Thunderbolt Protocol

Rozhin Taghizadegan

Interface Circuit Design – February 2025

Overview of Thunderbolt

- **Thunderbolt** is a high-speed data transfer and display technology developed by Intel in collaboration with Apple. It combines multiple protocols, including **PCIe, DisplayPort, and USB**, over a single high-bandwidth connection.
- It also supports **power delivery**, making it ideal for connecting high-performance peripherals.

Generations of Thunderbolt

Generation	Max Speed	PCIe Version	DisplayPort Version	Connector Type
Thunderbolt 1	10 Gbps	PCIe 2.0	DP 1.1	Mini DisplayPort
Thunderbolt 2	20 Gbps	PCIe 2.0	DP 1.2	Mini DisplayPort
Thunderbolt 3	40 Gbps	PCIe 3.0	DP 1.2	USB-C
Thunderbolt 4	40 Gbps	PCIe 3.0	DP 1.4	USB-C

Key Features of Thunderbolt

1. **High-Speed Data Transfer**
2. **Daisy-Chaining**
3. **Multiple Protocols**
4. **External GPU (eGPU) Support**
5. **Power Delivery (PD)**
6. **Backward Compatibility**

Thunderbolt Physical Layer Blocks

- **Transceivers (TX/RX):** Convert digital data into electrical signals and vice versa
- **Differential Signaling Circuit:** Uses **Low Voltage Differential Signaling (LVDS)** to reduce noise
- **Multiplexers/Demultiplexers (MUX/DEMUX):** Switch between PCIe, DisplayPort, and USB signals
- **Clock Management Unit:** Provides timing for accurate data transmission
- **Power Delivery Circuitry:** Manages power supply (up to **100W** for charging)
- **Encoding/Decoding Units:** Convert data into a format suitable for transmission.

