Compute Express Link™ (CXL™): An Open Industry Standard for Composable Computing

Tutorial for Flash Memory Summit

August 2023

Presented by: Debendra Das Sharma[1], Mahesh Wagh[2], Vincent Hache[3], and Mahesh Natu[1]

[1]: Intel Corporation, [2]: AMD Corporation, [3]: Rambus Inc

Alibaba Group AMD arm cisco DELL EMC

Google Hewlett Packard Enterprise HUAWEI IBM intel

Meta Microsoft Micron NVIDIA SAMSUNG

CXL Board of Directors

CXL Compute Express Link™

Industry Open Standard for High Speed Communications

250+ Member Companies

Flash Memory Summit

CXL Compute Express Link™

CXL Specification Release Timeline

- **March 2019**
  - CXL 1.0 Specification Released
- **September 2019**
  - CXL Consortium Officially Incorporates
  - CXL 1.1 Specification Released
- **November 2020**
  - CXL 2.0 Specification Released
- **August 2022**
  - CXL 3.0 Specification Released

Agenda

- Introduction to CXL
- Protocols: Memory Pooling and Sharing
- Software and Error Management
- Fabric and Fabric Management
- Conclusions

Industry Landscape

Proliferation of Cloud Computing

Growth of AI & Analytics

Cloudification of the Network & Edge

CXL Overview

- New breakthrough high-speed fabric
  - Enables a high-speed, efficient interconnect between CPU, memory and accelerators
  - Builds upon PCI Express® (PCIe®) infrastructure, leveraging the PCIe physical and electrical interface
  - Maintains memory coherency between the CPU memory space and memory on CXL attached devices
    - Enables fine-grained resource sharing for higher performance in heterogeneous compute environments
    - Enables memory disaggregation, memory pooling and sharing, persistent memory and emerging memory media
- Delivered as an open industry standard
  - The CXL 3.0 Specification was released in August with full backward compatibility with CXL 1.1
  - Future CXL Specification generations will include continuous innovation to meet industry needs and support new technologies

CXL™ Approach

Coherent Interface

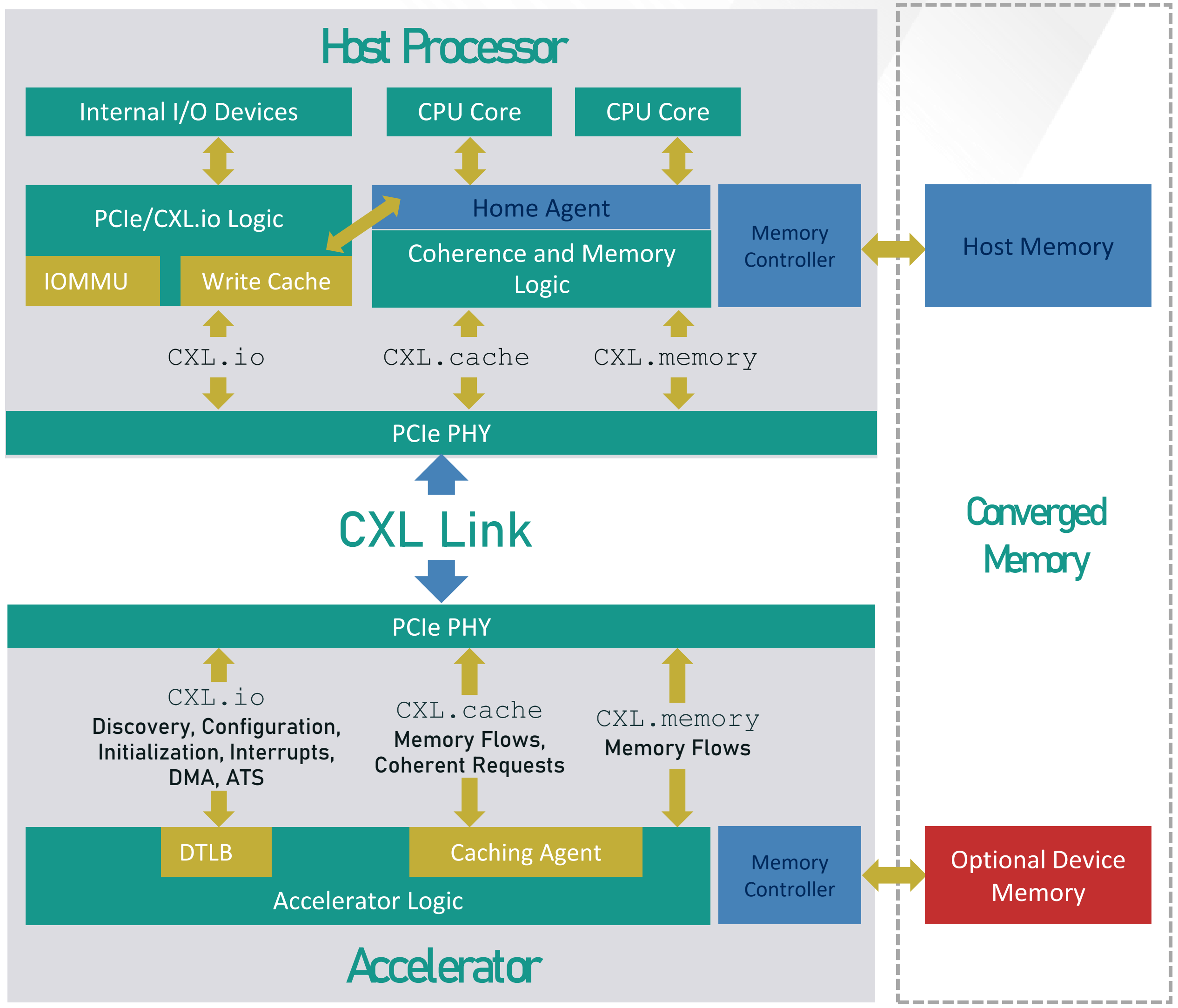
- Leverages PCIe® with three multiplexed protocols
- Built on top of PCIe infrastructure

Low Latency

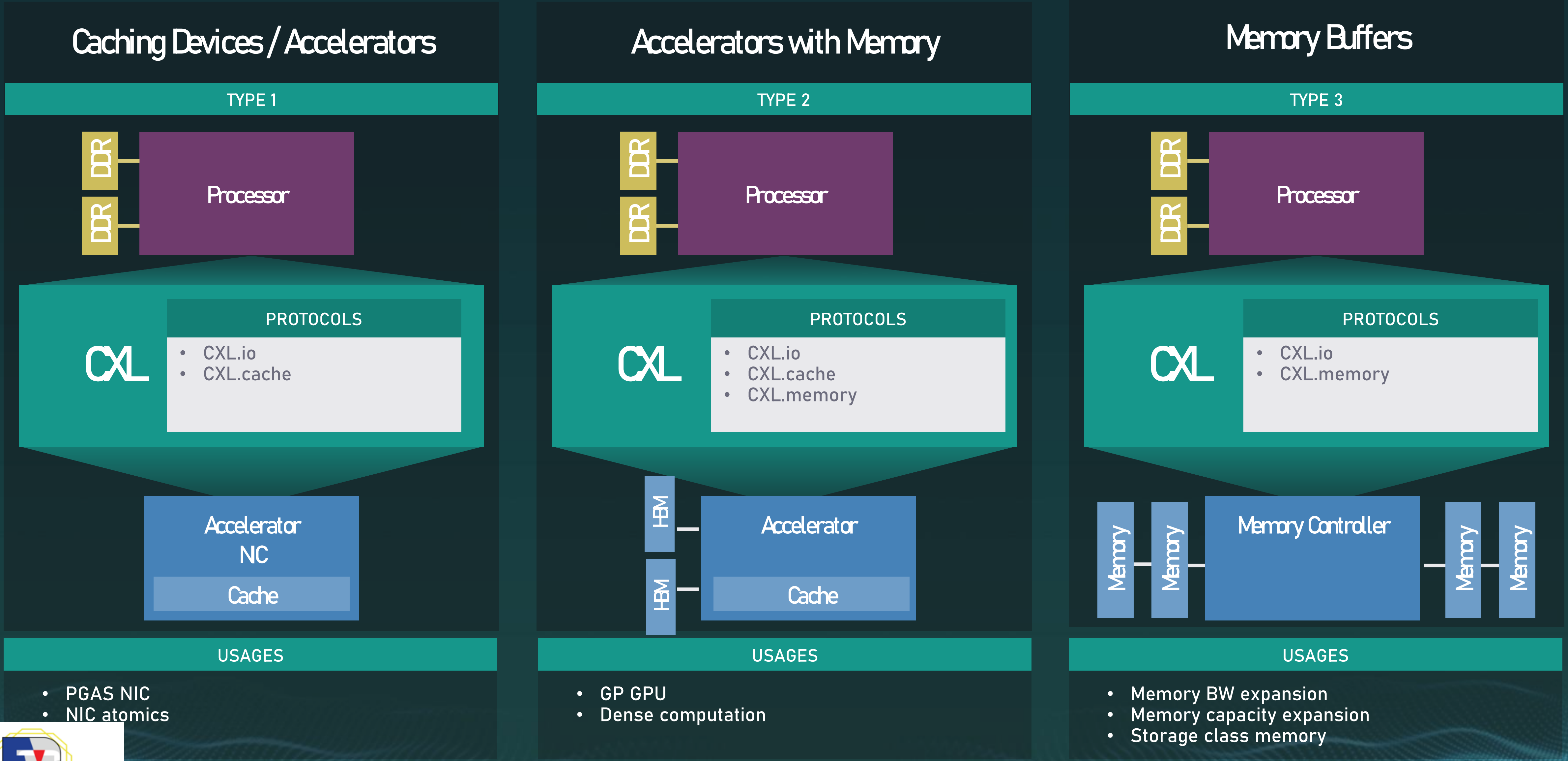
- `CXL.Cache/CXL.Memory` targets near CPU cache coherent latency (<200ns load to use)

Asymmetric Complexity

- Eases burdens of cache coherence interface designs for devices



Representative CXL Usages with CXL 1.0



Caching Devices / Accelerators (TYPE 1)

USAGES

- PGAS NIC
- NIC atomics

Accelerators with Memory (TYPE 2)

USAGES

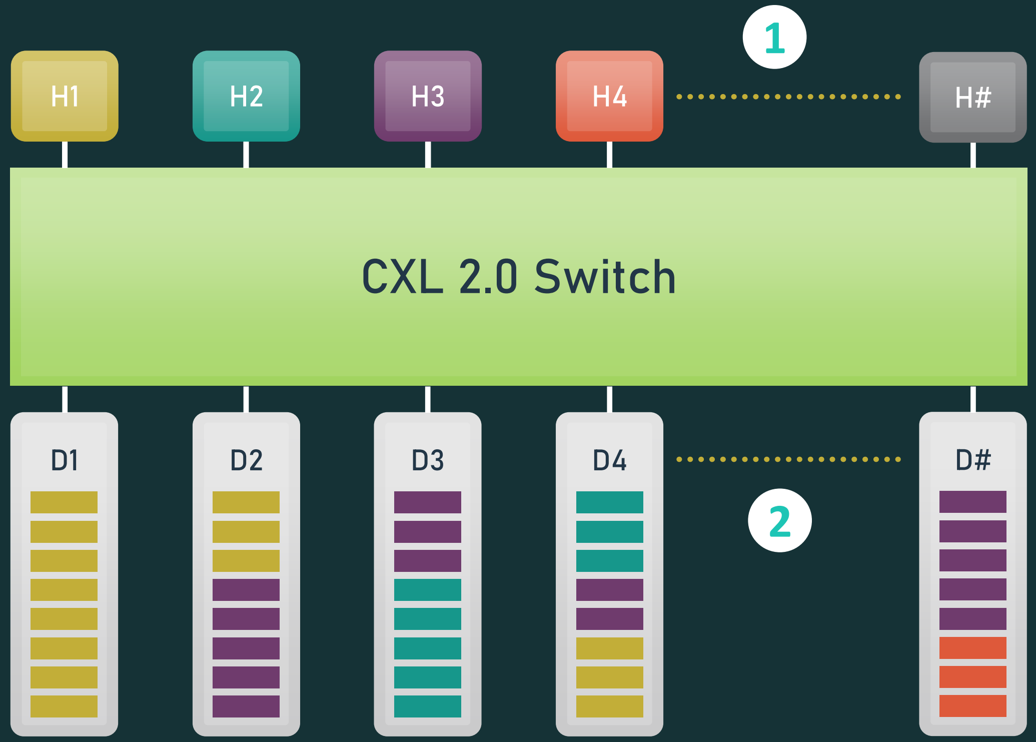
- GP GPU
- Dense computation

Memory Buffers (TYPE 3)

USAGES

- Memory BW expansion
- Memory capacity expansion
- Storage class memory

CXL 2.0 Feature Summary
MEMORY POOLING



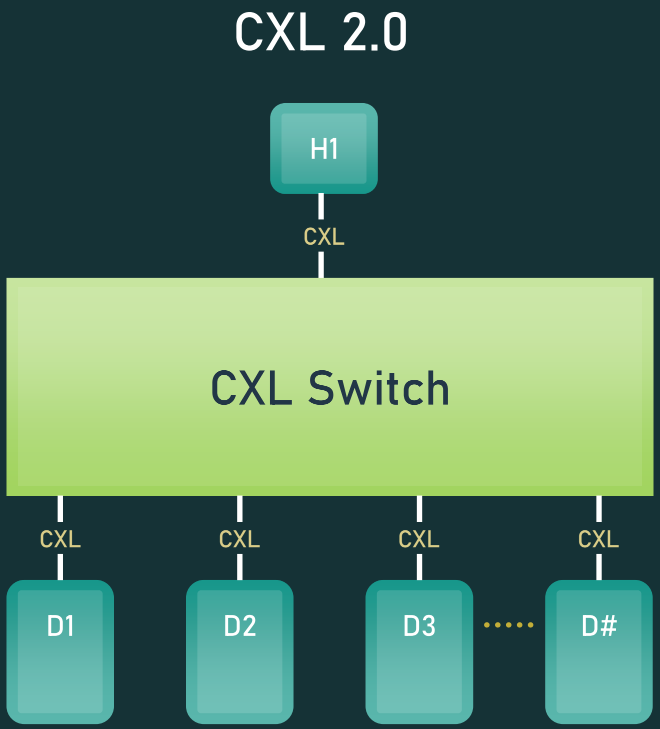
1. Device memory can be allocated across multiple hosts.
2. Multi Logical Devices allow for finer grain memory allocation

Flash Memory Summit

Compute Express Link™ and CXL™ are trademarks of the Compute Express Link Consortium.

CXL 2.0 Feature Summary

SWITCH CAPABILITY



- Supports single-level switching
- Enables memory expansion and resource allocation

Compute Express Link™ and CXL™ are trademarks of the Compute Express Link Consortium.

CXL™ 2.0: Resource Pooling at Rack Level, Persistent Memory Support and Enhanced Security

- Resource pooling/disaggregation
  - Managed hot-plug flows to move resources
  - Type-1/Type-2 device assigned to one host
  - Type-3 device (memory) pooling at rack level
  - Direct load-store, low-latency access – similar to memory attached in a neighboring CPU socket (vs. RDMA over network)
- Persistence flows for persistent memory
- Fabric Manager/API for managing resources
- Security: authentication, encryption
- Beyond node to rack-level connectivity!

