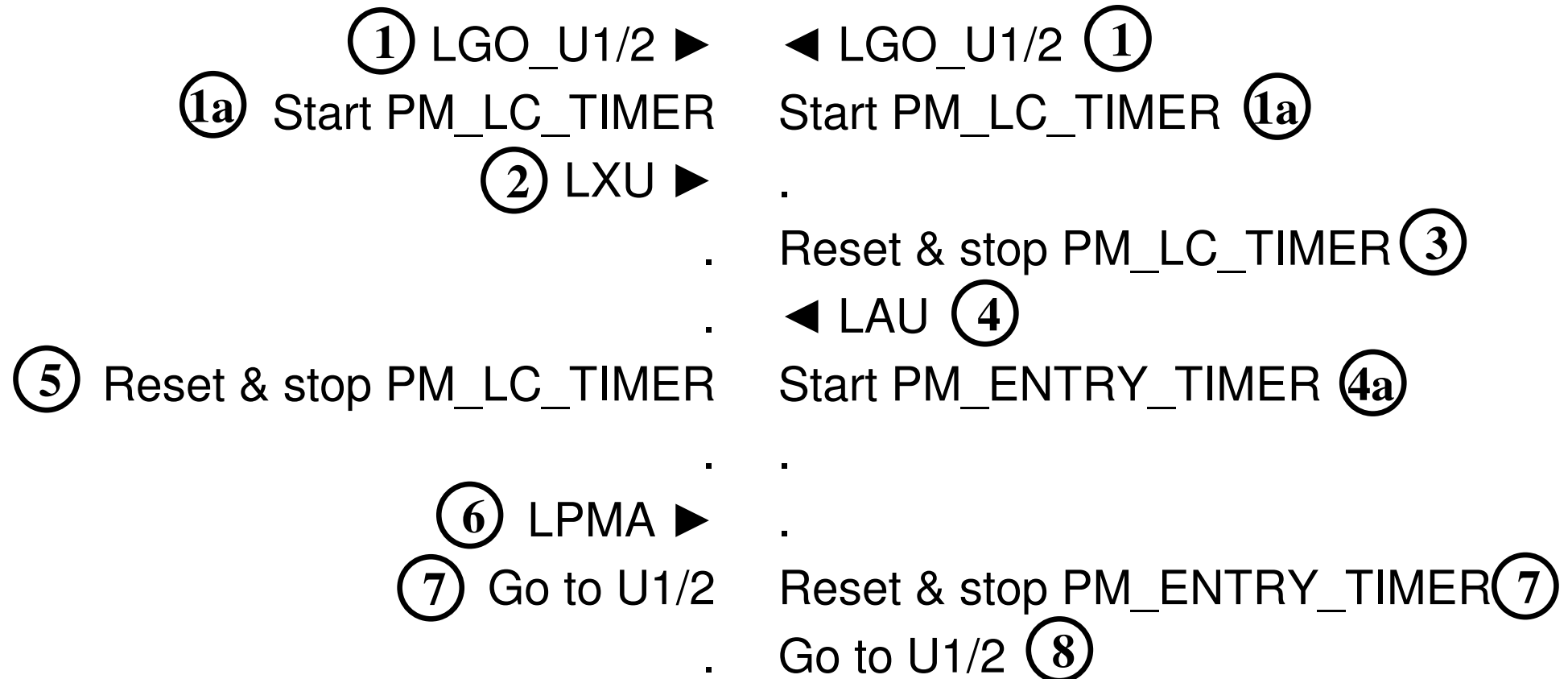
**U2 inactivity timer:** Runs while link is idle to qualify entry to U2  
Timeout value set by USB system software

# Link Power State Transitions: Concurrent Requests



## DSP

## USP



# Link Power State Transitions: Single Rejected U1 or U2 Request



## Link "a"

## Link "b"

② LXU ►

- . ◀ LGO\_U1/2 ①
- . Start PM\_LC\_TIMER ①a
- . .
- . Reset & stop PM\_LC\_TIMER ③
- . Stay in U0 ③a
- . .
- . .
- . .
- . .
- . .
- . .