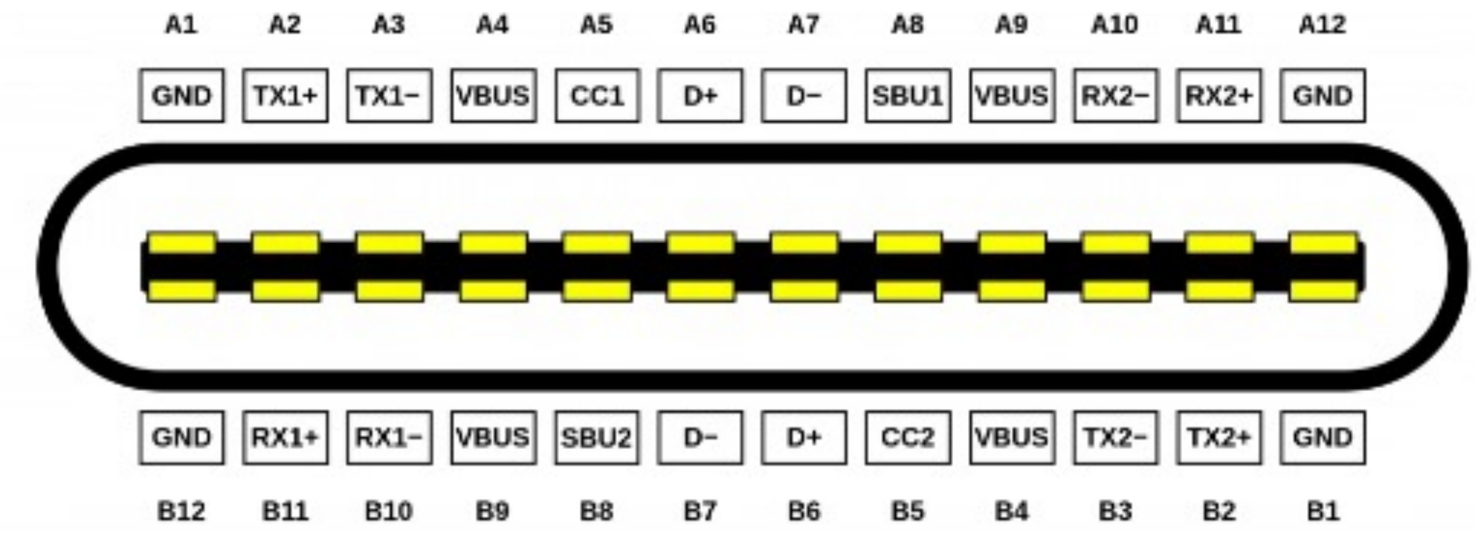
Differential Signaling in Thunderbolt

- The transmitter sends **two complementary signals** over two wires
- The receiver calculates the **voltage difference (**V_{diff}**)**:
$$V_{diff} = V_{+} - V_{-}$$
- Since noise affects both wires equally, the difference remains **constant**, making the system **immune to external interference**.

Thunderbolt Version	Encoding Scheme	Efficiency
Thunderbolt 1 & 2	8b/10b Encoding	80%
Thunderbolt 3	128b/130b Encoding	98%
