USB4™ Logical Layer, Re-timer and Transport Layer

Uri Hermoni – Architect, Intel Corporation
Gal Yedidia – Architect, Intel Corporation

USB Developer Days 2019 – Taipei, Taiwan
November 19, 2019
Agenda

- Logical Layer
- Re-timer
- Transport Layer
Agenda

- Logical Layer
- Re-timer
- Transport Layer
Logical Layer Agenda

- Terms and Abbreviations
- Logical Layer Roles
- Link Bring-up Flow
- Sideband Channel
  - Transactions
  - Lane Initialization
- Lane Adapter State Machine
- Link Encoding and Operation
- CLx State
- Sleep and Wake
Terms and Abbreviations

• **Lane Adapter** – The Adapter that interfaces to a Lane. A USB4 Port contains two Lane Adapters: a Lane 0 Adapter and a Lane 1 Adapter. A Lane Adapter implements an Electrical Layer, a Logical Layer and a Transport Layer.

• **Lane** – The dual simplex high speed differential signaling pair that provides communication between two interconnected Routers. The signaling rate at which a Lane operates defines the speed of communication for that Lane.

• **Sideband (SB) Channel** – The connection between the USB4 Ports of two interconnected Routers that provides initial communication and setup functionality.

• **USB4 Link (Link)** – The logical connection between the USB4 Ports of two interconnected Routers. A USB4 Link may include either one or two Lanes. Unless otherwise noted, the term “Link” without any preceding descriptor refers to a USB4 Link.

• **USB4 Port (Port)** – A Router interface consisting of two Lane Adapters and a Sideband Channel. Protocol Adapters and Control Adapters are never part of a USB4 Port.

• **CM** – Connection Manager

• **UFP/DFP** – Upstream/Downstream Facing Port
Logical Layer Roles

• Established Link between two Routers

• Provide services to the transmitter and receiver
  • Error detection & recovery
  • Different media
  • Power management

• Using Sideband to manage Link
Link Bring-up Flow

- USB PD
  - Negotiation determines USB4
- Sideband
  - Negotiate Link Parameters
    - Lanes
    - Gen
    - Coding
  - Lane Equalization Management
- High-Speed
  - Lane Equalization
  - Bonding

![Diagram of USB Type-C Full-Featured Cable and USB Type-C Plugs](image-url)
Link Bring-up Flow

- **USB PD**
  - Negotiation determines USB4
- **Sideband**
  - Negotiate Link Parameters
    - Lanes
    - Gen
    - Coding
  - Lane Equalization Management
- **High-Speed**
  - Lane Equalization
  - Bonding
Link Bring-up Flow

- **USB PD**
  - Negotiation determines USB4
- **Sideband**
  - Negotiate Link Parameters
    - Lanes
    - Gen
    - Coding
    - Lane Equalization Management
- **High-Speed**
  - Lane Equalization
  - Bonding

![USB Type-C Full-Featured Cable Diagram](image)
Link Bring-up Flow

- USB PD
  - Negotiation determines USB4
- Sideband
  - Negotiate Link Parameters
    - Lanes
    - Gen
    - Coding
  - Lane Equalization Management
- High-Speed
  - Lane Equalization
  - Bonding
Link Bring-up Flow

• USB PD
  • Negotiation determines USB4
• Sideband
  • Negotiate Link Parameters
    • Lanes
    • Gen
    • Coding
  • Lane Equalization Management
• High-Speed
  • Lane Equalization
  • Bonding
Logical Layer Agenda

- Terms and Abbreviations
- Logical Layer Roles
- Link Bring-up Flow
- **Sideband Channel**
  - Transactions
  - Lane Initialization
- Lane Adapter State Machine
- Link Encoding and Operation
- CLx State
- Sleep and Wake
Sideband Channel

- Sideband Transactions are sent over SBTX and received over the SBRX
- Re-timers forward and/or process the Sideband Transactions
Sideband Channel (cont.)