Disaggregated system with CXL optimizes resource utilization delivering lower TCO and power efficiency

Flash Memory Summit

CXL 3.0 Specification

Industry trends

- Use cases driving need for higher bandwidth include: high performance accelerators, system memory, SmartNIC and leading edge networking
- CPU efficiency is declining due to reduced memory capacity and bandwidth per core
- Efficient peer-to-peer resource sharing across multiple domains
- Memory bottlenecks due to CPU pin and thermal constraints

CXL 3.0 introduces…

- Fabric capabilities
  - Multi-headed and fabric attached devices
  - Enhance fabric management
  - Composable disaggregated infrastructure
- Improved capability for better scalability and resource utilization
  - Enhanced memory pooling
  - Multi-level switching
  - New symmetric coherency capabilities
  - Improved software capabilities
- Double the bandwidth
- Zero added latency over CXL 2.0
- Full backward compatibility with CXL 2.0, CXL 1.1, and CXL 1.0

CXL 3.0 Specification

Fabric capabilities and management

Improved memory sharing and pooling

Enhanced coherency

Peer-to-peer

Expanded capabilities for increasing scale and optimizing resource utilization

- Fabric capabilities and fabric attached memory
- Enhance fabric management framework
- Memory pooling and sharing
- Peer-to-peer memory access
- Multi-level switching
- Near memory processing
- Multi-headed devices
- Multiple Type 1/Type 2 devices per root port
- Fully backward compatible to CXL 2.0, 1.1, and 1.0
- Supports PCIe® 6.0

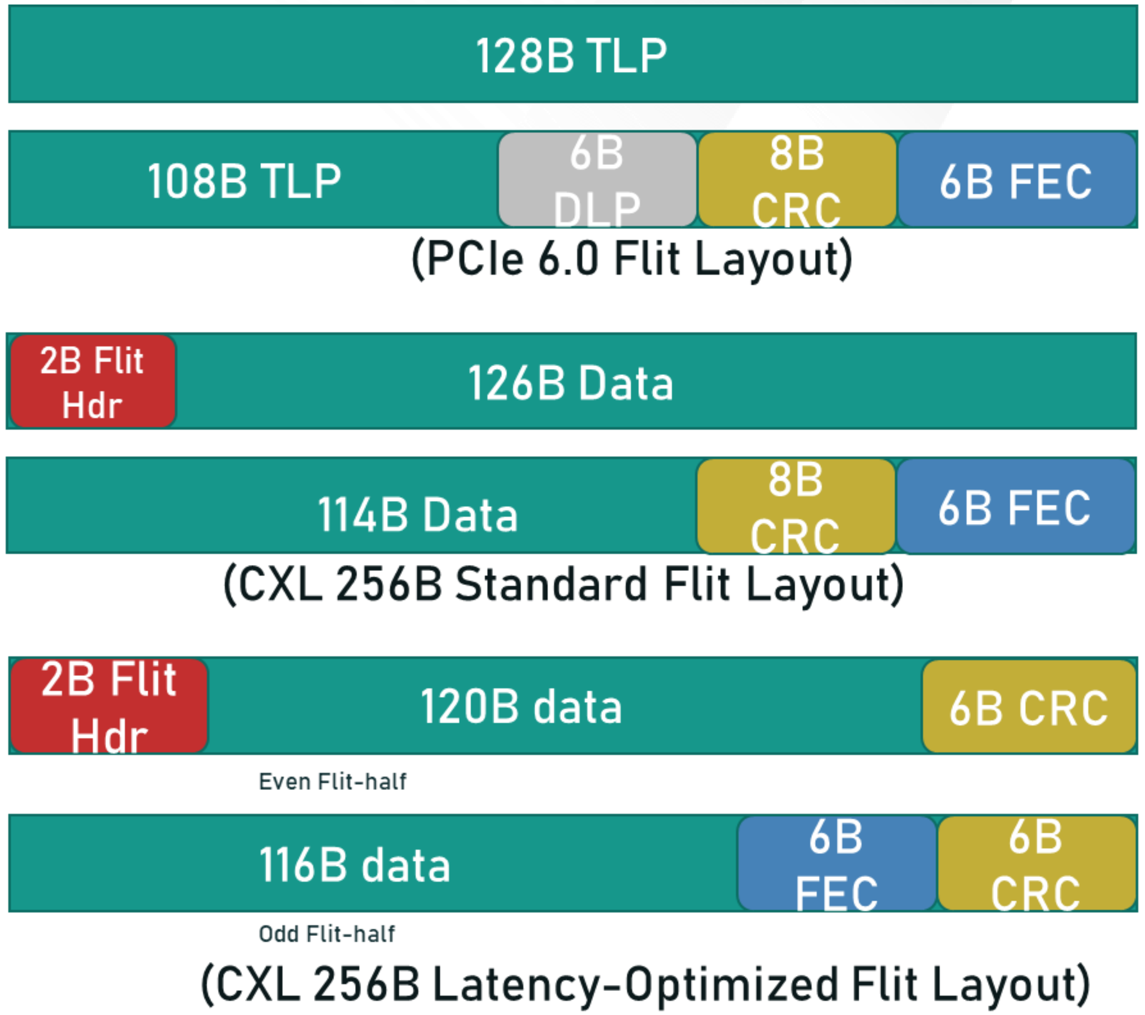
CXL 3.0 Spec Feature Summary

Features	CXL 1.0 / 1.1	CXL 2.0	CXL 3.0
Release date	2019	2020	1H 2022
Max link rate	32GTs	32GTs	64GTs
Flit 68 byte (up to 32 GTs)	✓	✓	✓
Flit 256 byte (up to 64 GTs)			✓
Type 1, Type 2 and Type 3 Devices	✓	✓	✓
Memory Pooling w/ MLDs		✓	✓
Global Persistent Flush		✓	✓
CXL IDE		✓	✓
Switching (Single-level)		✓	✓
Switching (Multi-level)			✓
Direct memory access for peer-to-peer			✓
Enhanced coherency (256 byte flit)			✓
Memory sharing (256 byte flit)			✓
Multiple Type 1/Type 2 devices per root port			✓
Fabric capabilities (256 byte flit)			✓

	Not supported
✓	Supported

CXL 3.0: Doubles Bandwidth with Same Latency

- Uses PCIe® 6.0 PHY @ 64 GT/s
- PAM-4 and high BER mitigated by PCIe 6.0 FEC and CRC (different CRC for latency optimized)
- Standard 256B Flit along with an additional 256B Latency Optimized Flit (0-latency adder over CXL 2)
  - 0-latency adder trades off FIT (failure in time, 109 hours) from 5x10-8 to 0.026 and Link efficiency impact from 0.94 to 0.92 for 2-5ns latency savings (x16 – x4)1
- Extends to lower data rates (8G, 16G, 32G)
- Enables several new CXL 3 protocol enhancements with the 256B Flit format



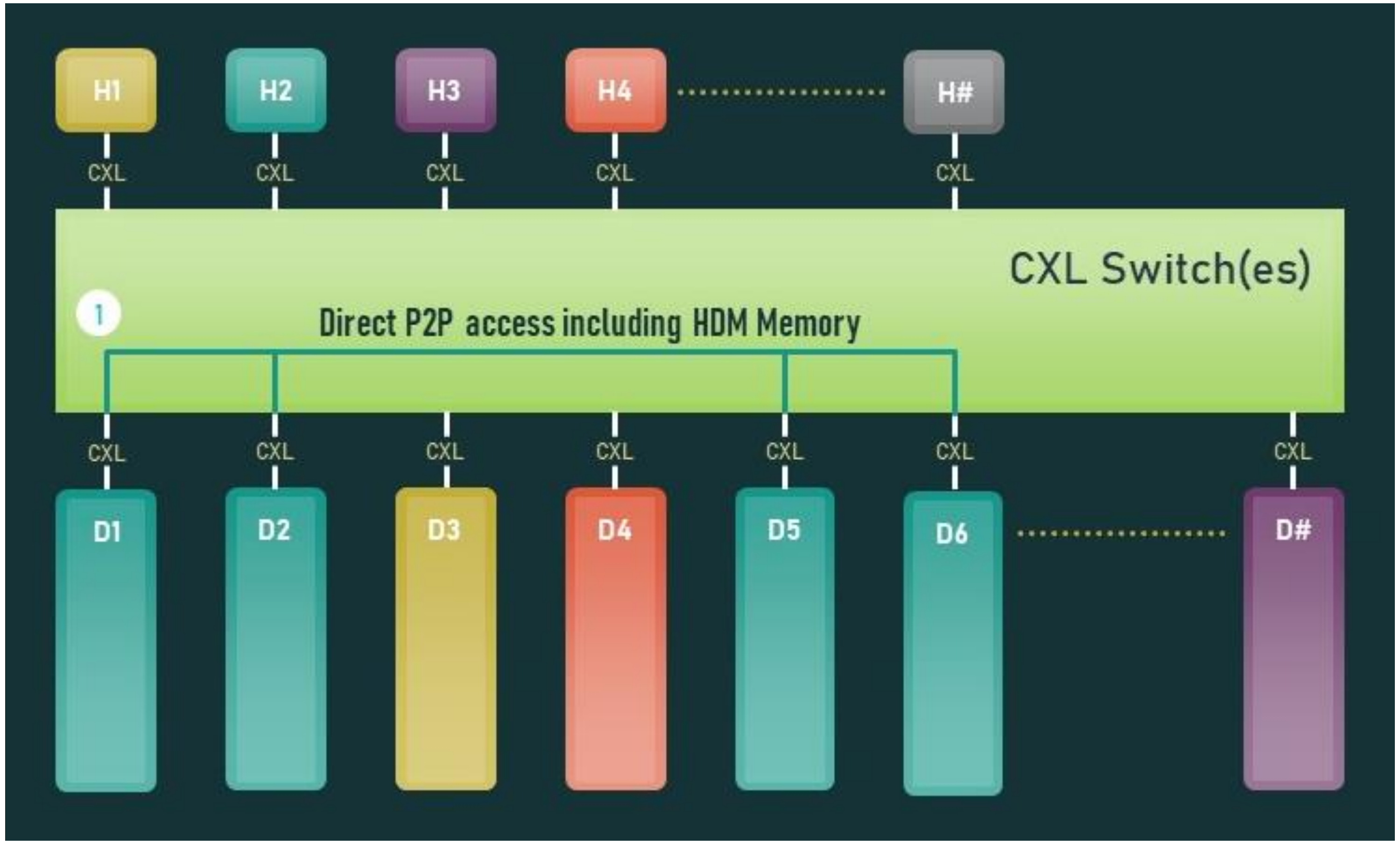
(PCIe 6.0 Flit Layout)

(CXL 256B Standard Flit Layout)

(CXL 256B Latency-Optimized Flit Layout)

1: D. Das Sharma, "A Low-Latency and Low-Power Approach for Coherency and Memory Protocols on PCI Express 6.0 PHY at 64.0 GT/s with PAM-4 Signaling", IEEE Micro, Mar/ Apr 2022 (https://ieeexplore.ieee.org/document/9662217)

CXL 3.0 Protocol Enhancements (UIO and BI) for Device to Device Connectivity



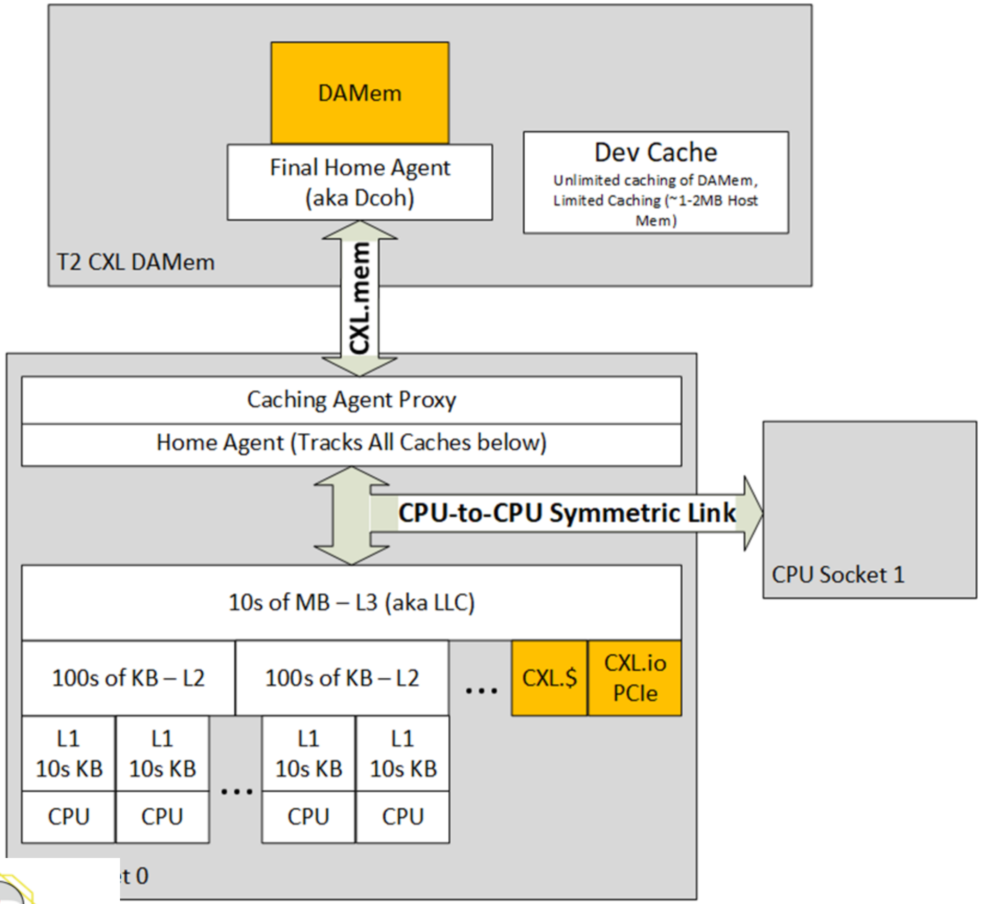
CXL 3.0 enables non-tree topologies and peer-to-peer communication (P2P) within a virtual hierarchy of devices

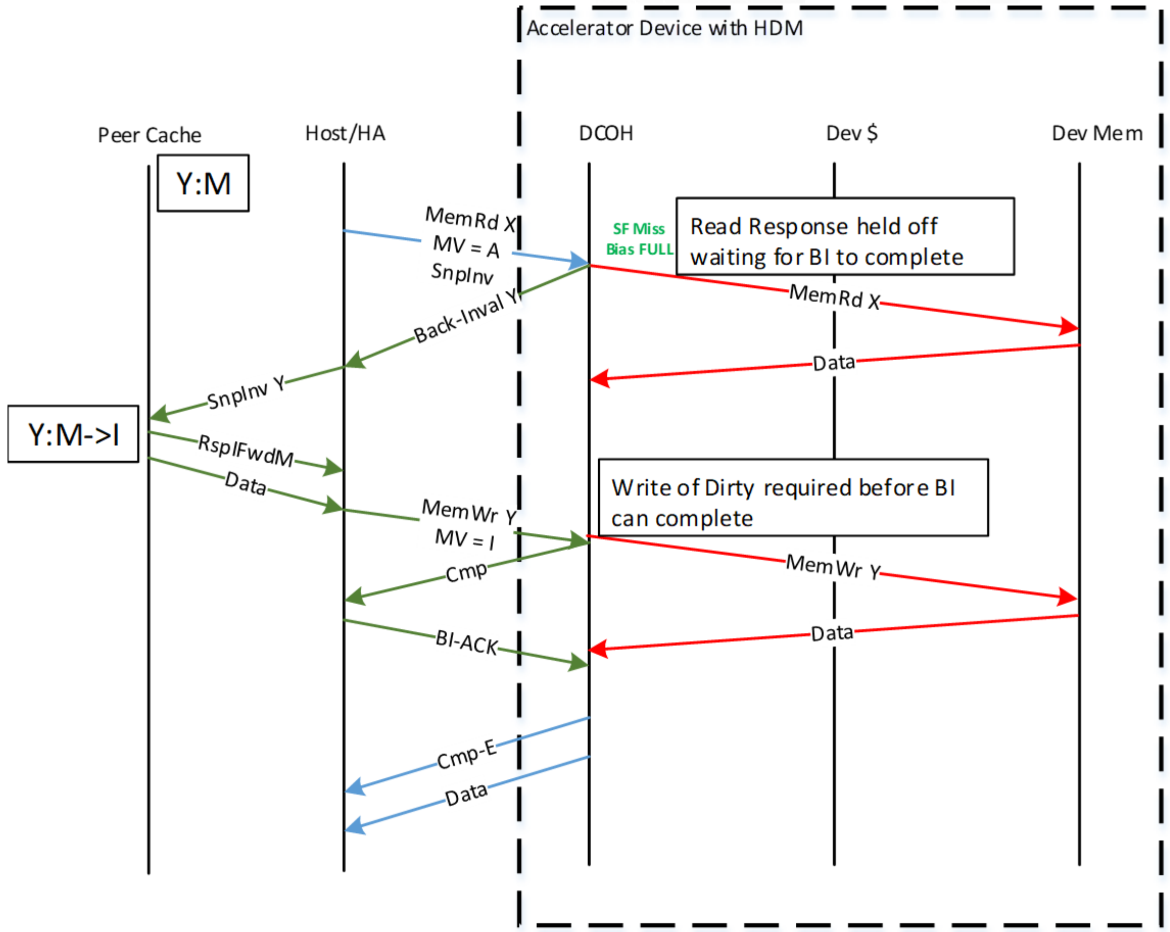
- Virtual hierarchies are associations of devices that maintains a coherency domain

- P2P to HDM-DB memory is I/O Coherent: a new Unordered I/O (UIO) Flow in CXL.io – the Type-2/3 device that hosts the memory will generate a new Back-Invalidation flow (CXL.Mem) to the host to ensure coherency if there is a coherency conflict

CXL 3.0 Protocol Enhancements: Mapping Large memory in Type-2 Devices to HDM with Back Invalidate

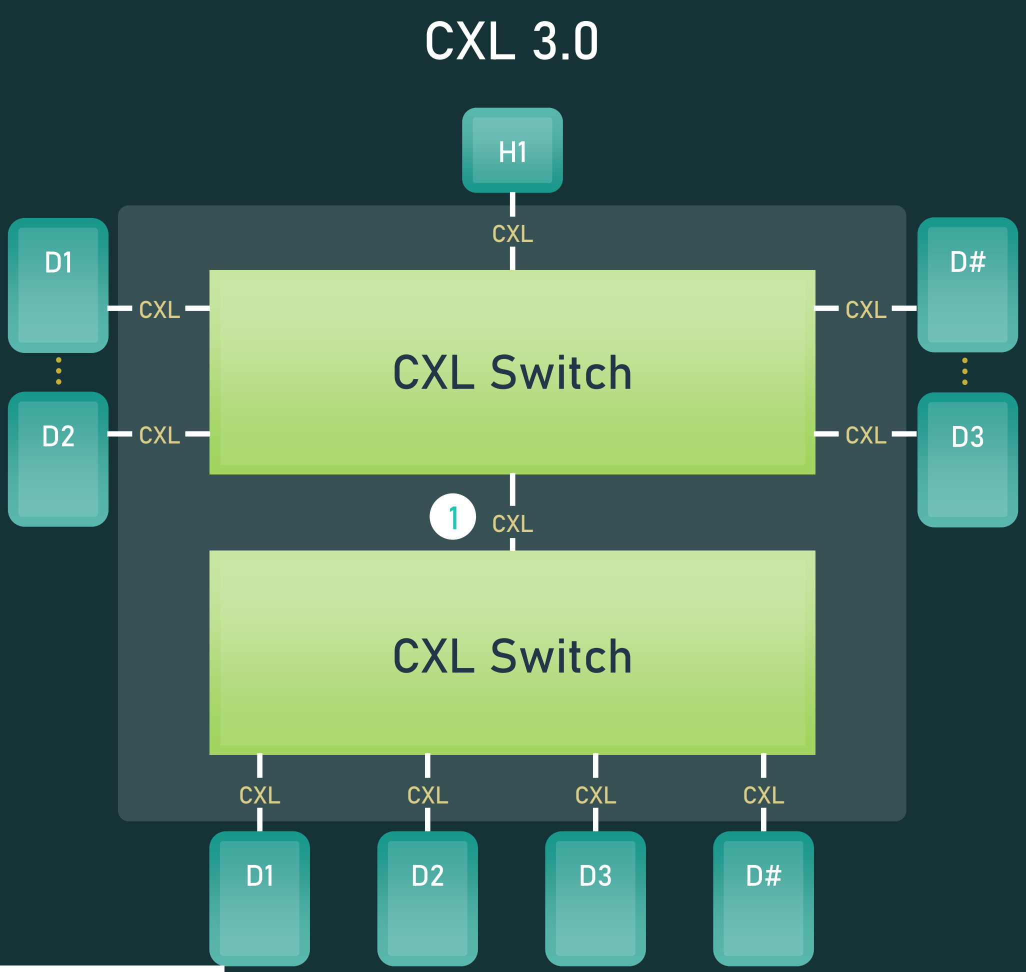
Existing Bias – Flip mechanism needed HDM to be tracked fully since device could not back snoop the host. Back Invalidate with CXL 3.0 enables snoop filter implementation resulting in large memory that can be mapped to HDM