# Link Power State Transitions: Corrupted LAU



## Link "a"

- ① LGO\_U1/2/3 ►
- ①a Start PM\_LC\_TIMER .
- ③ LAU is corrupted & lost .
- ③a PM\_LC\_TIMER expires .
- ④ Send TS1 to initiate Recovery .
- ⑦ Complete Recovery .
- ⑧ Re-request U1/2/3

## Link "b"

- . .
- ◀ LAU, ②
- Start PM\_ENTRY\_TIMER ②a
- . .
- . .
- . .
- Reset & stop PM\_ENTRY\_TIMER ⑤
- Join Recovery ⑥
- Complete Recovery ⑦
- . .

# Link Power State Transitions: Corrupted LPMA



## Link "a"

- ① LGO\_U1/2/3 ►
- ①a Start PM\_LC\_TIMER
- .
- .
- ③ Reset & stop PM\_LC\_TIMER
- ④ LPMA ►
- ⑤ Go to U1/2/3
- .
- .
- .
- .

## Link "b"

- .
- ◄ LAU, ②
- Start PM\_ENTRY\_TIMER ②a
- .
- .
- .
- .
- LPMA is corrupted & lost ⑤
- PM\_ENTRY\_TIMER expires ⑤a
- Go to U1/2/3 ⑥
- .
- .

# Link Power State Transitions: Direct Entry to U2 from U1



## Link "b"

① Go to U1

①a Reset & start U2 inactivity timer

①b U2 inactivity timer expires

② Go to U2

.  
. .  
. .  
. .

## Link "b"

Go to U1 ①

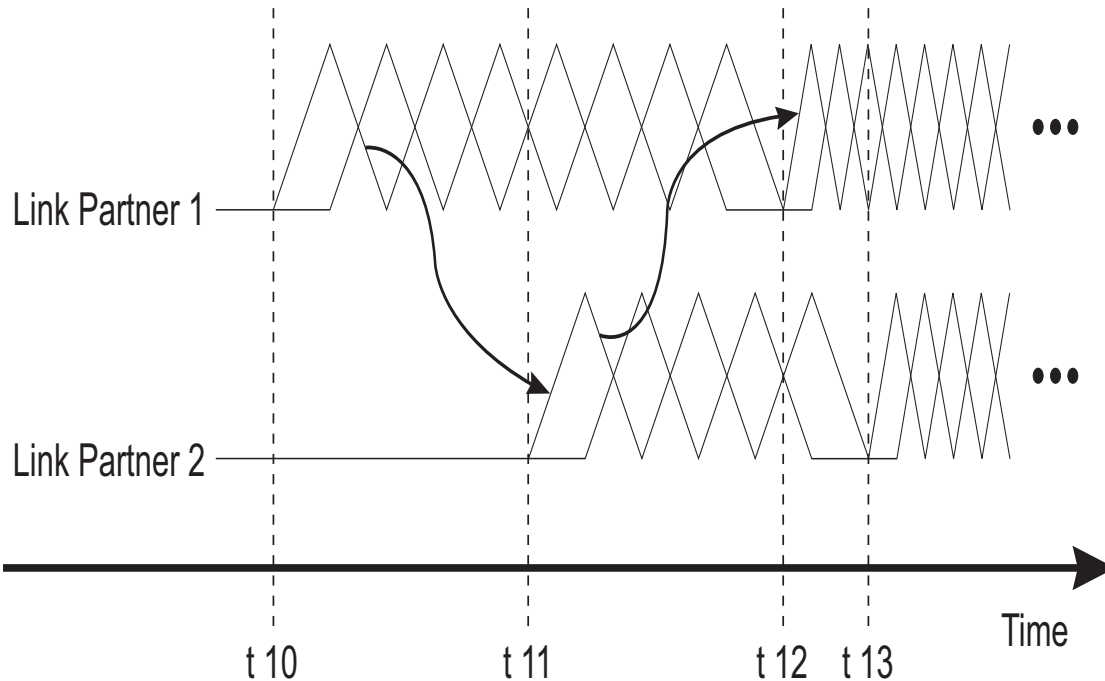
Reset & start U2 inactivity timer ①a

U2 inactivity timer expires ①b

Go to U2 ②

.  
. .  
. .  
. .