Thunderbolt 4	PAM-3 (Pulse Amplitude Modulation)	~100%

Essential Connections in Thunderbolt

Pin	Function
TX+ / TX-	Differential data transmission
RX+ / RX-	Differential data reception
GND (Ground)	Power return path
Vbus (Power Delivery)	Up to 100W for charging
CC (Configuration Channel)	Manages device connection
SBU (Sideband Use)	Auxiliary signal for DisplayPort



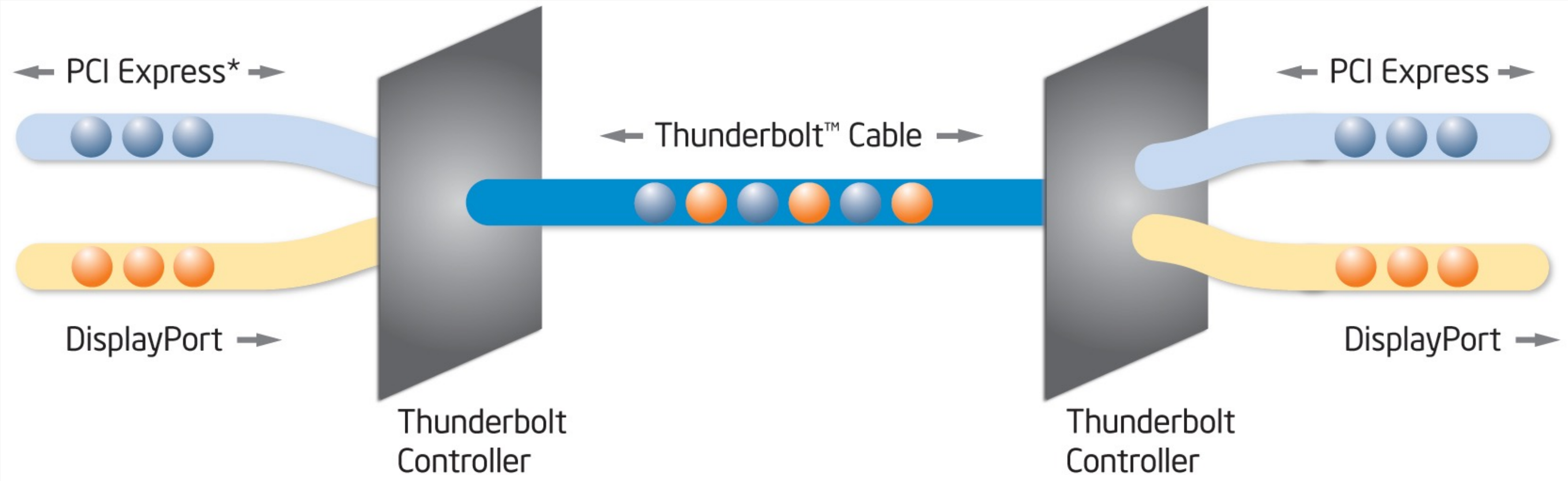
Thunderbolt 3 and 4 pinout

Thunderbolt Bus Communication: Serial or Parallel?