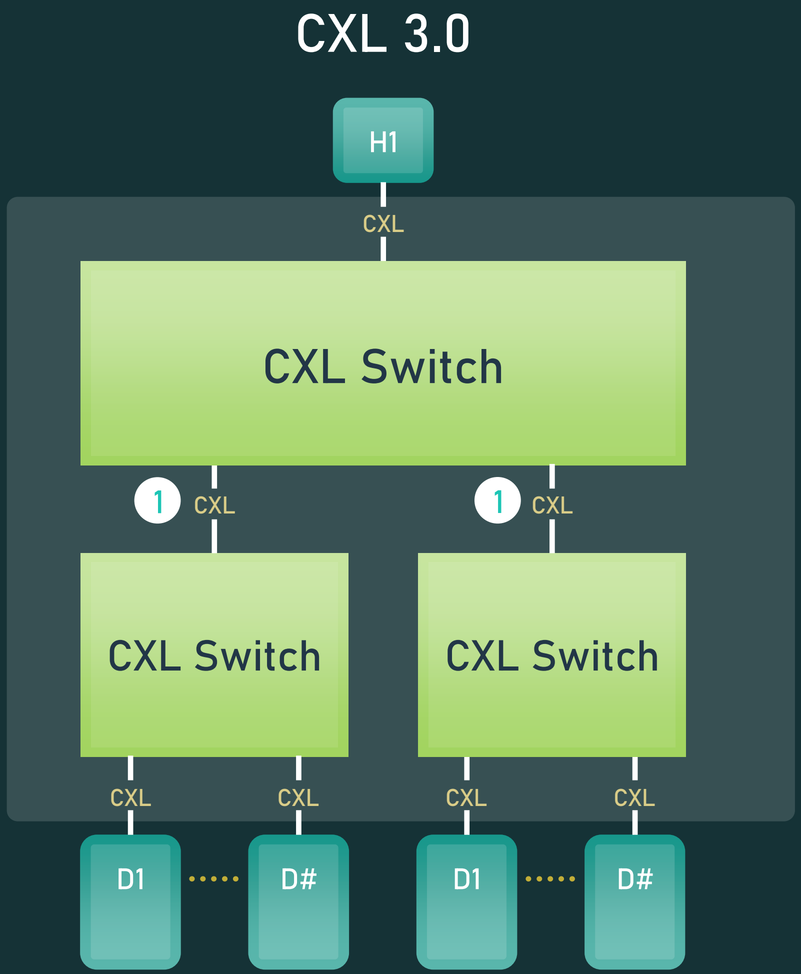




CXL 3.0: SWITCH CASCADE/FANOUT

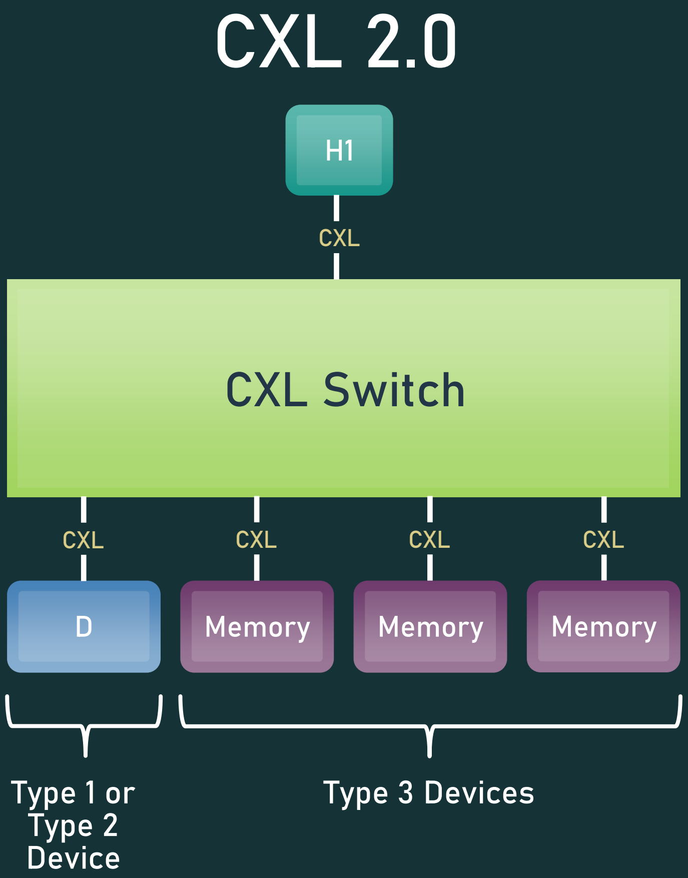
Supporting vast array of switch topologies

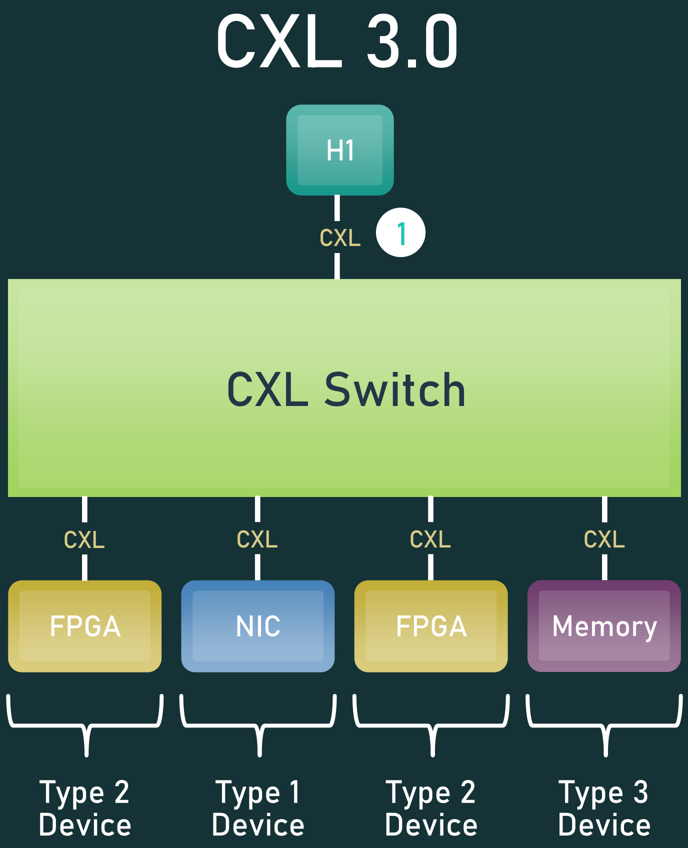




1 Multiple switch levels (aka cascade)
- Supports fanout of all device types

CXL 3.0: MULTIPLE DEVICES OF ALL TYPES PER ROOT PORT

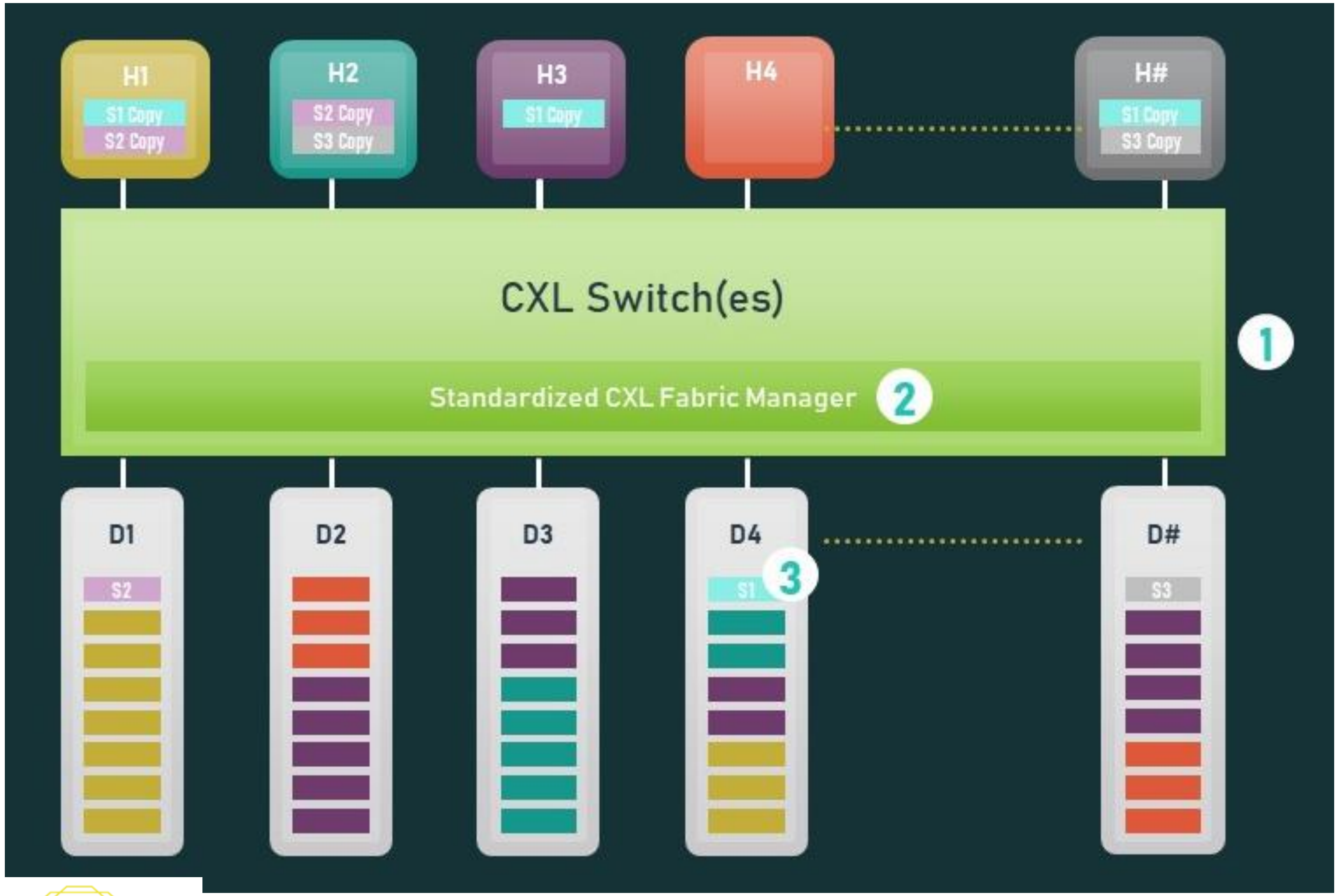




1 Each host's root port can connect to more than one device type

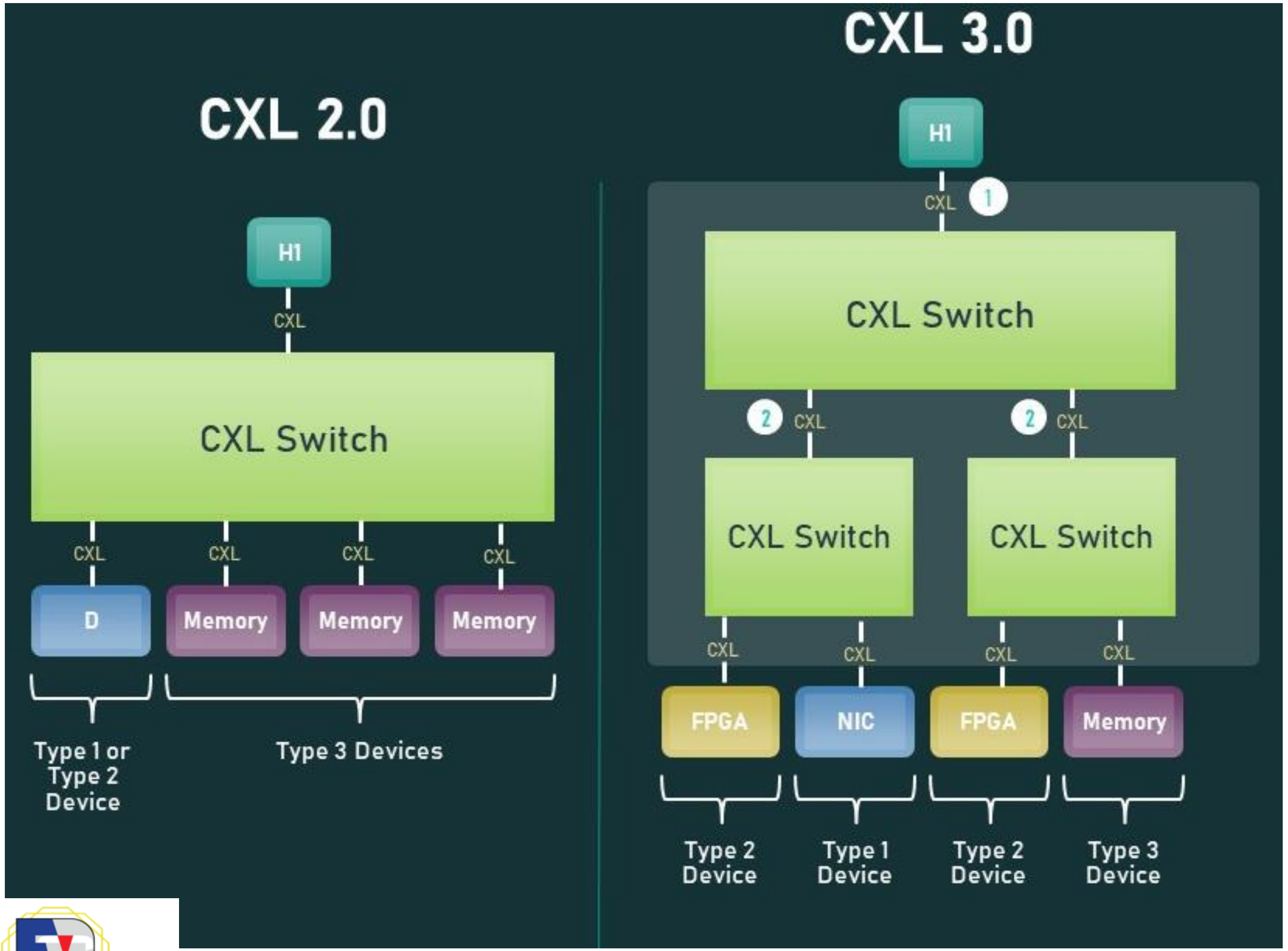
Flash Memory Summit

CXL 3.0: Pooling & Sharing



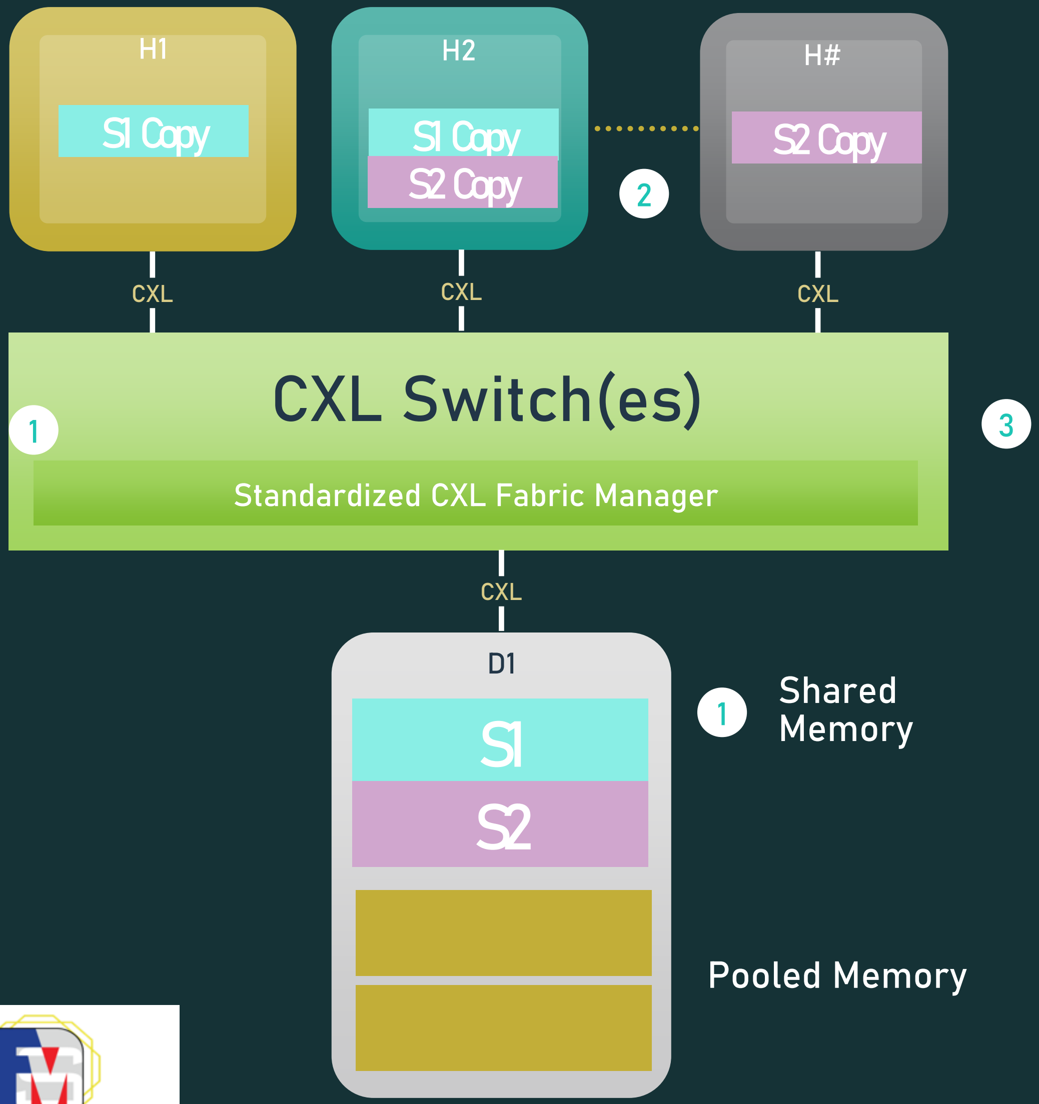
1. Expanded use case showing memory sharing and pooling
2. CXL Fabric Manager is available to setup, deploy, and modify the environment
3. Shared Coherent Memory across hosts using hardware coherency (directory + Back-Invalidate Flows). Allows one to build large clusters to solve large problems through shared memory constructs. Defines a Global Fabric Attached Memory (GFAM) which can provide access to up to 4095 entities