# Link Power State Transitions: U1 Exit, U2 Exit, U3 Wakeup



U-033

## LFPS handshake sequence

- Link partner 1 initiates exit by transmitting LFPS at time  $t_{10}$
- Link partner 2 detects valid LFPS on its receiver and responds by transmitting LFPS at time  $t_{11}$
- Link partner 1 enters Recovery and starts transmitting TS1 Ordered Sets at time  $t_{12}$
- Link partner 2 enters Recovery and starts transmitting TS1 Ordered Sets at time  $t_{13}$

# Disconnect Detection



From U0 (active state)

- LUP/LDN Link Command “ping” sent by Upstream/Downstream Port once every 10 us
  - Downstream/Upstream Port (DSP/USP) goes to Recovery if it doesn't see a link command within 1ms. From Recovery, it would go to SS.Inactive and then Rx.Detect. An Upstream Port would end in SS.Disabled, if Rx.Detect failed 8 times, and the device would switch to USB 2.0

From U1 (fast-exit reduced-power idle link state)

- Upstream Port sends a Ping.LFPS every 200 ms
  - Can't use Rx.Detect – must maintain common mode to reduce U1 exit latency
  - DSP goes to Rx.Detect if doesn't get one Ping.LFPS per 300 ms

From U2 (slow-exit reduced-power idle link state) or U3 (suspend)

- Downstream Port performs an Rx.Detect every 100 ms
  - Common mode is disturbed – 64 us to restore is OK for U2, U3 exit latencies
  - DSP remains in Rx.Detect if it doesn't detect the appropriate Rx termination

# Inband Reset



- Inband Reset relies on SuperSpeed or LFPS signaling, depending on type
- Two types of inband reset:
  - Hot Reset – A reset bit within TS2 Ordered Sets (SuperSpeed signaling)
  - Warm Reset – “Big Hammer”– 100 ms LFPS meeting tReset requirements
- Inband reset is initiated by software request to downstream port
- Two types of reset request
  - PORT\_RESET: LTSSM decides which inband reset based on link states
    - LTSSM = U0, Polling, Recovery – use Hot Reset
    - LTSSM = U1, U2 – exit to Recovery – use Hot Reset
    - LTSSM = U3, SS.Inactive, Loopback, Compliance – use Warm Reset
    - LTSSM = SS.Disabled – reset is prohibited
    - If Hot Reset fails at Recovery – Go to Rx.Detect – then do a Warm Reset
  - BH\_PORT\_RESET: LTSSM must issue warm reset



# Summary

- Robust and reliable delivery of HPs
- Enables receiver detection by phy
- Controls and coordinates link training with link partner
- Completes entry to and exit from low power link states
- Ensures both link partners are in the same power state
- Signals or detects inband reset

# Backup: LUP for Disconnect Detection During U0



- During U0,
  - The downstream port does not check Rx termination
  - It can detect a disconnect only by lack of traffic
    - To ensure regular traffic during long periods of logical idle in U0 when U1 or U2 is either not enabled or not possible, LUP was defined
    - It is intentionally unique to an upstream port to allow the opportunity to distinguish between active USP Tx and crosstalk (NEXT) from DSP Tx to DSP Rx while Rx is electrically idle due to disconnect
- LUP is sent every 10 us
  - And not more frequently, to limit its impact on throughput
  - And not less frequently, to ensure that it is seen within the 1ms window even if the timer expiration is very loose and 1 or more of these LUPs are missed
  - Transition to Recovery, then to SS.Inactive and then to Rx.Detect is to enable distinction between link error and disconnect