- Thunderbolt uses a **serial** communication method rather than parallel. It transmits data in a **high-speed, serialized format** over differential pairs, which reduces electromagnetic interference (EMI) and allows for longer cable distances.

Why Serial and Not Parallel?

- Serial communication is faster and more efficient at high data rates
- Parallel transmission suffers from clock skew and crosstalk, making it unsuitable for high-speed protocols like Thunderbolt
- Thunderbolt employs multiplexing, allowing multiple data streams (PCIe, DisplayPort) to be sent simultaneously over a single serial link.


PCI Express*
DisplayPort
Thunderbolt™ Cable
Thunderbolt Controller
Thunderbolt Controller
PCI Express
DisplayPort

Is Thunderbolt Synchronous or Asynchronous?

- **Thunderbolt Uses Synchronous Transmission**
  - Thunderbolt operates using a **shared clock signal** for precise synchronization.
  - Data packets are **aligned with a timing reference**, ensuring minimal delays.
  - Thunderbolt's **PCIe tunneling and DisplayPort streams** rely on synchronized packet transfers.

- **Why Synchronous?**
  - Ensures **low-latency** and **real-time** data delivery.
  - Avoids **timing mismatches** in high-speed data transmission.
  - Essential for **daisy-chained devices**, where multiple devices rely on a common clock.

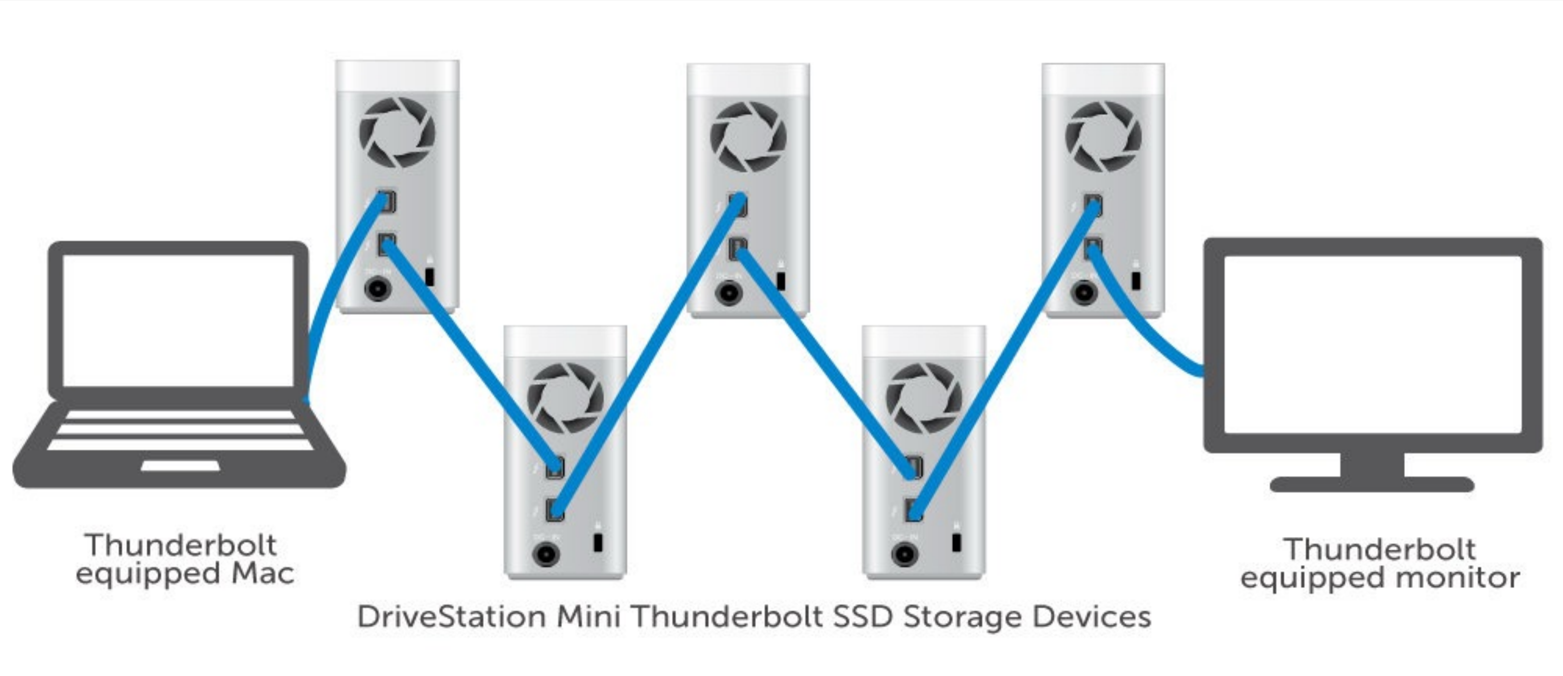
Multi-Device Connectivity in Thunderbolt