CXL 3.0: Multiple Level Switching, Multiple Type-1/2



1. Each host's root port can connect to more than one device type (up to 16 CXL.cache devices)
2. Multiple switch levels (aka cascade)
   - Supports fanout of all device types

CXL 3.0: COHERENT MEMORY SHARING



1. Device memory can be shared by all hosts to increase data flow efficiency and improve memory utilization

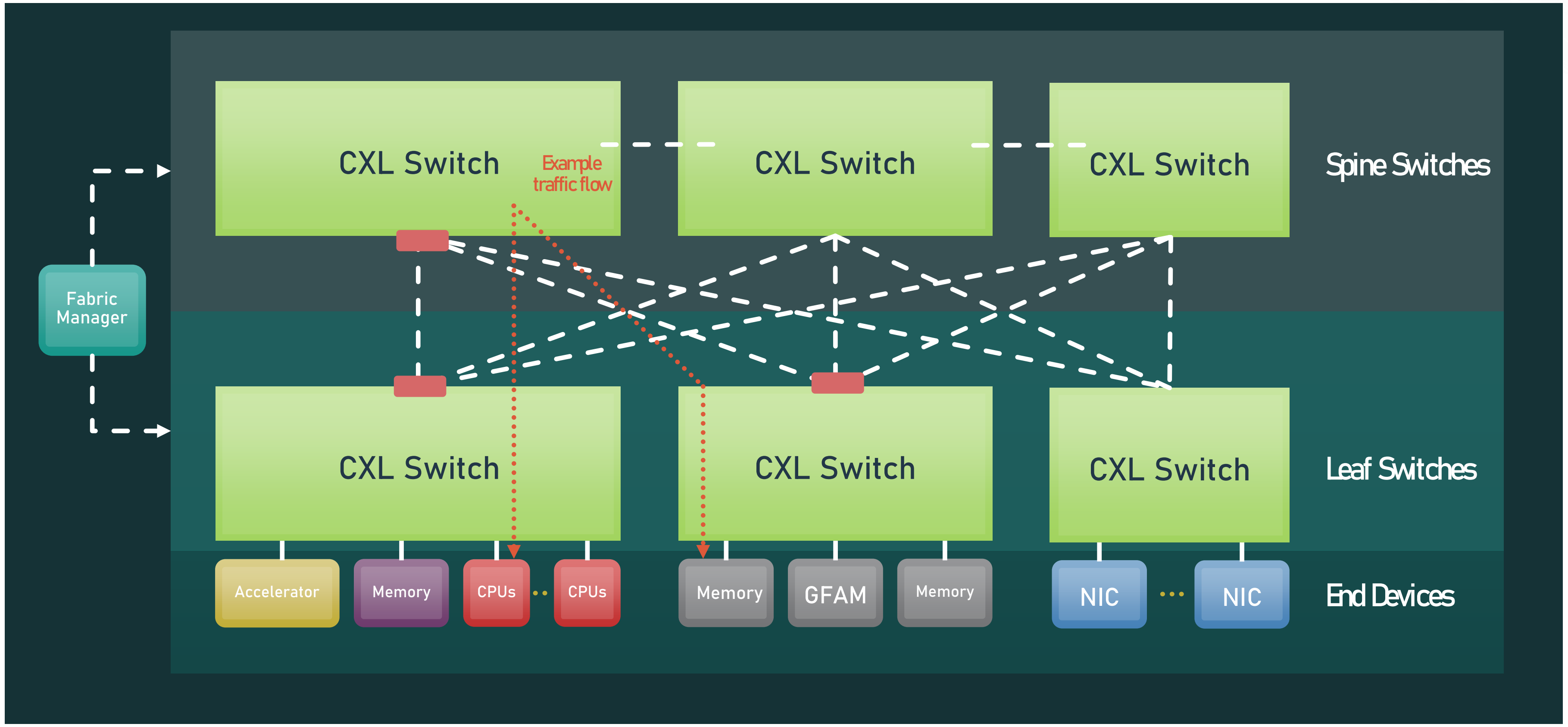
2. Host can have a coherent copy of the shared region or portions of shared region in host cache

3. CXL 3.0 defined mechanisms to enforce hardware cache coherency between copies

Flash Memory Summit

CXL 3.0 Fabrics

Composable Systems with Spine/Leaf Architecture at Rack/ Pod Level



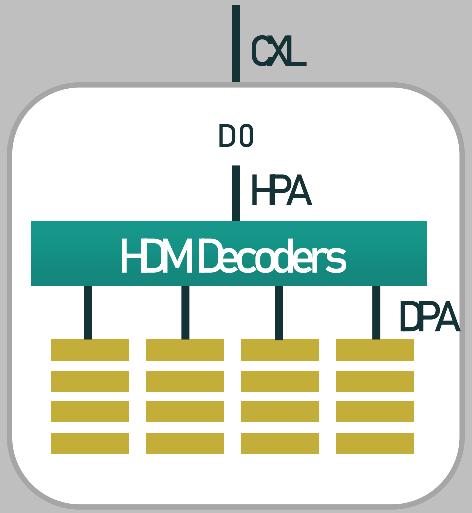
CXL 3.0 Fabric Architecture

- Interconnected Spine Switch System
- Leaf Switch NIC Enclosure
- Leaf Switch CPU Enclosure
- Leaf Switch Accelerator Enclosure
- Leaf Switch Memory Enclosure

Protocol: Memory Pooling and Sharing

CXL SLD Memory Device

- Support one or more PCIe® Endpoint Functions
  - Type 0 header in PCI Configuration space
- Primary function (device number 0, function number 0) must carry one instance of CXL DVSEC ID 0 with Revision 1 or greater.
- Non-CXL Function Map DVSEC to advertise Non-CXL functions
- Must support operating in CXL 1.1 mode
  - PCIe Endpoint → RCIEP
- Type 3 device Component Register Block includes HDM Decoder registers
- Connected to a Single Virtual Hierarchy

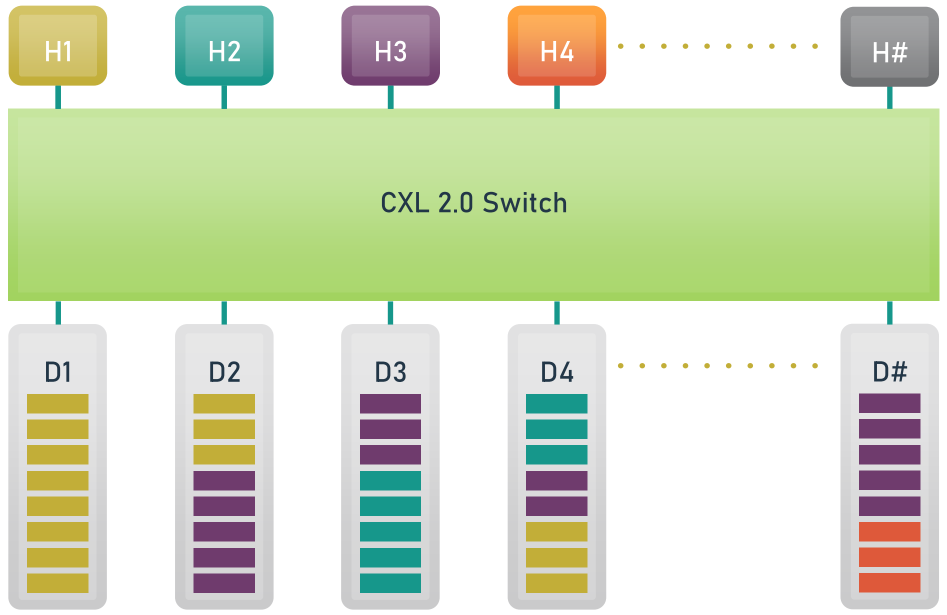


HPA – Host Physical Address
DPA – Device Physical Address

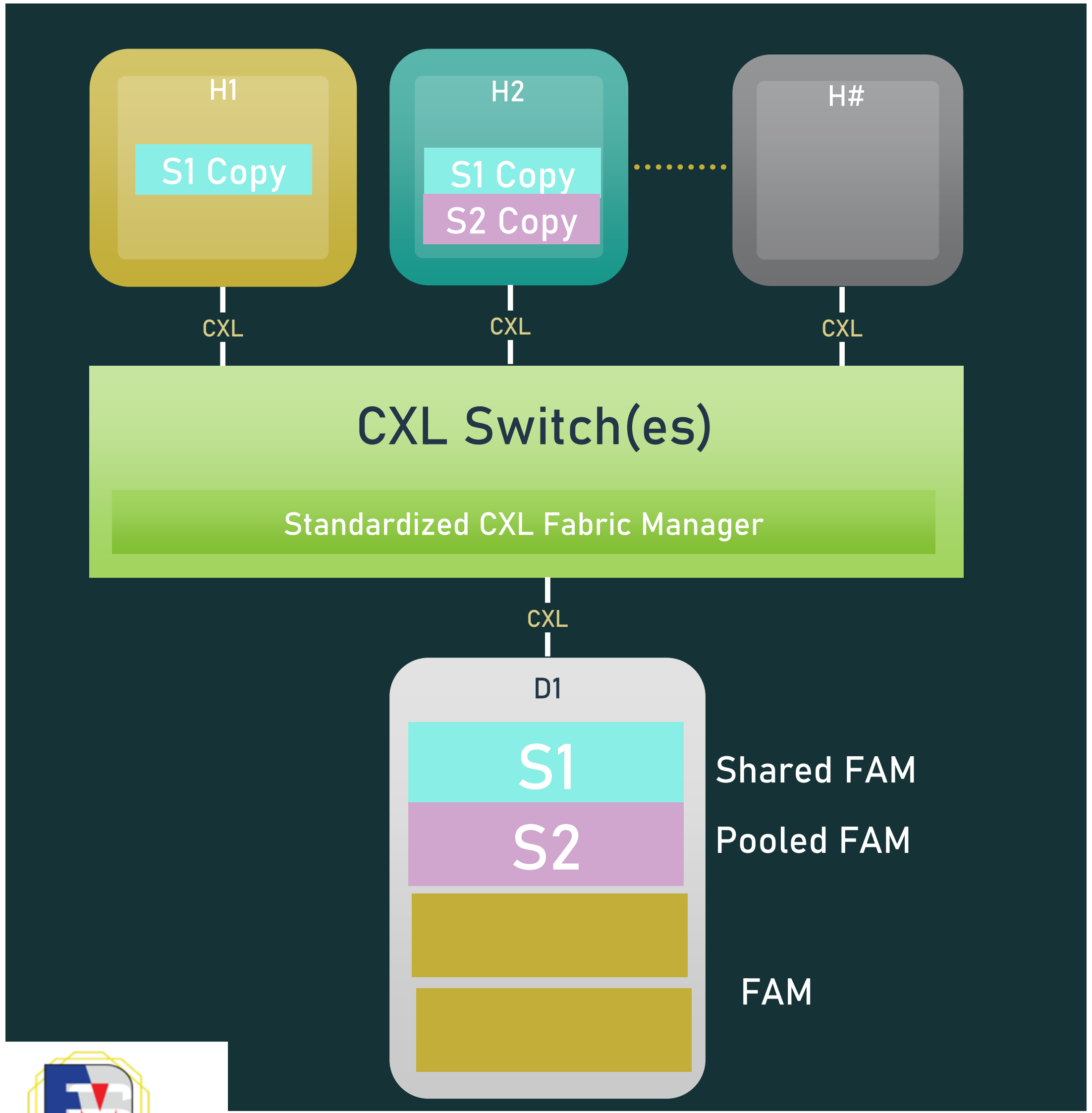
Pooling in CXL 2.0

- Pooling in 2.0 – Multi-Logical Device (MLD)
- Up to 16 hosts
- Requires MLD-capable switch

Pooling in CXL 2.0 – MLD



CXL 3.0: Pooled FAM, Shared FAM, Global-FAM



Host-managed Device Memory (HDM) that is exposed from a device that supports multiple hosts is referred to as Fabric Attached Memory (FAM)

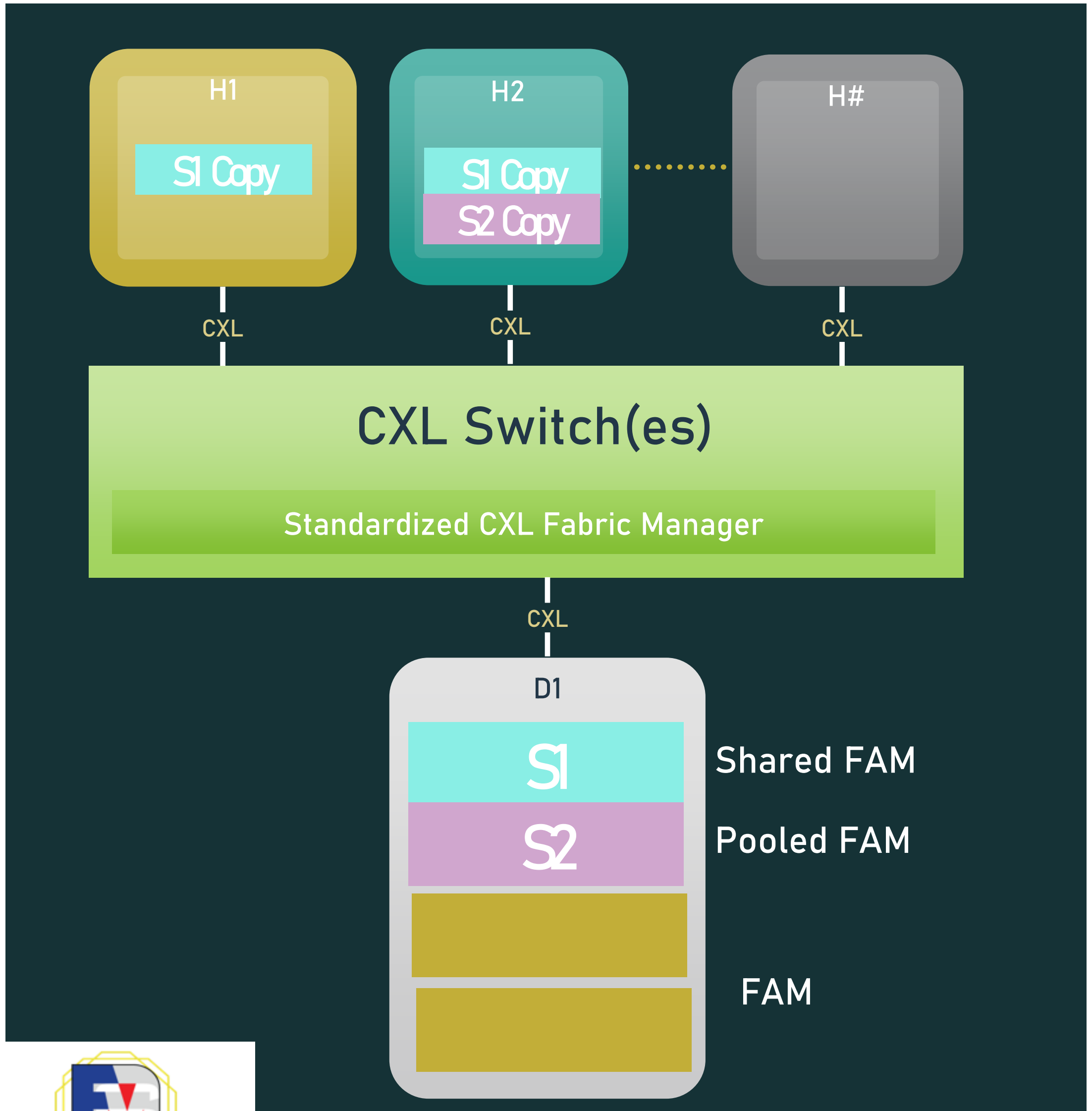
FAM exposed via Logical Devices (LD) is known as LD-FAM

FAM exposed in a more scalable manner using Port-Based Routing (PBR) links is known as Global-FAM (G-FAM)

FAM where each HDM region is dedicated to a single host is known as Pooled Memory or Pooled FAM.

FAM where multiple hosts are configured to access a single HDM region concurrently is known as Shared FAM.