- Sideband Symbols 10-bit Start/Stop (Start bit 0, 8 bits of payload, Stop bit 1)
Sideband Transactions

• Three types of transactions:
  • Link Type – LT – used in Lane initialization
  
  • Administrative Type – AT – used to access SB Register space (Cmd/Rsp) of a Router
  
  • Re-timer Type – RT – used to communicate with Router/Re-timer, can be Broadcast
Link Type Transactions

- LT Transactions are used during Link Initialization and also to inform a change in the Lane Adapter state

<table>
<thead>
<tr>
<th>Byte</th>
<th>Payload Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FEh</td>
<td>Data Link Escape (DLE) Symbol – indicates the beginning of a transaction.</td>
</tr>
<tr>
<td>1</td>
<td>See Table 4-2</td>
<td>Lane State Event (LSE) – identifies a change in originator’s state or in the Adapter state.</td>
</tr>
<tr>
<td>2</td>
<td>Bitwise complement of LSE</td>
<td>Complement LSE (CLSE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
</table>
| [3:0] | LSESymbol  | This field defines the LT Transaction type. Undefined values are Rsvd.  
0000b – LT_Fall (Lane Disable event)  
0010b – LT_Resume (USB4™ Port started USB4 transmission)  
0011b – LT_LRoff (disconnect or System Sleep state) |
| [4]   | Rsvd       | Reserved |
| [5]   | LSELane    | Indicates the Lane affected by this transaction. Shall be 0b (for Lane 0) or 1b (for Lane 1).  
Shall be set to 0b when issuing an LT_LRoff Transaction. |
| [7:6] | StartLT    | Identifies that this is an LT Transaction. Shall be set to 10b. |
Administrative Type Transactions

- Bits [7:6] in the STX indicate it’s a AT transaction (00b)
- Can access a Router Sideband Register space
- Can read from or write to a multi-byte register
- Source Router initiate a Command and responding Router answers a Response
- Data Symbols include Address, Length, WnR (Read/Write indication) and Data (for Write Commands or Read Responses)

<table>
<thead>
<tr>
<th>Byte</th>
<th>Payload Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FEh</td>
<td>Data Link Escape (DLE) Symbol – indicates the beginning of a transaction.</td>
</tr>
<tr>
<td>1</td>
<td>See Table 4-4</td>
<td>Start Transaction (STX) Symbol – defines the operation of the transaction.</td>
</tr>
<tr>
<td>[n+1:2]</td>
<td>See Section 4.1.1.3</td>
<td>Data Symbols – the number of Data Symbols is not explicitly defined and is inferred from the end-of-transaction delimiters (DLE-ETX). The number of Data Symbols (n) shall not exceed 66.</td>
</tr>
<tr>
<td>n+2</td>
<td>Low-order byte of 16-bit CRC</td>
<td>Low CRC (LCRC) Symbol</td>
</tr>
<tr>
<td>n+3</td>
<td>High-order byte of 16-bit CRC</td>
<td>High CRC (HCRC) Symbol</td>
</tr>
<tr>
<td>n+4</td>
<td>FEh</td>
<td>Data Link Escape (DLE) Symbol</td>
</tr>
<tr>
<td>n+5</td>
<td>40h</td>
<td>End of Transaction (ETX) Symbol</td>
</tr>
</tbody>
</table>
Re-timer Type Transactions

- Can access Re-timers with two types of transactions:
  - Broadcast RT
    - Initiated by Routers
    - Holds the Link Parameters
    - Enumerates Re-timers
  - Addressed RT
    - Initiated by Router or Re-timer
    - Access Router/Re-timer Sideband Register space
    - Used in TxFFE negotiation
Broadcast RT Transaction

- Bits [7:5] in the STX indicate it’s a Broadcast RT transaction (011b)
- Link Parameters holds information about the Link that was negotiated by Routers
  - Gen, RS-FEC, Lanes
- CRC is used for errors detection

<table>
<thead>
<tr>
<th>Byte</th>
<th>Payload Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FEh</td>
<td><strong>Data Link Escape (DLE) Symbol</strong> – indicates the beginning of a transaction.</td>
</tr>
<tr>
<td>1</td>
<td>See Table 4-6</td>
<td><strong>Start Transaction (STX) Symbol</strong> – defines the operation of the transaction.</td>
</tr>
<tr>
<td>3:2</td>
<td>See Table 4-7 and Table 4-8</td>
<td><strong>Link Parameters</strong> – contains a list of the Link parameters selected during Lane Initialization.</td>
</tr>
<tr>
<td>4</td>
<td>Low-order byte of 16-bit CRC</td>
<td><strong>Low CRC (LCRC) Symbol</strong></td>
</tr>
<tr>
<td>5</td>
<td>High-order byte of 16-bit CRC</td>
<td><strong>High CRC (HCRC) Symbol</strong></td>
</tr>
<tr>
<td>6</td>
<td>FEh</td>
<td><strong>Data Link Escape (DLE) Symbol</strong></td>
</tr>
<tr>
<td>7</td>
<td>40h</td>
<td><strong>End of Transaction (ETX) Symbol</strong></td>
</tr>
</tbody>
</table>
Addressed RT Transaction