- The Thunderbolt protocol supports multiple device connections. It allows multiple peripherals (such as external GPUs, storage devices, and displays) to be connected in two ways:
  - **Daisy-Chaining**
  - **Hub-Based Connection**

Daisy-Chaining in Thunderbolt

- Thunderbolt natively supports daisy-chaining.
- Up to **6 devices** can be connected in a chain using a single Thunderbolt port.
- Devices are connected **serially**, meaning each device acts as a relay for the next.
- A single cable carries both PCIe and DisplayPort signals.

How Daisy-Chaining Works:



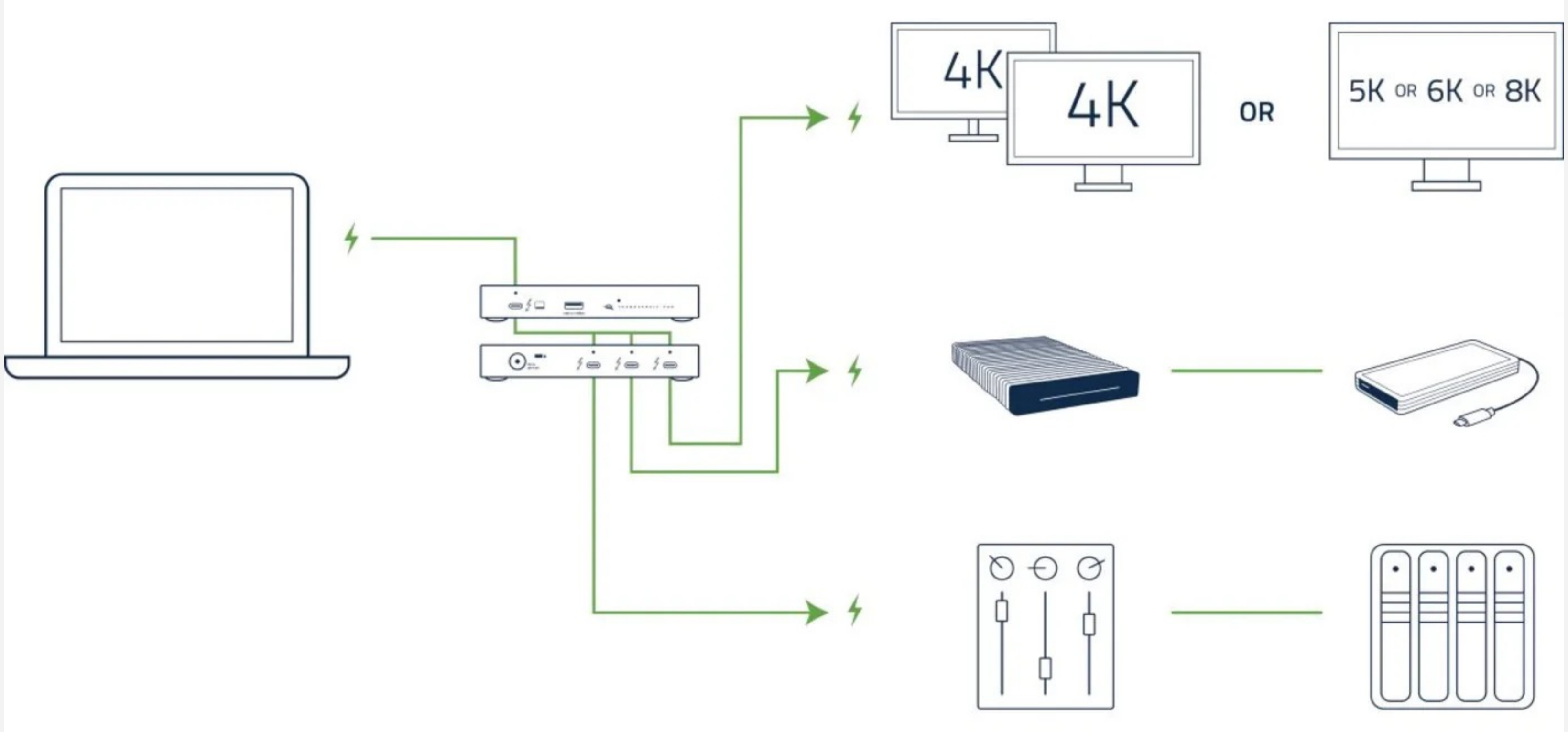
Thunderbolt Daisy-Chaining

- **How daisy-chaining works:**
  1. The host device (computer) has a Thunderbolt controller.
  2. The first device connects to the host using a Thunderbolt cable.
  3. Each additional device is connected to the previous device rather than the host.
  4. The last device in the chain does not need to support Thunderbolt output.

- **Daisy-Chaining limitations:**
  - Not all devices support daisy-chaining (e.g., some USB-C devices).
  - If a device in the middle of the chain fails, all devices connected after it lose connection.
  - Bandwidth is **shared** among all devices, potentially limiting performance.

Thunderbolt Hub-Based Connection

- With **Thunderbolt 4**, multi-device connection is more efficient using a **Thunderbolt hub**.
    - Unlike daisy-chaining, **each device connects directly to a central hub.**
    - All devices receive **full bandwidth** without relying on the previous device.