CXL 3.0: Coherency Models for Shared FAM



Coherency Model for Shared FAM is designated by the FM as either Software-Managed Coherency or Multi-Host Hardware Coherency

Software-Managed Coherency does not require the Device hardware to track Host Coherency.

- Mechanisms for software co-ordinating cache line ownership is outside the scope of CXL specification

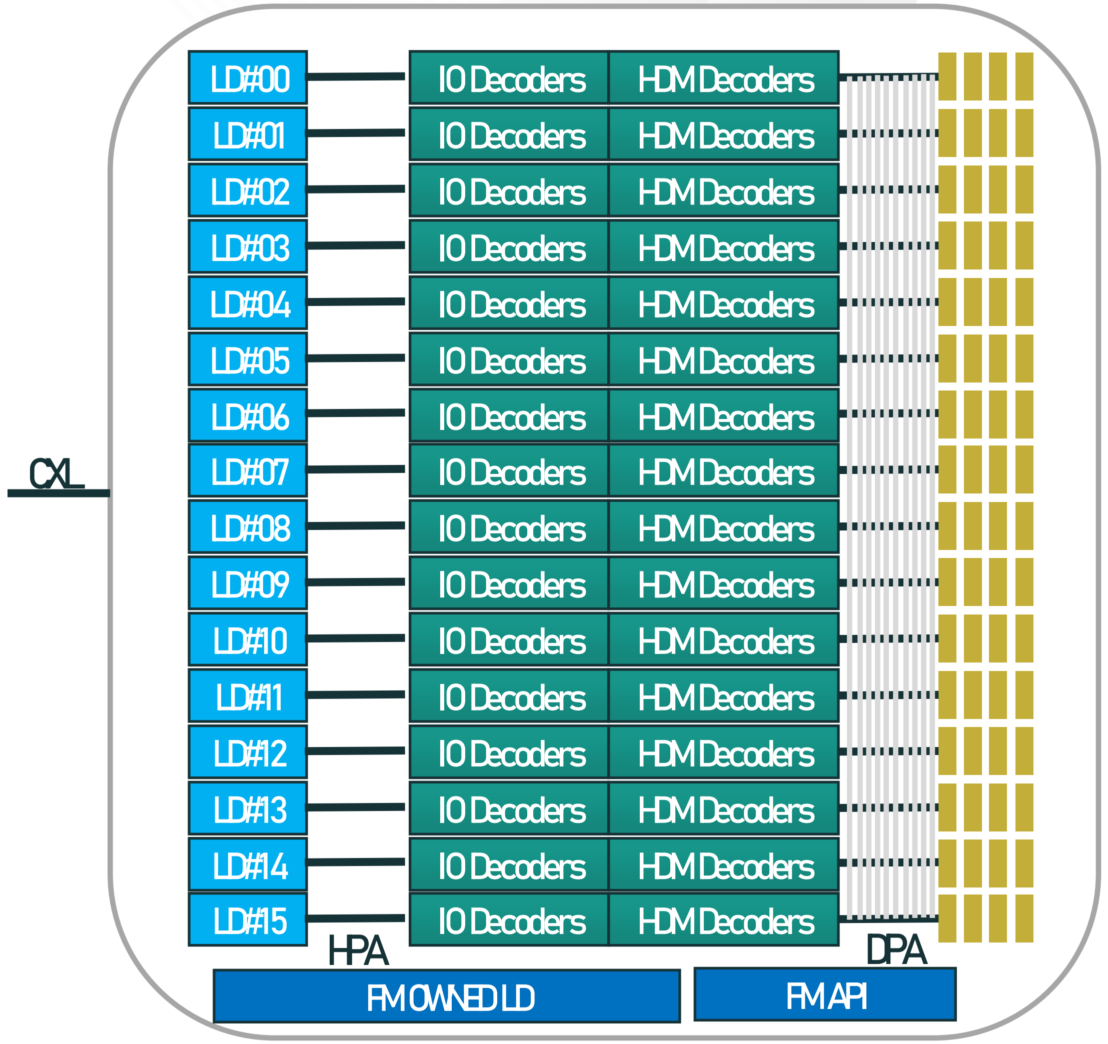
Multi-Host Hardware Coherency requires the Device hardware to track Host Coherency.

- Unique hardware flows are required for coherency
- Shared FAM region must be mapped as HDM-DB
- Support BI channel
- Direct Peer-Peer requires support for Unordered IO

CXL Multi-Logical Memory Device

FAM

- A FAM Type 3 device can partition its resources into Logical Devices (LD)
  - Up to 16 LDs (Type 3 only) AND
  - One Fabric Manager (FM) owned LD
- Each LD
  - Appears as Type 3 SLD device
  - Identified by LD-ID
  - FM binds each LD to a Virtual Hierarchy
- FM Owned LD
  - Accessible by FM only by using LD-ID of 0xFFFFh
  - Manage Link and Device
  - Memory resources are not assigned to LD owned LD
  - Error messages generated by LD are routed to FM
  - Does not participate in GPF Flows
- MLD Link
  - MLD Link Discovery & Link Operation configured via Alternate Protocol Negotiation

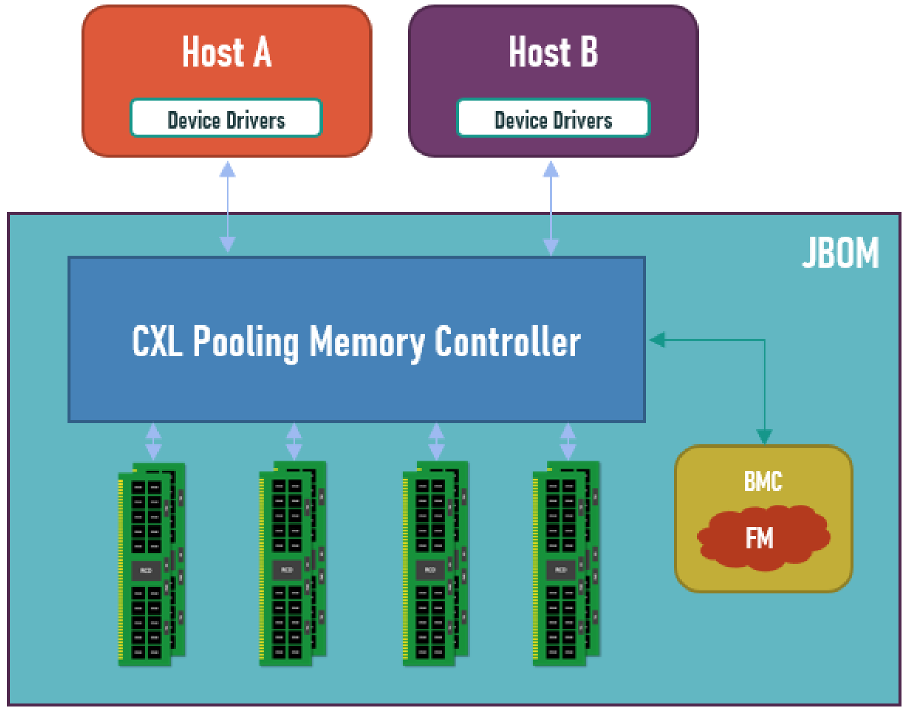


HPA – Host Physical Address DPA – Device Physical Address

Introducing Multi Headed Devices (MHDs)

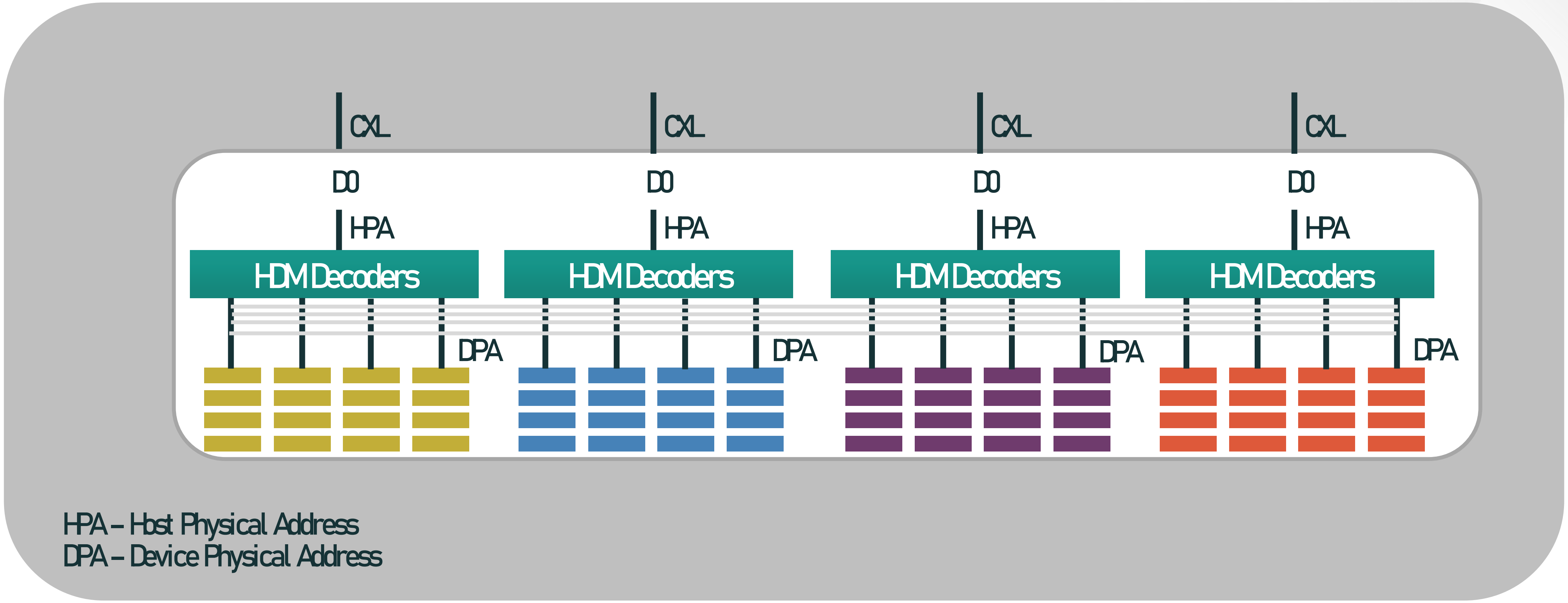
- New FAM option in 3.0
- Type 3 device with multiple CXL "heads"
- Two types of MHD:
  - Multi-Headed-Single Logical Device (MH-SLD)
  - Multi-Headed-Multiple Logical Device (MH-MLD)
- Support Pooled FAM or Shared FAM

New in CXL 3.0 – MHD



CXL Multi-Headed Memory Device

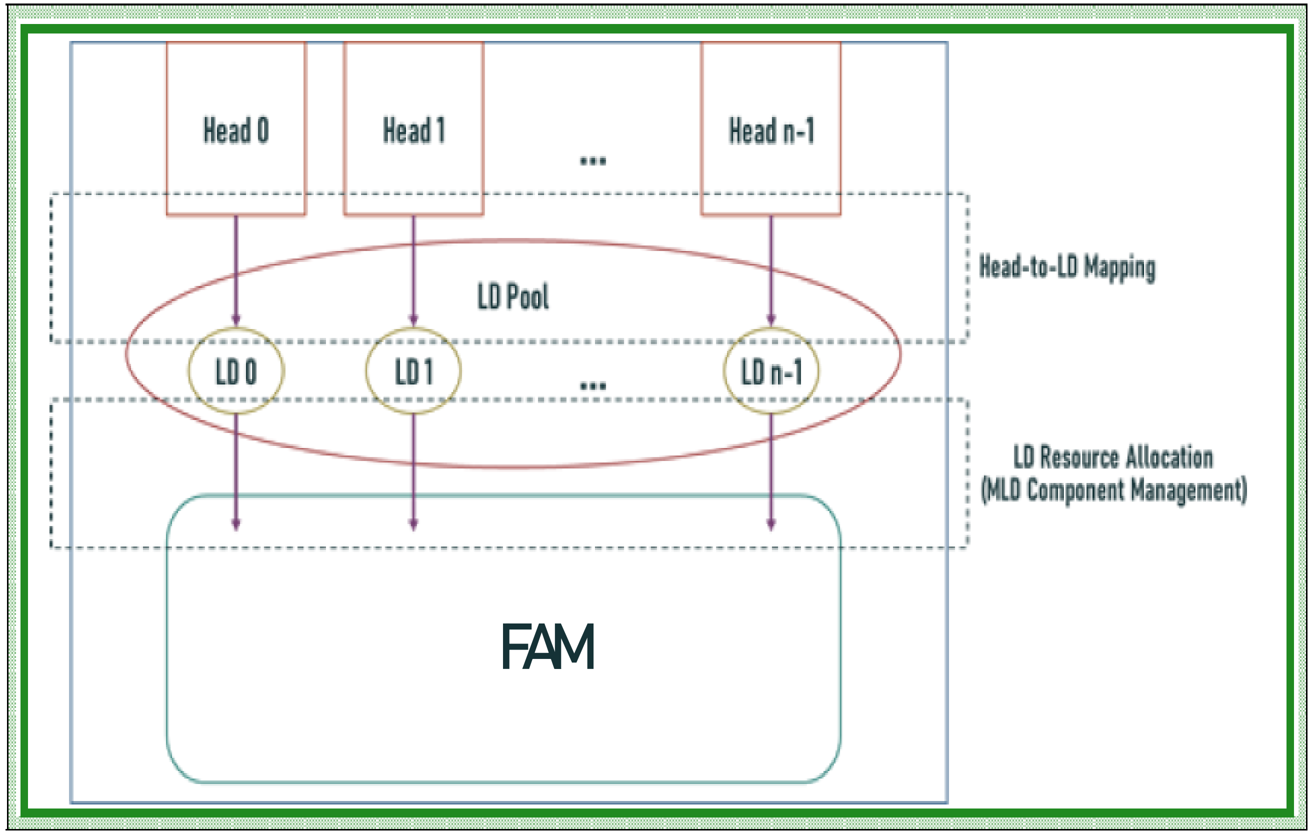
FAM Device



- Represents an SLD behind each CXL Port
- The LD Pool CCI is used for management of all LDs within the device
- Example – 4 Ported CXL Memory Device with equal allocation of pooled resources across ports

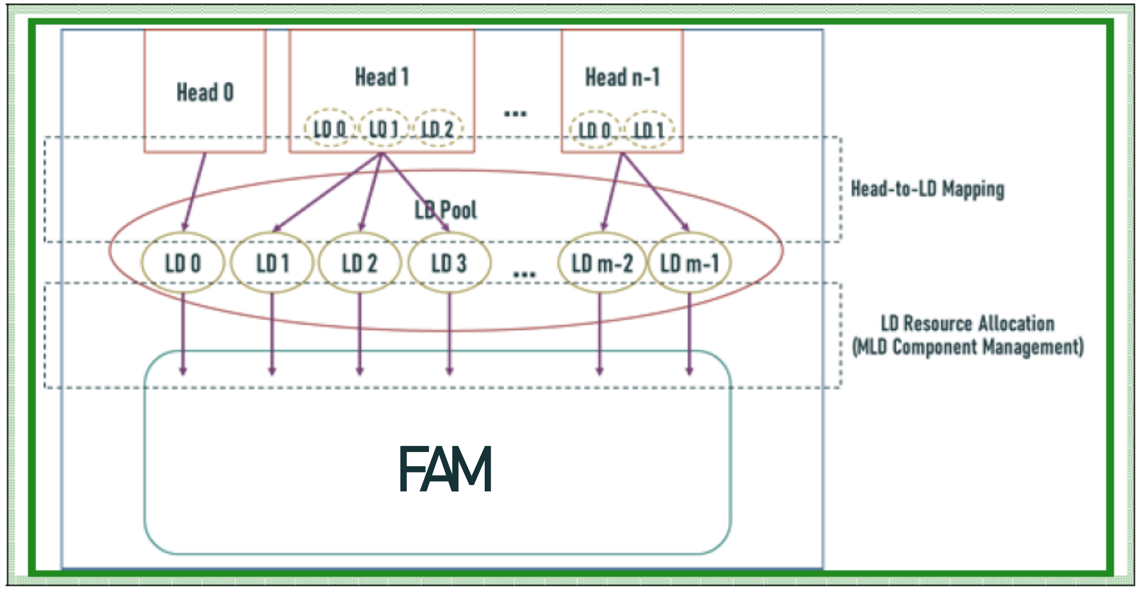
MH-SLD & MH-MLD

- MH-SLD presents an SLD to each head
  - 1:1 mapping of LDs to heads
  - Heads can direct attach to different hosts