- Bits [7:5] in the STX indicate it’s an Addressed RT transaction (010b)
- STX holds the Re-timer Index to access where 0 is adjacent Re-timer/Router
- Can access a Router/Re-timer Sideband Register space
- Can read from or write to a multi-byte register
- *Data Symbols* are the same as in AT transactions

<table>
<thead>
<tr>
<th>Byte</th>
<th>Payload Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FEh</td>
<td>Data Link Escape (DLE) Symbol – indicates the beginning of a transaction.</td>
</tr>
<tr>
<td>1</td>
<td>See Table 4-4</td>
<td>Start Transaction (STX) Symbol – defines the operation of the transaction.</td>
</tr>
<tr>
<td>[n+1:2]</td>
<td>See Section 4.1.1.3</td>
<td>Data Symbols – the number of Data Symbols is not explicitly defined and is inferred from the end-of-transaction delimiters (DLE-ETX). The number of Data Symbols (n) shall not exceed 66.</td>
</tr>
<tr>
<td>n+2</td>
<td>Low-order byte of 16-bit CRC</td>
<td>Low CRC (LCRC) Symbol</td>
</tr>
<tr>
<td>n+3</td>
<td>High-order byte of 16-bit CRC</td>
<td>High CRC (HCRC) Symbol</td>
</tr>
<tr>
<td>n+4</td>
<td>FEh</td>
<td>Data Link Escape (DLE) Symbol</td>
</tr>
<tr>
<td>n+5</td>
<td>40h</td>
<td>End of Transaction (ETX) Symbol</td>
</tr>
</tbody>
</table>
Lane Initialization

- Phase 1 – Determination of Initial Conditions
- Phase 2 – Router Detection
- Phase 3 – Determination of USB4 Port Characteristics
- Phase 4 – Lane Parameters Synchronization and Transmit Start
- Phase 5 – Link Equalization
Phase 1 – Determination of Initial Conditions

- Router discovers connection information (see USB PD & USB Type-C® Specifications for more information)
  - USB4 Link?
  - Reverse insertion of USB Type-C connector?
  - Active Cable with Re-timers?
  - TBT3 Active Cable?
  - Cable supports Gen3?

- Router continue to Phase 2 only if the Link is USB4
- Router continue to Phase 2 only after all information is obtained
- SBTX is driven low by default
Phase 1 (cont.) – Lane reversal

- Lane mismatch is caused by reverse insertion of a USB Type-C connector
- PD controller notify the Router and relevant Re-timer
- Each Router Assembly resolves the mismatch on its side
Phase 2 – Router Detection

- Host Router driving SBTX to logic high
- Device Router driving SBTX to logic high after detecting logic high on its SBRX
Phase 3 – Determination of USB4 Port Characteristics

• Each Router acquires information about its Link Partner using AT transactions over the Sideband

• Deciding on the link parameters:
  • Enabling – Lanes on each side
  • Dual-Lane – 2 Lanes & Lane Bonding are enabled on both sides
  • Speed – Adapters and Cable support Gen3 or not
  • RS-FEC – Enabled if both side request

• Decision is made by specific rules and will be the same on both Routers
• Decision is final for the entire Link and may change only on next Link Initialization
Phase 4 – Lane Parameters Synchronization and Transmit Start

1. Each Router periodically sends a Broadcast RT transaction with its decision until receives a Broadcast RT transaction
   - Broadcast RT transaction enumerates Re-timers on the link
   - Each Re-timer holds two indexes

2. Upon reception of a Broadcast RT transaction a Router activate its transmitter and start sending training sequence

3. Router sends LT_Resume transaction for each active Lane

4. Moving to Phase 5
Phase 5 – Link Equalization

- Transmitter Feed Forward Equalization (TxFFE) initiated by the receiver & sets transmitter parameters to improve signal over media
  - Executed in parallel in each segment, symmetrically on each direction for each lane
  - Re-timers use local clock (without SSC) to generate TX signal
- Clock switch in Re-timers happens after TxFFE is done
Phase 5 – Link Equalization

Transmitter flow
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

Receiver Flow
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
**Phase 5 – Link Equalization**

**Transmitter flow**
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

**Receiver Flow**
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

Transmitter flow
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

Receiver Flow
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

Transmitter flow
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

Receiver Flow
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

**Transmitter flow**
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

**Receiver Flow**
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
**Phase 5 – Link Equalization**

**Transmitter flow**
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

**Receiver Flow**
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

Transmitter flow
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

Receiver Flow
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

**Transmitter flow**
1. TX with default parameters
2. **Checks if “RX locked”**
3. If not “RX Locked” & there is new request, it updates the TX parameters