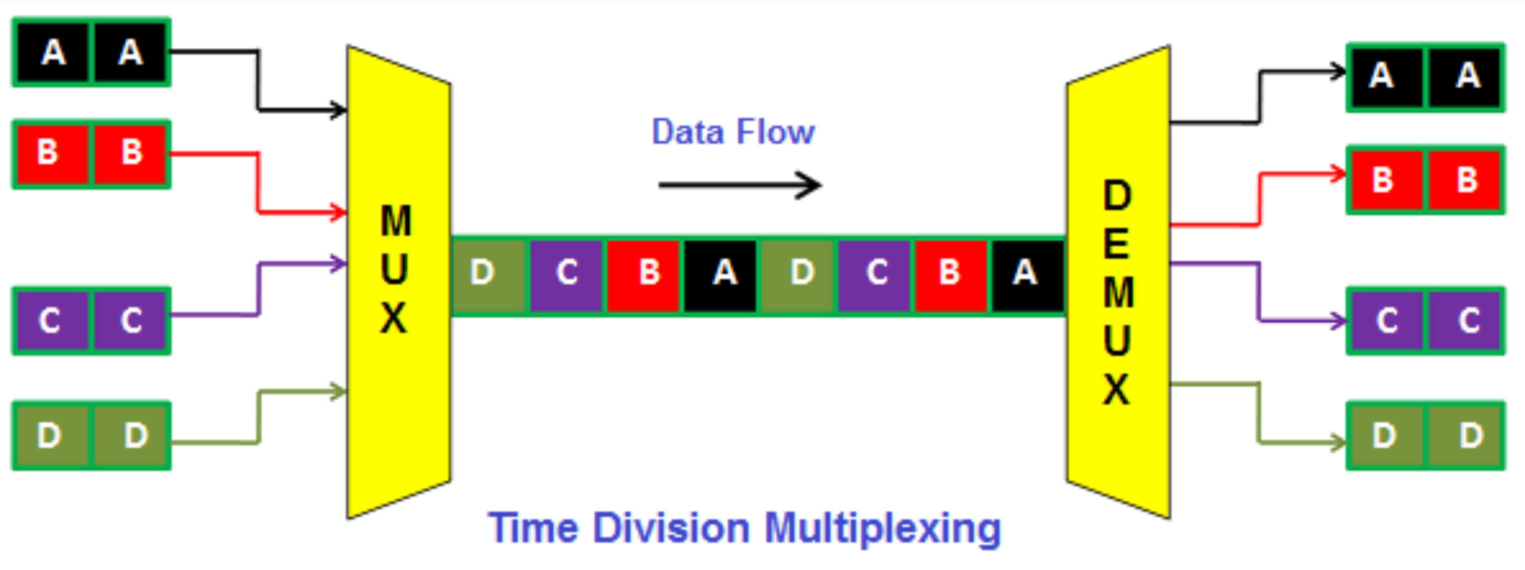

4K
4K
OR
5K OR 6K OR 8K

Thunderbolt Hub-Based Connection

- Advantages of Hub-Based Connection
  - **More reliable** (a failure in one device does not affect others).
  - **No need for Thunderbolt pass-through support** in connected devices.
  - **Better bandwidth allocation** since each device gets a dedicated connection.

Collision Management in Thunderbolt

- Since Thunderbolt allows multiple devices to communicate over a **single data link**, there must be an efficient method to handle **data collisions** and **bandwidth allocation**.
- Thunderbolt uses a **time-division multiplexing (TDM)** approach for handling multiple data streams **without collisions**.
  - **Thunderbolt Controller** manages data traffic using a **centralized arbitration system**.
  - **Each device gets a dedicated time slot**, preventing simultaneous data transmission collisions.
  - Devices **request bandwidth dynamically**, and the controller assigns it based on priority.


A A
B B
C C
D D
MUX
Data Flow
D C B A D C B A
DEMUX
A A
B B
C C
D D
Time Division Multiplexing

Flow Control in Thunderbolt

- Thunderbolt flow control is managed through **three main techniques**:
  - **Time-Division Multiplexing (TDM) for Bandwidth Allocation**
  - **Credit-Based Flow Control for PCIe Tunneling**
  - **Frame Buffering for DisplayPort Streams**

- These mechanisms ensure that different types of traffic can share the Thunderbolt link **efficiently** without overwhelming the system.

Error Detection in Thunderbolt Protocol

- Thunderbolt, like many high-speed communication protocols, incorporates error detection and correction mechanisms across multiple layers to ensure data integrity. These mechanisms exist in:
    - **Physical Layer (Bit-Level Error Detection)**
    - **Data Link Layer (Frame-Level Error Detection)**
    - **Transport & Protocol Layers (Higher-Level Error Detection)**

- Each layer has specific techniques to detect and correct transmission errors, ensuring reliability across PCIe, DisplayPort, and USB tunneling.

Error Detection in the Physical Layer

- **8B/10B Encoding (Thunderbolt 1 & 2) / 128B/130B Encoding (Thunderbolt 3 & 4)**
    - Adds **redundant bits** for detecting invalid bit patterns.
    - Ensures **signal synchronization and bit recovery**.

- **CRC (Cyclic Redundancy Check) for Physical Frames**
    - Each transmitted block has a **CRC checksum**.
    - The receiver **recalculates the CRC** and checks for mismatches.

- **Error Handling**
    - If an error is detected, the receiver **requests retransmission** or **drops the corrupted data**.

Error Detection in the Data Link Layer

- **CRC for Data Packets**
  - Thunderbolt packets include a **CRC field** that protects against transmission errors.
  - If a CRC mismatch occurs, the frame is **discarded and retransmitted**.

- **Flow Control to Prevent Data Loss**
  - Credit-based **flow control** ensures that data is only sent when **buffer space is available**.
  - This reduces **buffer overflows** that can cause packet corruption.