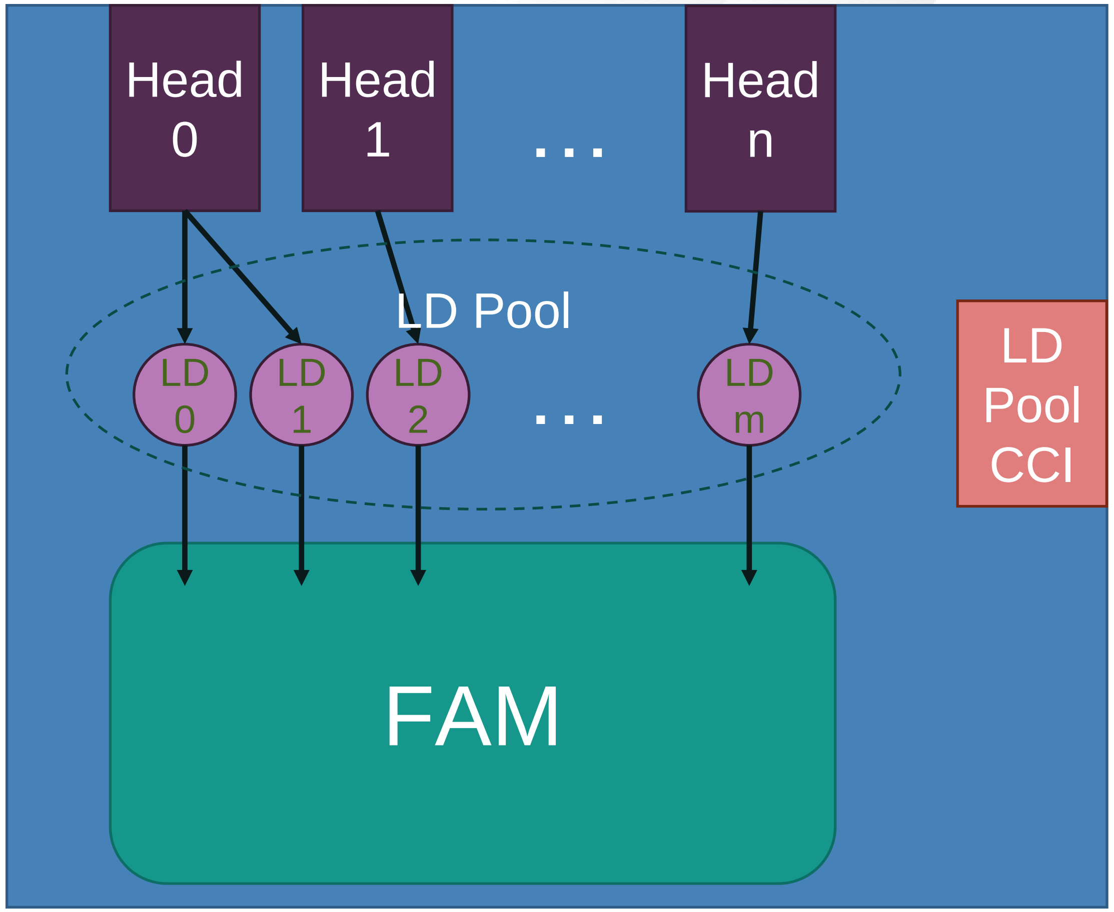
MH-SLD & MH-MLD

- MH-MLD presents an MLD to each head
  - Up to 16 LDs mapped to a single head
  - Requires a switch for MLD functionality



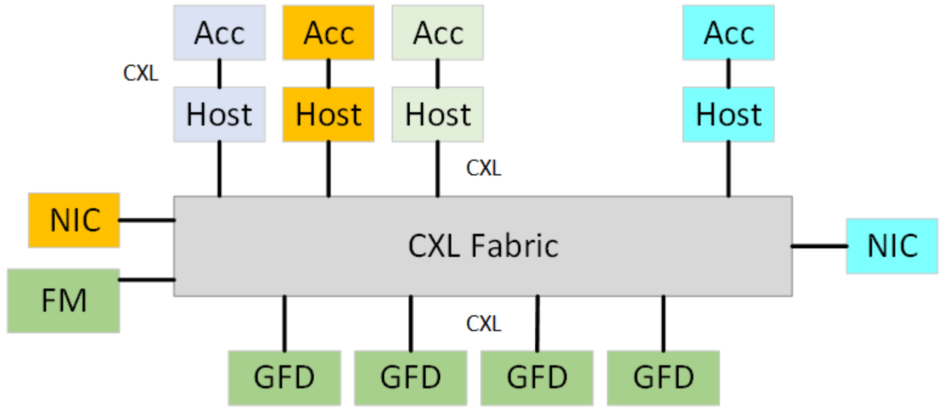
MHD Management

- MHD is a device with multiple LDs
- LD Pool managed from “LD Pool CCI”
- Reuse MLD management framework from 2.0

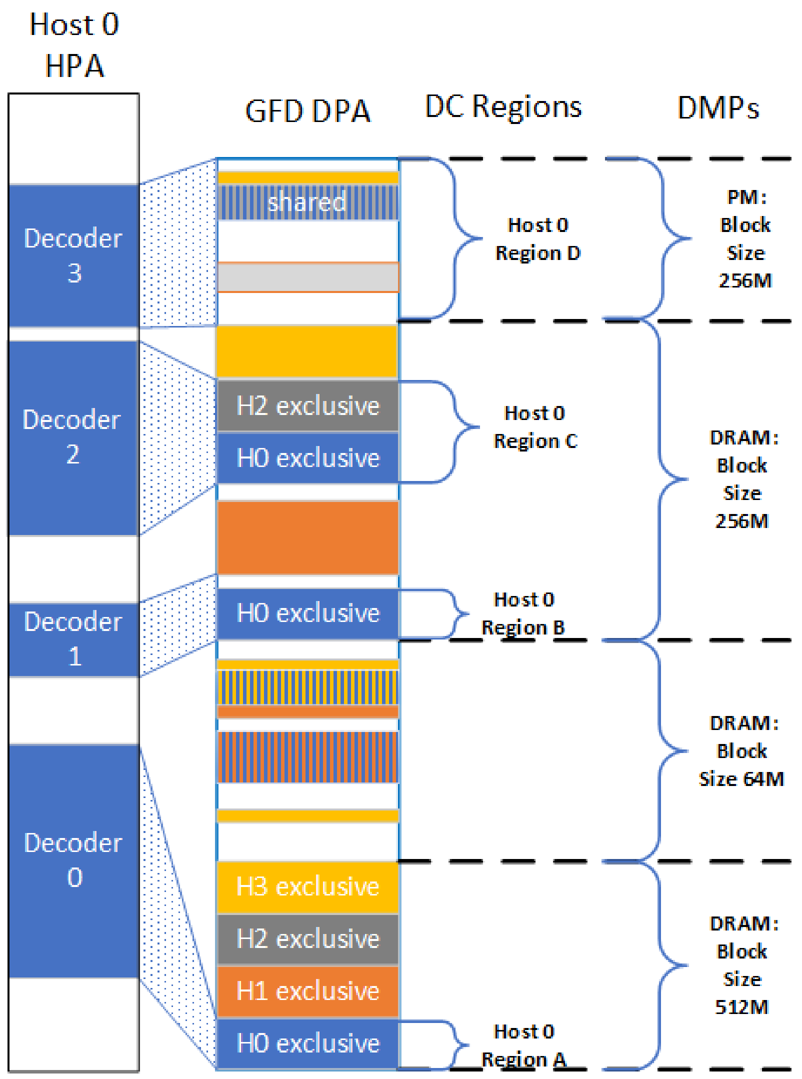


Global FAM (G-FAM) Type 3 Device

- G-FAM – highly scalable memory resource that is accessible by all host and peer devices within CXL Fabric
- Single Host Mapped or Multiple Host Shared
  - Software or Hardware Managed Coherency, when shared.
- Multiple Host Access – CXL.mem
- Peer Access – CXL.IO (UIO)
- GFD configuration space accessed only by FM
- Memory capacity managed using Dynamic Capacity Mechanisms
- Host Client Management of GFDs to be covered in future ECNs.



Example HPA Mapping to DMPs

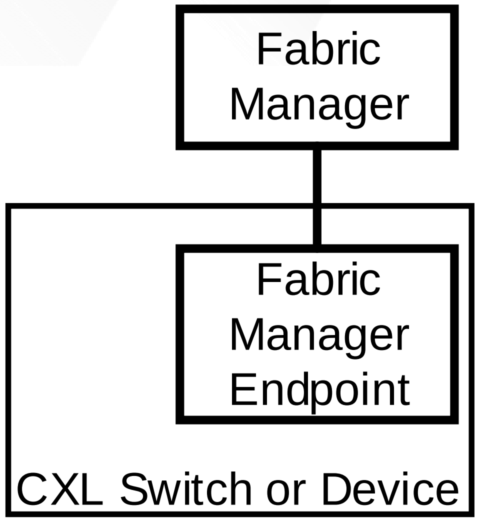


Differences between LD-FAM and G-FAM

Feature or Attribute	LD-FAM	G-FAM
Number of supported hosts	16 max	1000s architecturally; 100s more realistic
Support for DMPs	No	Yes
Architected FM API support for DMP configuration by the FM	N/A	Planned
Routing and decoders used for HDM addresses	Interleave RP routing by host HDM Decoder Interleave VH routing in USP HDM Decoder 1–10 HDM Decoders in each LD	Interleave RP routing by host HDM Decoder Interleave fabric routing in USP Fabric Address Segment Table (FAST) and Interleave DPID Table (IDT) 1–8 GFD Decoders per SPID in the GFD
Interleave Ways (IW)	1/2/4/8/16 plus 3/6/12	2–256 in powers of 2
DC Block Size	Powers of 2, as indicated by Region * Supported Block Size Mask	64 MB and up in powers of 2

Fabric Manager Endpoint and API

- Fabric Manager is a control entity that manages the CXL 3.0 Switch and the Memory Controller
  - FM can be an external BMC, a Host, or Firmware internal to the Switch
- FM Endpoint is a required feature for any switch that supports MLD ports or that supports dynamic SLD port binding
- FM API is the standardized interface for the FM to communicate with devices
- FM API uses an MCTP interface between Fabric Manager and devices
  - MCTP physical interface is switch vendor specific but could be PCIe®, CXL.io VDM, SMBus, Ethernet, UART, USB, internal, …
- In general there are no real-time response requirements for the Fabric Manager so it needn't be performant

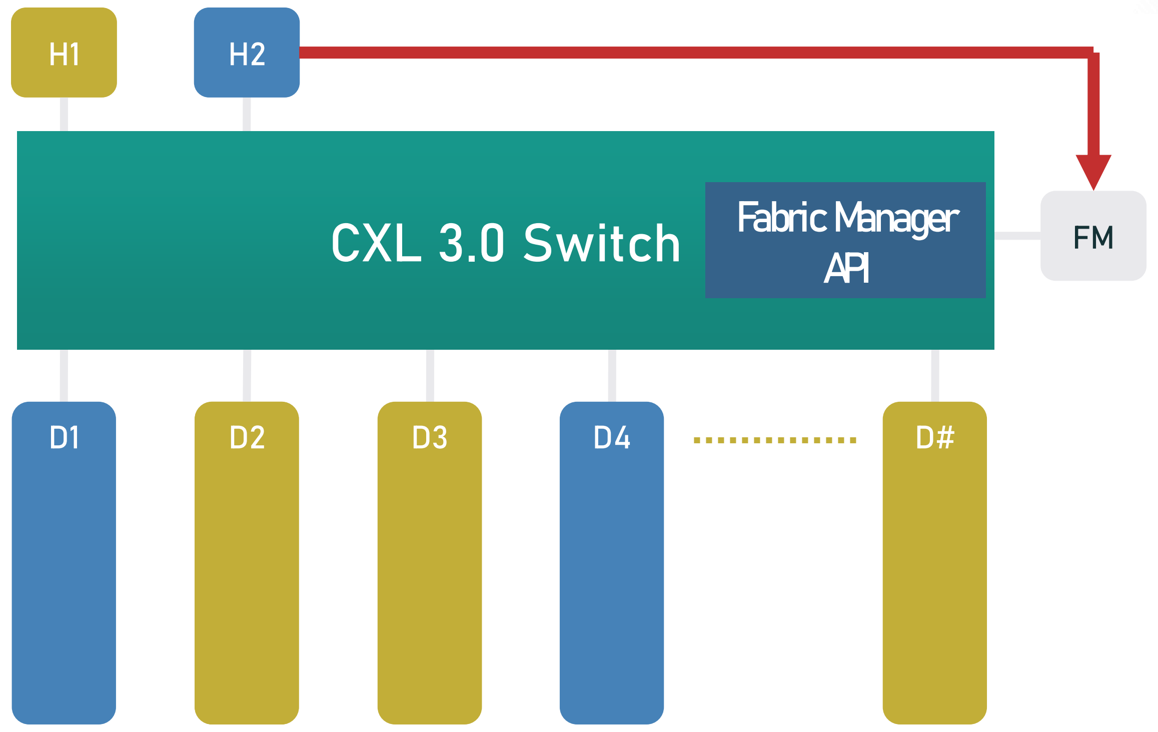


Fabric Manager

- Fabric Manager plays a critical role in CXL for systems supporting Memory Pooling
- The Fabric Manager enables dynamic system changes supporting Memory disaggregation
- Some examples:
    - Managing all devices that support traffic from multiple Hosts including:
        - Downstream ports connected to MLD ports
        - FM-owned Logical Device within an MLD component
        - Unbinding and rebinding of Logical Devices within an MLD between Hosts
    - Unbinding and rebinding of an SLD
    - Re-allocation of memory within an MLD
    - Re-allocation of memory within a MH-SLD or MH-MLD

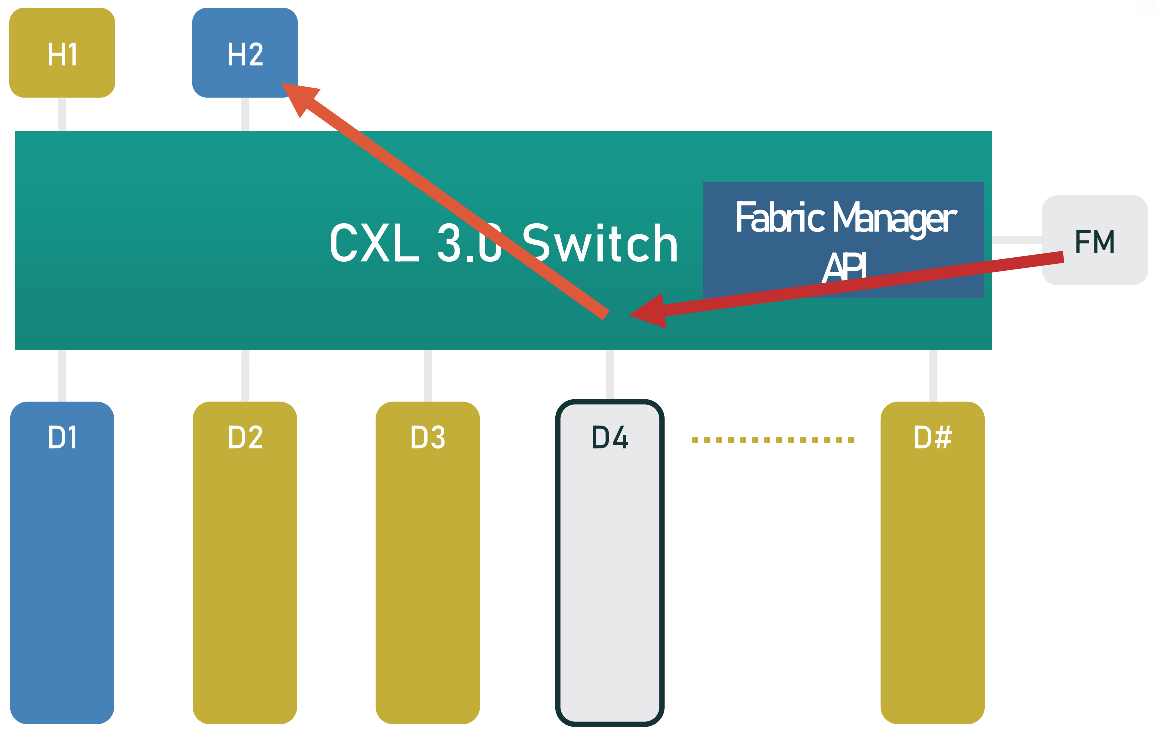
Memory Pooling with Single Logical Devices

H2 notifies FM that D4 memory is no longer needed



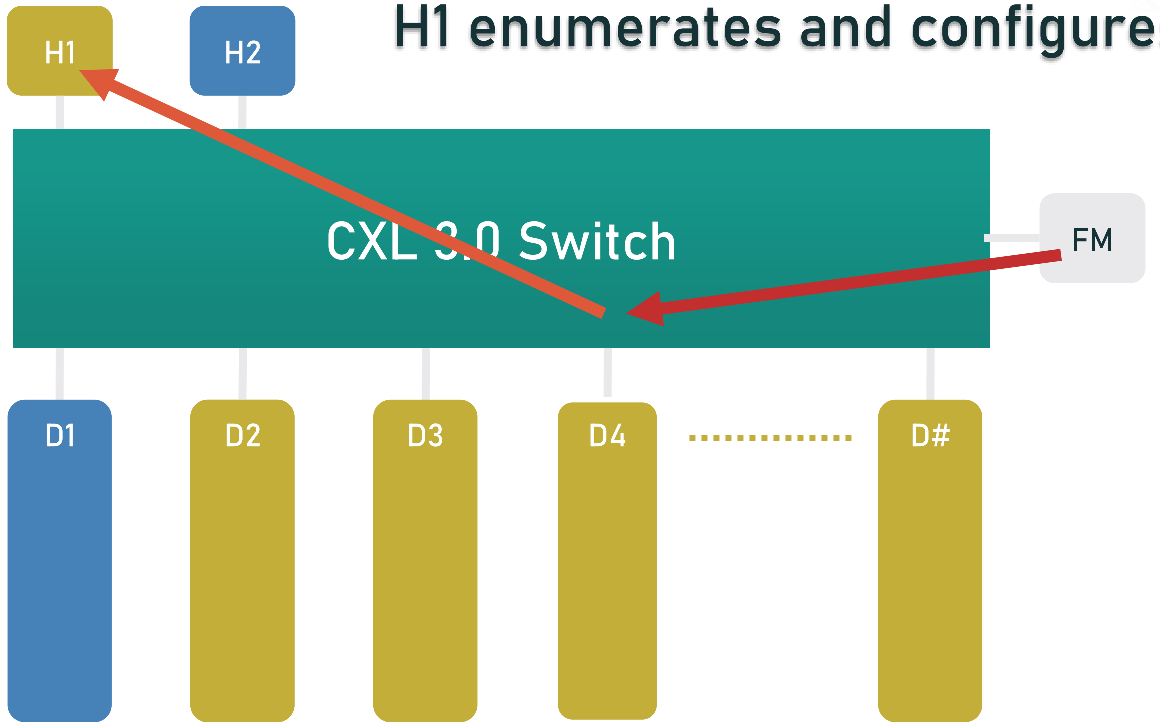
Memory Pooling with Single Logical Devices