**Receiver Flow**
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

**Transmitter flow**
1. TX with default parameters
2. Checks if “RX locked”
3. If not “RX Locked” & there is new request, it updates the TX parameters

**Receiver Flow**
1. Evaluate signal
2. If signal is good indicate “RX locked”
3. If want to try different parameters set “New Req”
Phase 5 – Link Equalization

- Same happens for all Lanes in both directions
- Sideband channel is shared for all TxFFE
Phase 5 – Clock Switch

• When a Re-timer finishes TxFFE on both Lanes it is ready for Clock Switch

• Clock Switch with transition to Forwarding state in the Re-timer happens from “last” to “first”
Take Five Speaker Break
Logical Layer Agenda

- Terms and Abbreviations
- Logical Layer Roles
- Link Bring-up Flow
- Sideband Channel
  - Transactions
  - Lane Initialization
- Lane Adapter State Machine
- Link Encoding and Operation
- CLx State
- Sleep and Wake
Lane Adapter State Machine

- **Disabled state** – The Lane Adapter disables the Lane
- **CLd state** – Lane Adapter transmitter and receiver are inactive
- **Training state** – The Lane Adapter performs Symbol synchronization and transfer of Lane parameters
- **CL0 state** – The Lane Adapter can transmit and receive Transport Layer Packets across the Lane
- **Lane Bonding state** – bonds two Single-Lane Links into a Dual-Lane Link
- **CL0s, CL1, CL2 states** – low power states
Lane Adapter State Machine

1. **Lane is Disabled by software**
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. **CM enables the Lane**
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. **Link Initialization start**
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. **In Phase 5 TX & RX are active**
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. **Training done – Lane is functional**
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. **CM configure Lane Bonding**
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. **Lane Bonding done**
8. Conditions to CL1 are met
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. Lane Bonding done
8. **Conditions to CL1 are met**
9. Objection to CL1 is set
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. Training done – Lane is functional
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. **Objection to CL1 is set**
Lane Adapter State Machine

1. Lane is Disabled by software
2. CM enables the Lane
3. Link Initialization start
4. In Phase 5 TX & RX are active
5. **Training done – Lane is functional**
6. CM configure Lane Bonding
7. Lane Bonding done
8. Conditions to CL1 are met
9. Objection to CL1 is set
Logical Layer Agenda

- Terms and Abbreviations
- Logical Layer Roles
- Link Bring-up Flow
- Sideband Channel
  - Transactions
  - Lane Initialization
- Lane Adapter State Machine
- **Link Encoding and Operation**
  - CLx State
  - Sleep and Wake
**USB4™ Link Encoding**

- When operating in Dual-Lane Link distribute Transport Layer bytes to Lanes

- Bytes are sent from BC $B_{N-1}$
USB4™ Link Encoding (RS-FEC disabled)

- Gen2 -> 64b/66b encoding
  - Data
  - Ordered Set

- Gen3 -> 128b/132b encoding
  - Data
  - Ordered Set
Ordered Sets

- SLOS1 & SLOS2 for training, never scrambled, PRBS11, 2048b
- TS1 & TS2 for Training and Bonding, 64b
- CL1/2_REQ, CL0s/1/2_ACK, CL_NACK, CL_OFF for CLx entry flow, 64b
- WAKE1/2.X for CLx exit flow, never scrambled, same as SLOS1/2 with index (X)

- Example of CL_OFF Ordered Set

<table>
<thead>
<tr>
<th>Bits</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>63:10</td>
<td>00 0011</td>
<td>Ordered Set contents.</td>
</tr>
<tr>
<td></td>
<td>1111 1100 0011 1111 1100 0011 1111 1100 0100b</td>
<td></td>
</tr>
<tr>
<td>9:0</td>
<td>00 1111 0010b</td>
<td>SCR = Shall be set to this value to indicate that the Ordered Set contents are scrambled.</td>
</tr>
</tbody>
</table>
USB4™ Link Encoding

• Sync bits are sent first (MSB to LSB)

• Bytes are send MSB to LSB

• Bits within a byte are sent LSB to MSB

• Example of data sent on Gen2:
USB4™ Link Encoding

- Scrambling on bits in the order they are sent on the line
- Not all bits are scrambled