Error Detection Mechanism in Higher Layers

- **PCIe Error Detection and Correction**
  - Uses **End-to-End CRC** to check for errors in PCIe packets.
  - Implements **Automatic Retransmission Requests (ARQ)** if errors are detected.
- **DisplayPort Frame Buffering & Error Checking**
  - Uses **frame buffering** to detect dropped frames.
  - Monitors **video stream consistency** to detect frame errors.
- **USB Protocol Error Checking**
  - USB tunneling uses **error-checking codes** at the transport layer.
  - If an error is detected, USB **requests retransmission** or drops corrupt data.

Types of Messages in Thunderbolt Protocol

- Thunderbolt communication involves several types of messages, each serving a distinct function within the system. These messages are transmitted across different layers, including:
  - **Physical Layer**
  - **Data Link Layer**
  - **Transport Layer**.

- The types of messages can be grouped into:
  - **Control Messages**
  - **Data Messages**
  - **Error Handling Messages**.

Control Messages

- Control messages are used for **management and configuration** of the Thunderbolt link. They include handshakes, link initialization, and link maintenance.

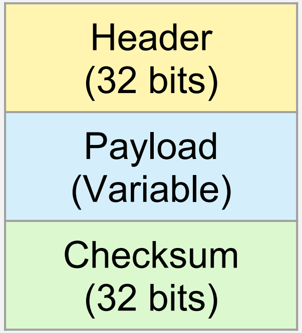
- **Message Types:**
  - **Link Initialization Messages**: Used during the establishment of the connection to configure the link.
  - **Link Maintenance Messages**: Used to manage the state of the link (e.g., link up, link down).
  - **Hot Plug Events**: Notifications when a new device is inserted or removed.
  - **Power Management Messages**: Control the power state of devices.

Message Type (8 bits)
Message Length (8 bits)
Message Data (Variable)
Checksum (32 bits)

Data Messages

- Data messages carry **user data** across the Thunderbolt connection, such as files or streaming video. These messages are often transmitted using **tunneling protocols** such as PCIe, DisplayPort, and USB.

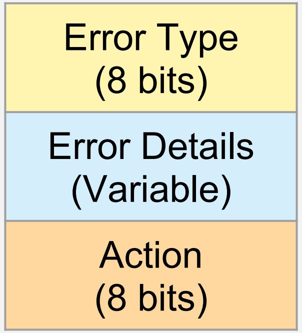
- **Message Types:**
  - **PCIe Data Messages**: Used to transport data from the host to peripheral devices and vice versa.
  - **DisplayPort Data Messages**: Carry video signals.
  - **USB Data Messages**: Carry USB data packets.
  - **Isochronous Data**: Used for time-sensitive data streams like video or audio.



Error Handling Messages

- Error handling messages are used to manage errors in the system, such as data corruption or transmission failures. These messages ensure that the system can recover from issues by requesting retransmissions or adjusting configurations.

- **Message Types:**
  - **Error Detection**: Messages indicating errors, such as corrupted frames or invalid packets.
  - **Error Recovery**: Messages requesting retransmission of corrupted or lost data.
  - **Flow Control**: Messages that manage the flow of data to avoid congestion and prevent errors.



References

- https://www.intel.com/content/dam/doc/technology-brief/thunderbolt-technology-brief.pdf
- https://www.thunderbolttechnology.net/sites/default/files/Thunderbolt3_TechBrief_FINAL.pdf
- https://www.intel.com/content/www/us/en/gaming/resources/upgrade-gaming-accessories-thunderbolt-4.html
- https://www.thunderbolttechnology.net/sites/default/files/Thunderbolt_3_Overview_Brief_FINAL.pdf
- https://www.erpublications.com/uploaded_files/download/download_01_04_2014_11_41_43.pdf
- https://www.ti.com.cn/cn/lit/ds/symlink/hd3ss0001.pdf?ts=1738578627921&ref_url=https%253A%252F%252Fwww.google.com%252F

Thank you for your attention