FM tells switch to UNBIND D4
Switch notifies H2 of the managed hot remove

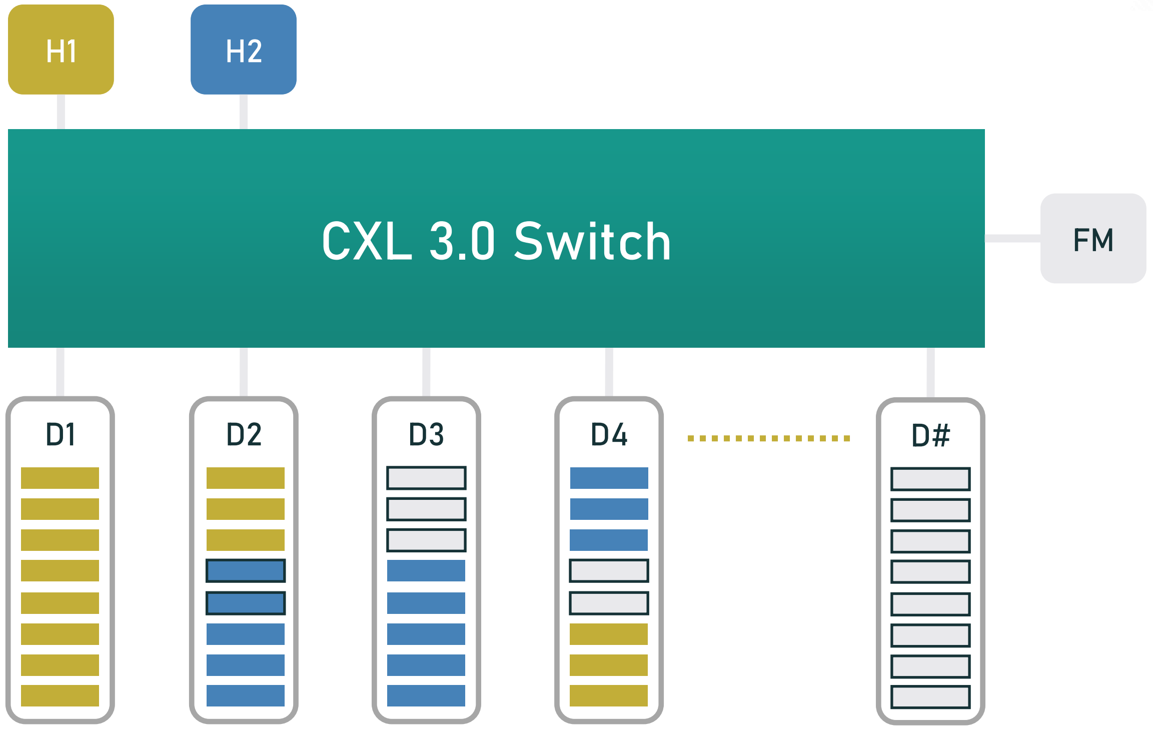


Memory Pooling with Single Logical Devices

FM tells switch to BIND D4 to H1
Switch notifies H1 using managed hot add
H1 enumerates and configures accesses to D4

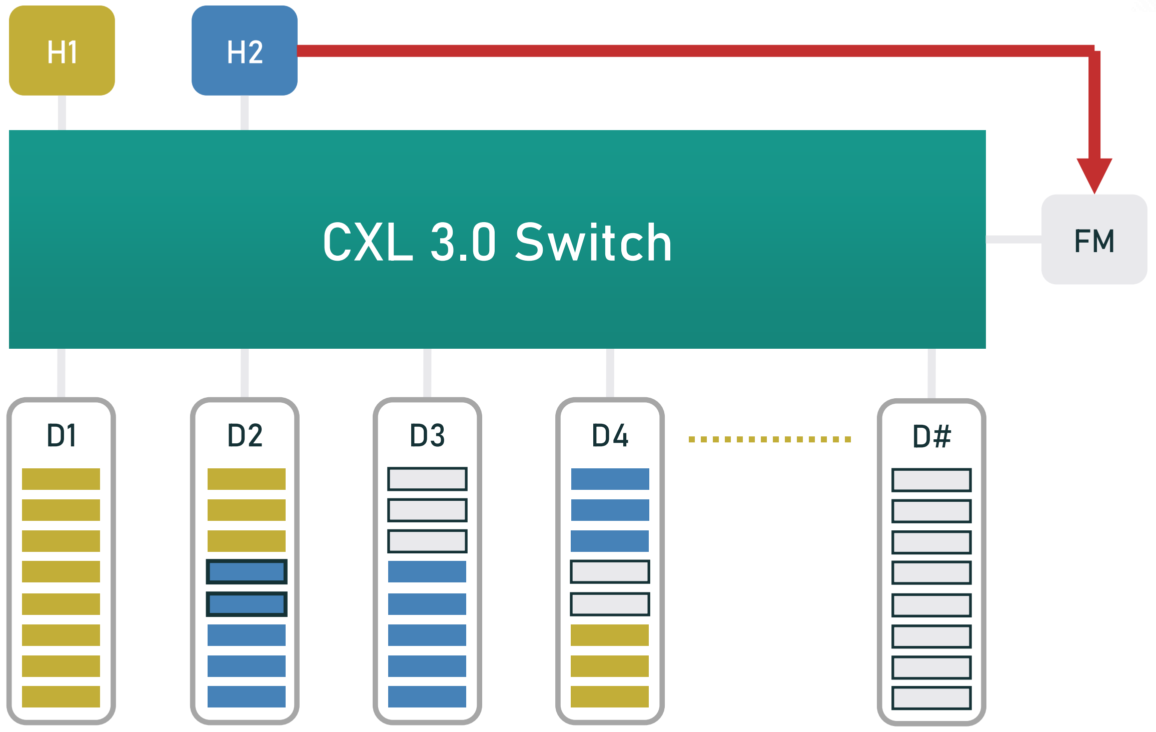


Memory Pooling with Multi-Logical Devices



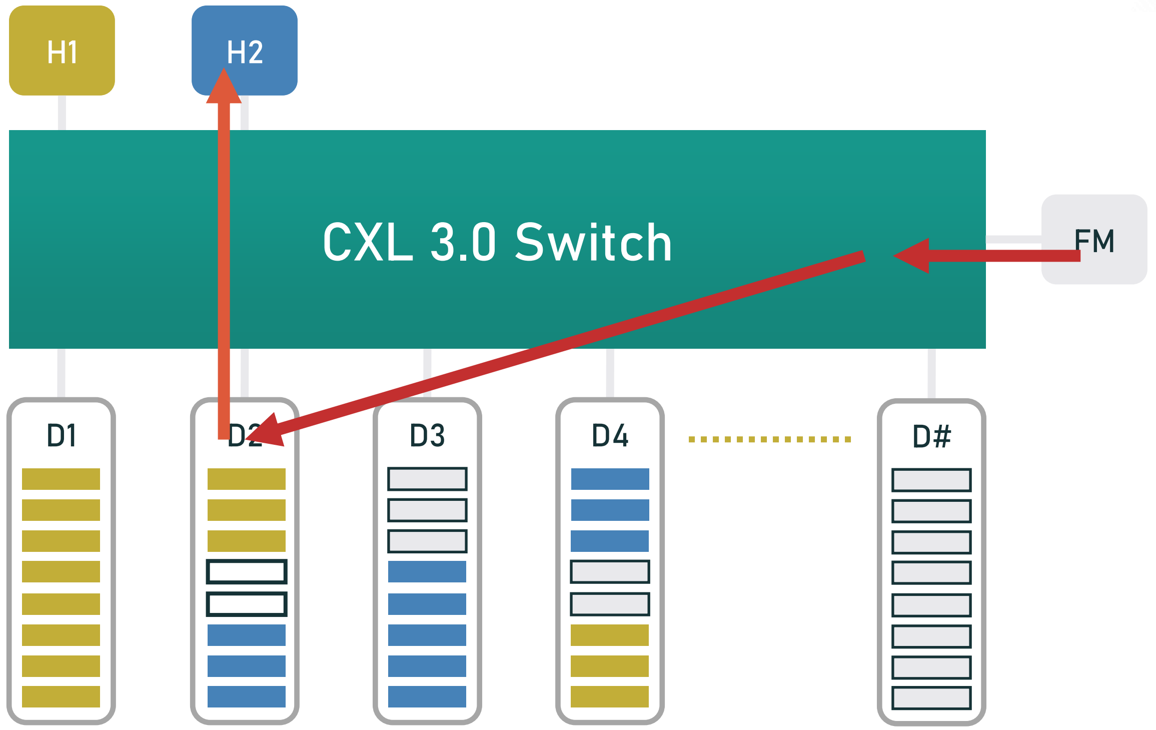
Memory Pooling with Multi-Logical Devices

H2 notifies FM that some D2 memory is no longer needed



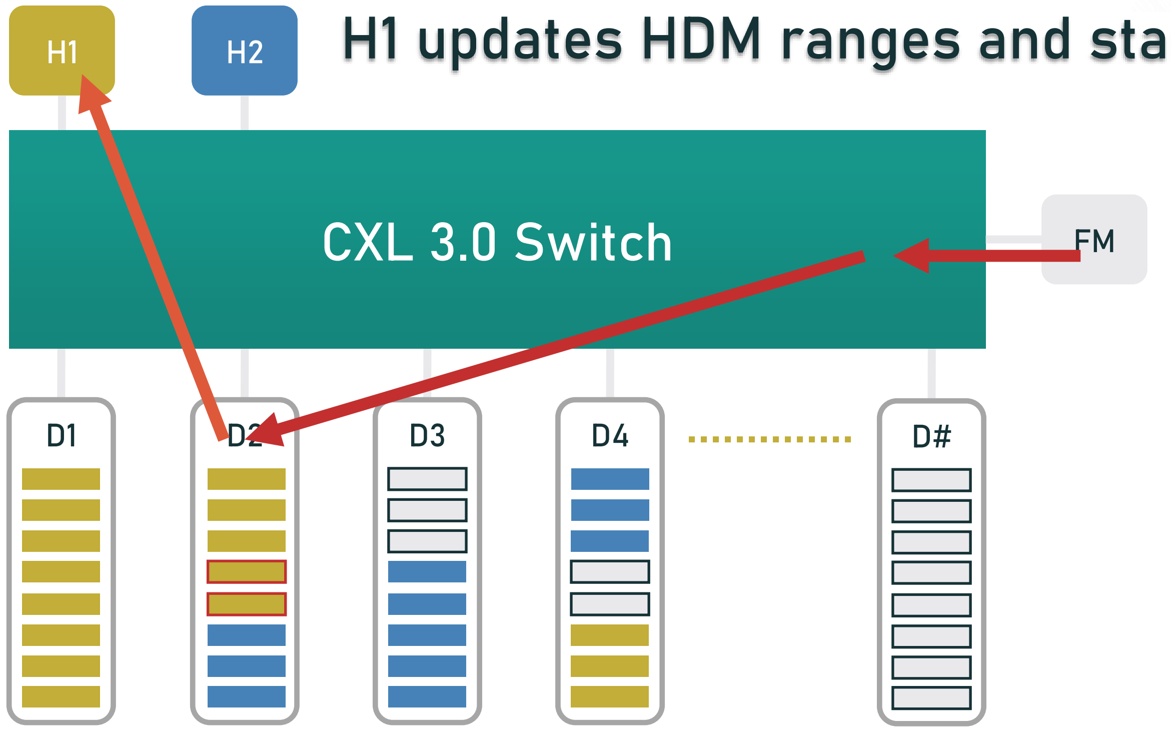
Memory Pooling with Multi-Logical Devices

FM tells D2 to de-allocate some blue memory
D2 notifies H2



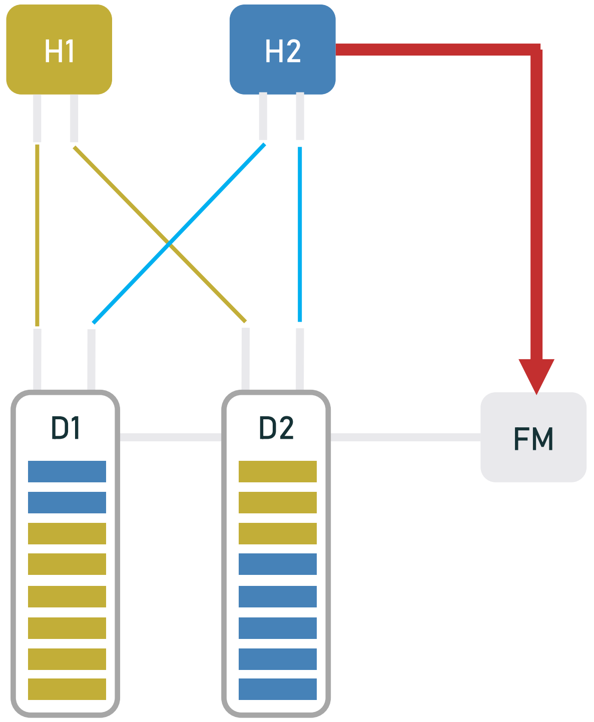
Memory Pooling with Multi-Logical Devices

FM tells D2 to allocate some yellow memory
D2 notifies H1
H1 updates HDM ranges and starts using memory



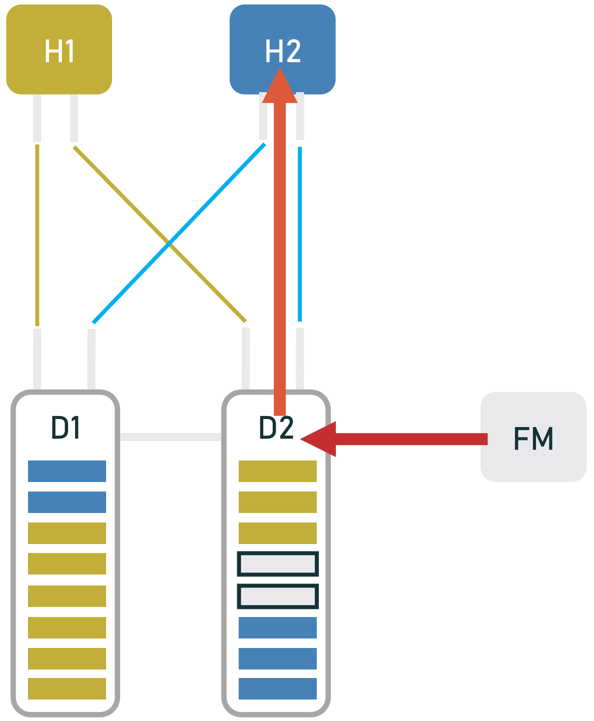
Memory Pooling without a Switch

H2 notifies FM that some D2 memory is no longer needed



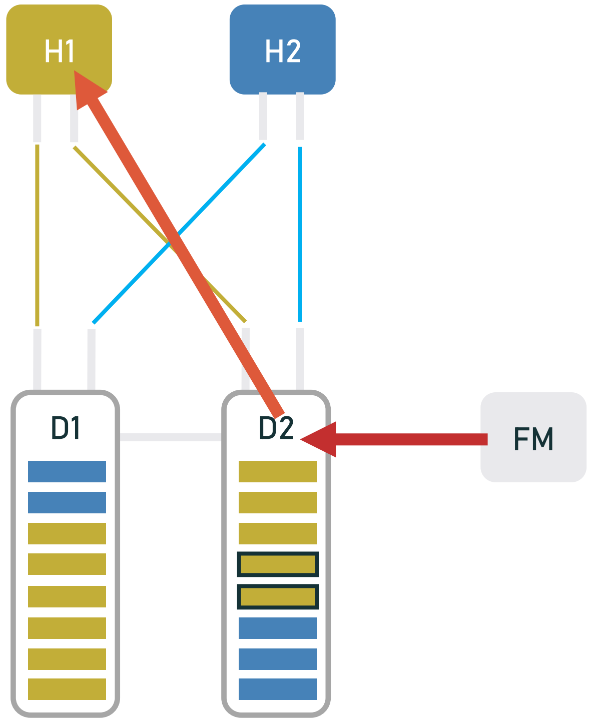
Memory Pooling without a Switch

FM tells D2 to de-allocate some blue memory
D2 notifies H2



Memory Pooling without a Switch

FM tells D2 to allocate some yellow memory
D2 notifies H1, H1 updates HDM ranges



FAM with CXL

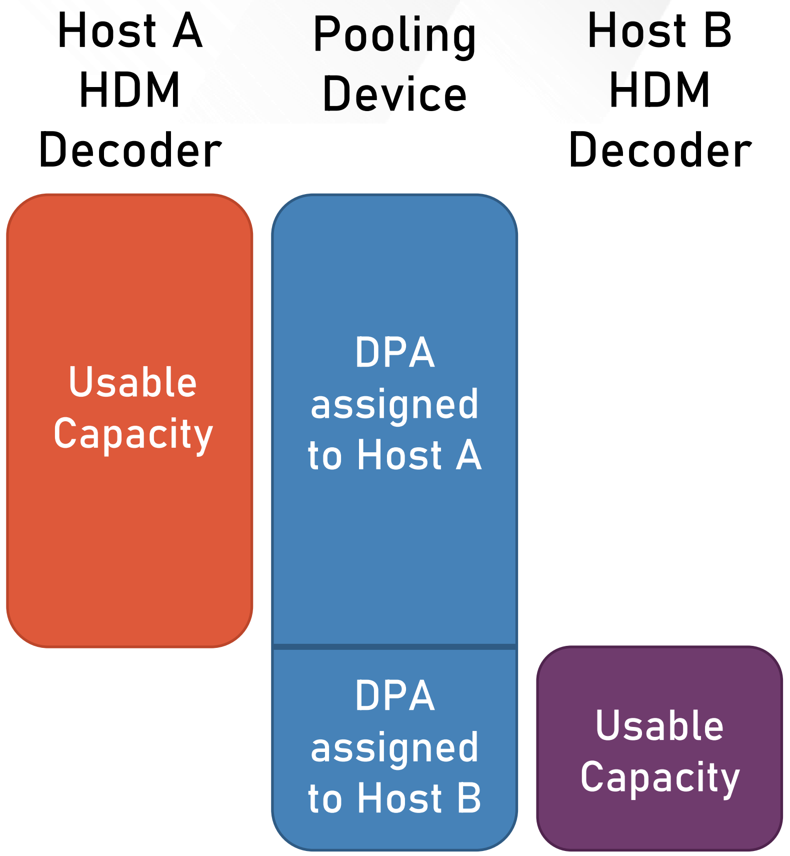
- Other FM API features beyond BIND, UNBIND, and SET LD ALLOCATIONS:
  - Switch Discovery including capacity, capabilities, and connected devices
  - Event notification such as switch link events and Advanced Error Reporting for FM owned resources
  - Manage MLD QoS parameters