<table>
<thead>
<tr>
<th></th>
<th>Scramble?</th>
<th>Advanced LFSR?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Symbols</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync Bits</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Transport Layer bytes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Ordered Sets</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync Bits</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Contents</td>
<td>Determined by SCR value</td>
<td>When scrambled</td>
</tr>
<tr>
<td>SCR</td>
<td>No</td>
<td>When contents are scrambled</td>
</tr>
</tbody>
</table>
USB4™ Link Encoding

- An Adapter may transmit SKIP Ordered Set at any time
- Recommended not to send unless explicitly required
- If in Gen2 RS-FEC is enabled SKIP is inserted when only one Ordered Set is required
- Shall not send while in Training.LOCK1 and Training.LOCK2

SKIP Ordered Set

<table>
<thead>
<tr>
<th>Bits</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>63:10</td>
<td>00 1010</td>
<td>Contents, specific to the Ordered Set.</td>
</tr>
<tr>
<td></td>
<td>1100 1010 1100 1010</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1100 1010 1100 1010</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1100 1010 1101 1100b</td>
<td></td>
</tr>
<tr>
<td>9:0</td>
<td>11 0000 1101b</td>
<td><strong>SCR</strong> – Shall be set to this value to indicate that the Ordered Set contents are not scrambled.</td>
</tr>
</tbody>
</table>
USB4™ Link Encoding

- Reed-Solomon forward error correction code uses RS(198,194) over GF(2^8)
- RS-FEC is mandatory in all speeds
- 12 16-byte Symbol payload and 2-byte Sync Bits
- Sync Bits indicate if a Symbol payload is Data or Ordered Set
USB4™ Link Encoding

- Redundancy bytes are sent last
- Can fix up to 2 error bytes
- Each RS-FEC Symbol payload (16-bytes) holds only Data or only Ordered Set
- In Gen3, if Ordered Set is only 64b (8-bytes) it is sent twice
- In Gen2, if only one 64b (8-bytes) Ordered Set is required a SKIP is added
USB4™ Link Encoding

- Pre-coding is XORing a bit with previous bit
- Active only when RS-FEC is active
- RS-FEC activated in TS1 in Training, in TS2 when exiting CL0s
Logical Layer Agenda

• Terms and Abbreviations
• Logical Layer Roles
• Link Bring-up Flow
• Sideband Channel
  • Transactions
  • Lane Initialization
• Lane Adapter State Machine
• Link Encoding and Operation
  • CLx State
• Sleep and Wake
**CLx State**

- Can be symmetric (CL1/CL2) or asymmetric (CL0s)
- Entry time < 9us, Exit time CL1<240us/CL2<1ms
- When requested?
  - All components in the Link support CLx
  - If there are no DP tunneled packets to transmit
  - If PCIe tunneled should be in L1 and LTR > Entry+Exit CLx time
  - If USB3 is tunneled should be in U2 or U3
  - TMU handshake should not be delayed
- When acknowledged?
  - Same as above
  - Can send CL1_ACK after getting CL2_REQ
  - If objecting, send CL0s_ACK
**CL2/CL1 Entry Flow**

1. A sends CL2_REQ back-to-back
2. B receives CL2_REQ & if ready, sends CL2_ACK back-to-back
3. A receives CL2_ACK, shut down RX & sends 375 CL_OFF
4. B receives CL_OFF and shut down RX & TX
5. After A sent 375 CL_OFF it shut down TX
**CL0s Entry Flow**

1. A sends CL2_REQ back-to-back
2. B receives CL2_REQ & not ready, sends 16 CL0s_ACK back-to-back
3. A receives CL0s_ACK & sends 375 CL_OFF
4. B receives CL_OFF and shut down RX
5. After A sent 375 CL_OFF it shut down TX
CL2 Exit Flow

• A initiate CL0s exit by sending LFPS
• B respond LFPS and start TX
• When A detects LFPS it also start TX
• Routers TX SLOS1 and lock on RX
• Move to Training state
CL2 Exit Flow

- A initiate CL0s exit by sending LFPS
- B respond LFPS and start TX
- When A detects LFPS it also start TX
- Routers TX SLOS1 and lock on RX
- Move to Training state
CL2 Exit Flow

- A initiate CL0s exit by sending LFPS
- B respond LFPS and start TX
- **When A detects LFPS it also start TX**
- Routers TX SLOS1 and lock on RX
- Move to Training state
CL2 Exit Flow

• A initiate CL0s exit by sending LFPS
• B respond LFPS and start TX
• When A detects LFPS it also start TX
• **Routers TX SLOS1 and lock on RX**
• Move to Training state
CL2 Exit Flow

- A initiate CL0s exit by sending LFPS
- B respond LFPS and start TX
- When A detects LFPS it also start TX
- Routers TX SLOS1 and lock on RX
- Move to Training state
Sleep and Wake

- When Host goes to Sleep it informs the entire USB4 tree
- Each Router indicate when ready
- USB4 Link is not active during Sleep, Sideband channel might remain connected
- Some data retained in the Router

<table>
<thead>
<tr>
<th>Wake Event</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wake on Connect</td>
<td>Connection detected</td>
</tr>
<tr>
<td>Wake on Disconnect</td>
<td>Disconnection detected</td>
</tr>
<tr>
<td>Wake on Inter-Domain</td>
<td>Disconnection of an Inter-Domain Link detected</td>
</tr>
<tr>
<td>Wake on PCIe</td>
<td>PCIe Wake event from connected PCIe Endpoint or Switch</td>
</tr>
<tr>
<td>Wake on USB3</td>
<td>USB3 Wake event from connected USB3 device</td>
</tr>
<tr>
<td>Wake on USB4</td>
<td>SBRX low for tWake time</td>
</tr>
</tbody>
</table>
Time for Q&A
Agenda

- Logical Layer
- Re-timer
- Transport Layer
Re-timer Agenda

• Terms and Abbreviations
• Re-timer?
• Sideband
  • Lane Initialization
• Channel State Machine
  • Low Power States (CL0s, CL1, CL2)
• Lane Decoding
Terms and Abbreviations

- Router Facing Port
- Cable Facing Port
- Adjacent
- Corresponding Receiver
- Corresponding Transmitter
- Corresponding Adapter
- Re-timer Channel
Re-timer?

- **Electrical** and **Logical** Active Device on a USB4 Link
- Can be part of Router Assembly – Onboard Re-timer
- Can be part of Active Cable – Cable Re-timer
- Re-timer spec defines different requirements for Cable Re-timer and Onboard Re-timer
Sideband

- Forward LT and AT transactions
- Broadcast RT transactions enumerate Re-timers
  - Re-timer increase *Index* field, use it and forward

- Addressed RT transactions
  - *Index* is 0 or match – Process and respond
  - *Index* not 0 and no match – Forward
Lane Initialization

- **Phase 1** – USB4 Link, Reverse insertion of USB Type-C connector*
- **Phase 2** – Detect SBRX logic high for \(t\text{ConnectRx}\) and set SBTX logic high
- **Phase 3** – Not Applicable
- **Phase 4** – Broadcast RT transaction informs the Re-timer about Link characteristics and enumerate it. Moving to Phase 5 on LT_Resume

* Only Onboard Re-timer adjacent to USB Type-C connector
Lane Initialization

- **Phase 5** – Lane Equalization same as Routers except clock switch

Before Clock Switch

After Clock Switch
Channel State Machine

Each Re-timer Channel has its own State Machine

- **CLd** – Lane Initialization
- **Bit Lock** – Bit synchronization, receiver equalization, Clock Switch
- **Forwarding** – Forwards without modifications bit stream from RX to TX
- **CL0s/CL1/CL2** – Similar to Router behavior with different actions
CL2/CL1 Entry Flow

1. Re-timer B detects CL2_ACK and start CL2_ACK counter
4. Re-timer B detects CL_OFF and start CL_OFF counter
5. Re-timer B CL2_ACK Counter reaches 372, shut down B->A direction
8. Re-timer B CL_OFF Counter reaches 372, shut down A->B direction
CL0s

• Router A transmitter is off
• Router B receiver is off
• Router B sends data to Router A
• Re-timer has 2 channels in CL0s and 2 channels in CL0
• Control Symbols from Router B to Router A are interleaved within the data
CL0s Exit Flow

- A initiate CL0s exit by sending 16 LFPS
- Re-timers forward LFPS
- Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index
- When Router B gets WAKE1.X it respond with WAKE2.X
- When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction
CL0s Exit Flow

- A initiate CL0s exit by sending 16 LFPS
- **Re-timers forward LFPS**
  - Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index
  - When Router B gets WAKE1.X it respond with WAKE2.X
  - When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction
CL0s Exit Flow

- A initiate CL0s exit by sending 16 LFPS
- Re-timers forward LFPS
- **Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index**
- When Router B gets WAKE1.X it respond with WAKE2.X
- When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction
CL0s Exit Flow

- A initiate CL0s exit by sending 16 LFPS
- Re-timers forward LFPS
- Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index
- **When Router B gets WAKE1.X it respond with WAKE2.X**
- When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction
**CL0s Exit Flow**

- A initiate CL0s exit by sending 16 LFPS
- Re-timers forward LFPS
- Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index
- When Router B gets WAKE1.X it respond with WAKE2.X

**When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction**
**CL0s Exit Flow**

- A initiate CL0s exit by sending 16 LFPS
- Re-timers forward LFPS
- Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index

**When Router B gets WAKE1.X it respond with WAKE2.X**

- When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction
CL0s Exit Flow

• A initiate CL0s exit by sending 16 LFPS
• Re-timers forward LFPS
• Router TX SLOS1, Re-timers TX WAKE1.X when X is their Index
• When Router B gets WAKE1.X it respond with WAKE2.X
• When a Re-timer gets WAKE2.X matching its Index it switch to Forwarding in the opposite direction
CL0s Exit Flow

- When Router B detects SLOS1 Symbols it sends 16 TS2 Symbols
- When Router A detects TS2 Symbols it sends at least 16 TS2 Symbols
CL0s Exit Flow

- When Router B detects SLOS1 Symbols it sends 16 TS2 Symbols
- When Router A detects TS2 Symbols it sends at least 16 TS2 Symbols
Lane Decoding

- To track CLx states and participate in CLx exit flow
- Re-timer decodes Ordered Sets
  - Removed pre-coding
  - RS-FEC redundancy removal (error checking is not mandatory)
  - De-scrambling
  - Symbol decoding
- Decoding path is independent of the Forwarding path and has no effect on it
Time for Q&A
Agenda

• Logical Layer
• Re-timer
• Transport Layer
Agenda

• Roles
• Constructs
  • Adapter
  • Path
  • Routing Table
• Packets
• QoS
Transport Layer Roles

• Routing Transport Layer Packets between Adapters

• Quality Of Service
  • Packet Arbitration
  • Flow Control
Adapter

- Provides an interface to the Transport Layer

- Types:
  - **Lane Adapter** – Implements the connection to a USB4 Link
  - **Control Adapter** – Implements the Configuration Layer
  - **Protocol Adapter** – Implements translation and encapsulation of Native protocol
    - USB3, DP, PCIe, Host Interface

- Lane and Protocol Adapters are SW addressable through Adapter Configuration Space
Path

- Provides a Uni-directional, logical end-to-end connection between two Protocol Adapters
- Spans through multiple Routers
- Represents a “virtual wire” for the native I/O protocol being carried by the Path
- Identified by a HopID in the Transport Layer Packet header – unique per Link per direction
Path

- Provides a Uni-directional, logical end-to-end connection between two Protocol Adapters
- Spans through multiple Routers
- Represents a “virtual wire” for the native I/O protocol being carried by the Path
- Identified by a **HopID** in the Transport Layer Packet header – unique per Link per direction

The 2 Red Paths going Downstream will have different **HopIDs**
The 2 Blue Paths going Upstream will have different **HopIDs**
Routing Table

• Path setup is done by Connection Manager, configuring each Ingress Adapter along the Path.

• Routing Table at Ingress Adapter determines the **Egress HopID** and **Egress Adapter** per Ingress HopID.
Routing Table

- Path setup is done by Connection Manager, configuring each Ingress Adapter along the Path.

- Routing Table at Ingress Adapter determines the Egress HopID and Egress Adapter per Ingress HopID.
Routing Table

- Path setup is done by Connection Manager, configuring each Ingress Adapter along the Path.

- Routing Table at Ingress Adapter determines the **Egress HopID** and **Egress Adapter** per Ingress HopID.
Routing Table

• Path setup is done by Connection Manager, configuring each Ingress Adapter along the Path

• Routing Table at Ingress Adapter determines the Egress HopID and Egress Adapter per Ingress HopID
Routing Table

• Path setup is done by Connection Manager, configuring each Ingress Adapter along the Path

• Routing Table at Ingress Adapter determines the Egress HopID and Egress Adapter per Ingress HopID

Two Paths, over the same Link, going in different direction may have the same HopID

The Egress HopID may remain the same as the Ingress HopID
Transport Packet

- **HEC** – Header Error Correction
- **Length** – Payload size in Bytes
- **HopID** – Uniquely identifies a Path in context of a Link
- **SuppID** – Used to distinguish certain types of Link Management Packets
- **PDF** – Protocol Defined Field. Defines the payload of the Packet within the context of a Path.
Transport Packet Types

• **Tunneled Packets**
  • Generated and consumed by Protocol Adapters
  • Travels through Paths setup by Connection Manager

• **Control Packets**
  • Generated and consumed by Control Adapters and Connection Manager (Through Host Interface Adapter)
  • Travels over Path 0

• **Link Management Packets**
  • Generated and consumed by Transport layer
  • Travels from a Transport Layer at one side of the Link and terminate at the Transport Layer at the other side of the Link.
Link Management Packets

• **Idle Packets**
  • Single DW, constant value
  • Ensures that the Transport Layer feeds the Logical Layer with a continuous byte stream at CL0

• **Flow Control Packets**
  • Update and Sync Credit information per Path

• **TMU Packets**
  • Follow up Packets which participate in the Time Management Unit handshake for time convergence.
QoS - Arbitration

• Traffic manager for an Egress Adapter uses 3-layer scheduling scheme:
  • **Link Scheduler**
    1. Flow Control
    2. Time Sync Packets
    3. Priority Group Scheduler
  • **Priority Group Scheduler**
    • Strict priority
    • “0” highest priority
  • **Path Scheduler ‘n’**
    • Weighted round-robin (WRR) scheduling scheme among the Paths in Priority Group ‘n’

• Connection Manager configures **Priority** and **WRR** at Path setup
QoS - Arbitration

- Traffic manager for an Egress Adapter uses 3-layer scheduling scheme:
  - **Link Scheduler**
    1. Flow Control
    2. Time Sync Packets
    3. Priority Group Scheduler
  - **Priority Group Scheduler**
    - Strict priority
    - “0” highest priority
  - **Path Scheduler ‘n’**
    - Weighted round-robin (WRR) scheduling scheme among the Paths in Priority Group ‘n’

- Connection Manager configures **Priority** and **WRR** at Path setup
QoS - Arbitration

- Traffic manager for an Egress Adapter uses 3-layer scheduling scheme:
  - **Link Scheduler**
    1. Flow Control
    2. Time Sync Packets
    3. Priority Group Scheduler
  - **Priority Group Scheduler**
    - Strict priority
    - “0” highest priority
  - **Path Scheduler ‘n’**
    - Weighted round-robin (WRR) scheduling scheme among the Paths in Priority Group ‘n’

- Connection Manager configures **Priority** and **WRR** at Path setup
QoS – Flow Control

• The Transport Layer employs a per-Link, credit-based flow control mechanism to prevent overflow of receive buffers at the Ingress Lane Adapter.

• Each Path is configured to use one of the Flow Control Schemes:
  • **Dedicated**
    • The Ingress Adapter has buffer space allocated for the Path
  • **Disabled**
    • Transport Layer Packets are not flow controlled
  • **Shared**
    • The Ingress Adapter has a shared buffer for all Paths that use shared flow control
  • **Restricted Shared**
    • The Ingress Adapter uses the shared buffer, but the Path can only use a limited amount of space in the shared buffer
QoS – Flow Control - Packets

• **Credit Grant Packet**
  - Sent by an Ingress Adapter to indicate its receive buffers status
  - Each DW represents a *single Credit Grant Record*
    - L = 0: Dedicated Path update for the HopID
    - L = 1: Shared Link update
  - HopID = 1
  - PDF = 1

• **Path Credit Sync Packet**

<table>
<thead>
<tr>
<th>PDF</th>
<th>HopID</th>
<th>Length</th>
<th>HEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Rsvd</td>
<td>HopID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Credits</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECC</td>
<td></td>
</tr>
</tbody>
</table>
QoS – Flow Control - Packets

• Credit Grant Packet

• Path Credit Sync Packet
  • Sent by an Egress Adapter to indicate the number of consumed buffers. Used to overcome a lost of a Tunneled Packet.
  • A Packet contains information for a single Path
    • PDF = 0
    • SuppID = 1

<table>
<thead>
<tr>
<th>PDF=0</th>
<th>HopID = Target</th>
<th>Length = 4</th>
<th>HEC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rsvd</td>
<td>PCC</td>
<td>ECC</td>
</tr>
</tbody>
</table>

* SuppID = 1
QoS – Flow Control – Dedicated

• **Connection Manager**
  • Path Setup – Sets Flow Control Scheme and Allocate Credits

• **Ingress Adapter**
  • First – Sends Credit Grant Packet with the \( \text{Credits} = \text{Allocation} \)
  • Subsequently – Upon Packet dequeue sends a Credit Grant Packet with the updated \( \text{Credits} \) value

• **Egress Adapter**
  • Set the \( \text{Limit} \) to be the value received from Credit Grant Packet
  • Upon sending a Packet, increment the \( \text{Consumed} \) credits
  • Allowed to send Packet only if \( \text{Consumed} < \text{Limit} \)
Time for Q&A