- Benefits of FAM
  - Effective utilization of memory resources within a system
  - Dynamic Allocation/deallocation of memory resources
  - Total Cost Of Ownership (TCO) savings

Dynamic Capacity Devices (DCDs)

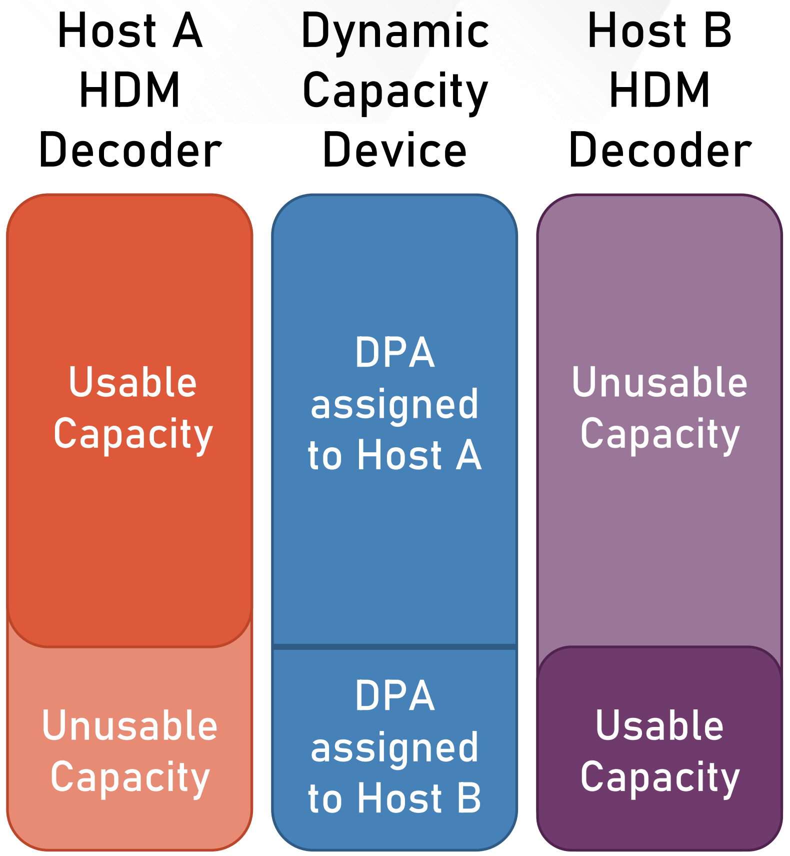
Problem Statement

- Before DCD, changing memory allocation was very disruptive
- Required HDM decoder reprogramming
  - Traffic quiesced
  - System reset likely



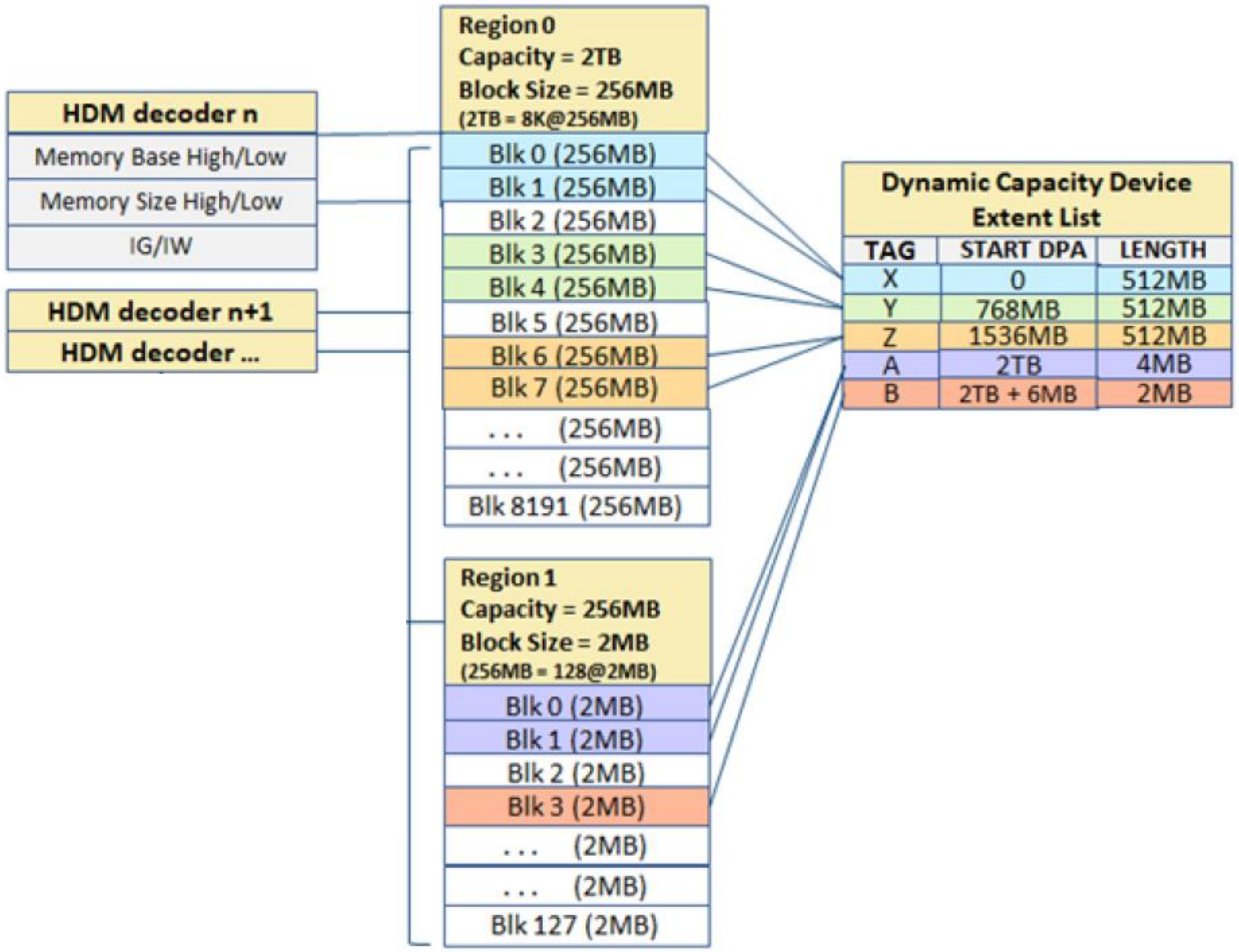
Dynamic Capacity Devices

- Allows memory capacity changes without HDM decoder reprogramming
- DCD presents maximum capacity to each host
  - HDM decoders programmed for full DPA range
  - DCD command set used to discover *actual* memory allocation



Dynamic Capacity Devices

- Capacity defined per host in a block-based "Extent List" – [DPA start, length, tag]
- Up to 8 DC regions with different properties (coherency, sharing, block size)
- Fabric Manager configures DCD allocations



Dynamic Capacity Device Extent List

TAG	START DPA	LENGTH
X	0	512MB
Y	768MB	512MB
Z	1536MB	512MB
A	2TB	4MB
B	2TB + 6MB	2MB

Software and Error Mgmt

CXL Hierarchy as Seen by Software

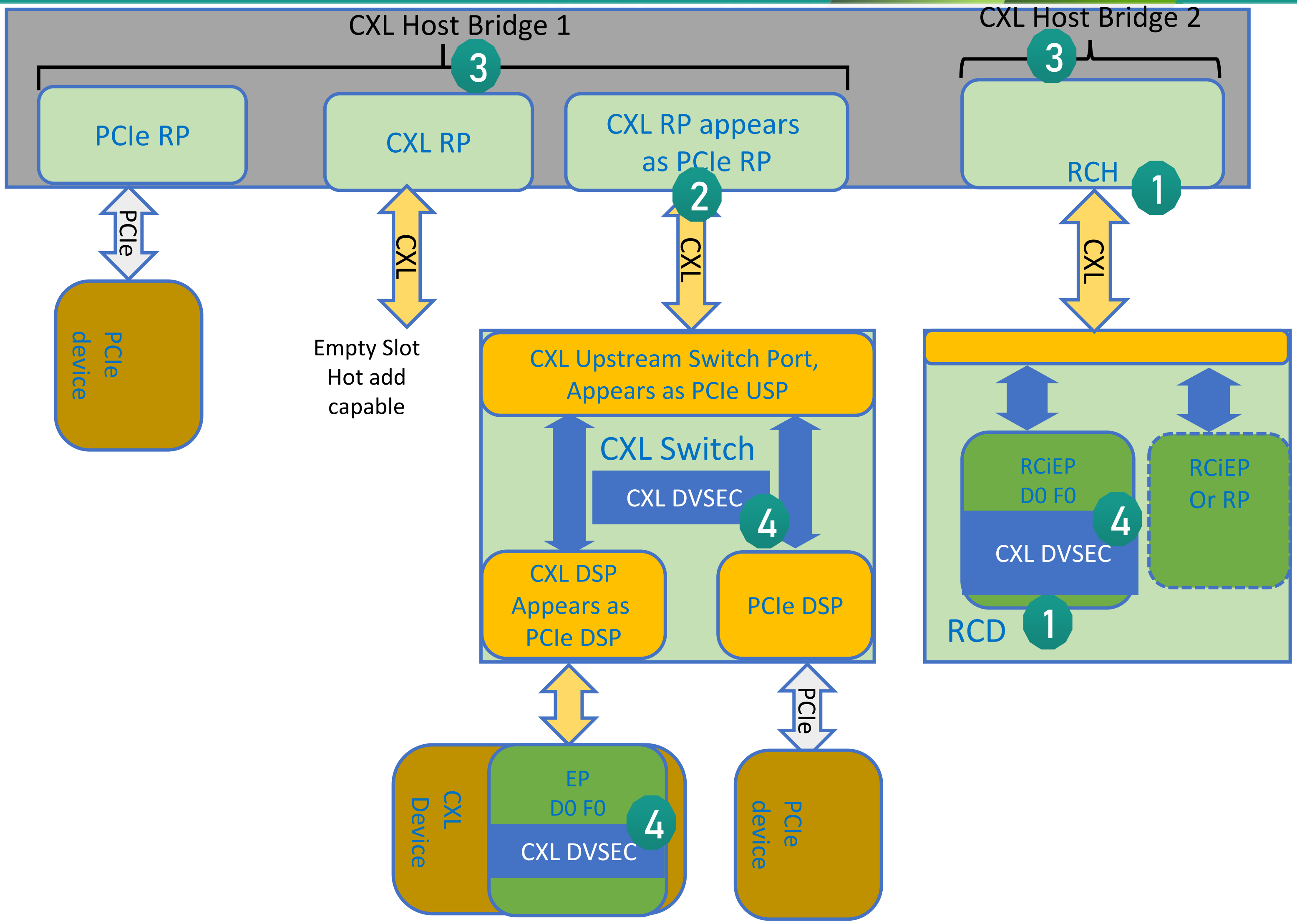
1 RCD/RCH – Restricted CXL Device/Host, follow CXL 1.1 spec

2 CXL Virtual Hierarchy (VH) – Made up of CXL Root Ports, CXL Switches and CXL Devices that follow CXL 2.0 and later specs

3 CXL Host Bridges can be discovered via CXL Early Discovery Table (ACPI CEDT).

- Next level of discovery is based on ACPI Namespace
  - CXL Host Bridge Hardware ID="ACPI0016"
  - Compatibility ID of PCIe® Host Bridge to enable enumeration by non-CXL enabled OSs
- The last leg of discovery uses standard PCIe Enumeration

  4 CXL components are decorated with appropriate CXL DVSEC structures



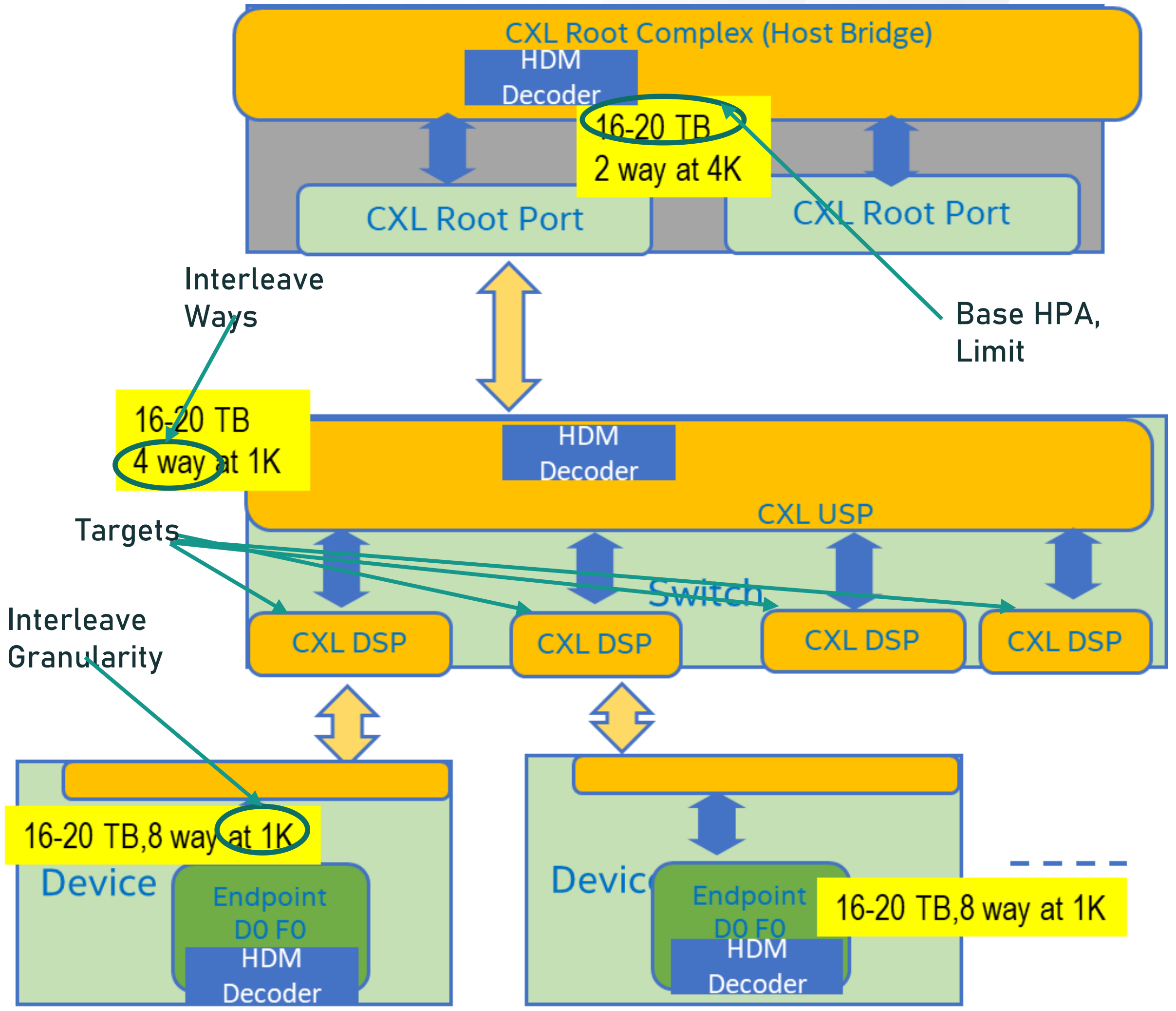
Flash Memory Summit

Discovering Hetero Memory Attributes

- CXL systems are heterogenous by nature
- SRAT and HMAT models work great when system firmware has apriori knowledge of coherent components and the config is relatively static.
- CXL breaks both assumptions
  - Open ecosystem, hot-plug, Dynamic Capacity Devices ..
- However, OS/VMMs still need SRAT and HMAT equivalent information to allocate memory optimally ..
- Coherent Device Attribute Table is the answer
  - Each coherent device reports its local latency/BW characteristics via DOE interface. This data structure is called CDAT. Expressed in terms of Device Physical Address (DPA).
  - System Firmware or OS/VMM stiches together CDAT obtained from each component to construct SRAT/HMAT equivalent structures.

Memory Interleaving

- CXL memory devices may be interleaved for performance reasons
- An Interleave Set is identified by
  - Base HPA, Limit - Multiples of 256 MB
  - Interleave Way – 2, 4 or 8
  - Interleave Granularity - 2** (8,9,10, 11, 12, 13,14)
  - Targets (applicable to RC and USP)
- Configured via HDM Decoder registers in USP, RC and Device
  - USP and RC use these decoders to pick the target
  - Device uses these decoders to translate Host Physical Address (HPA) to Device Physical Address (DPA)

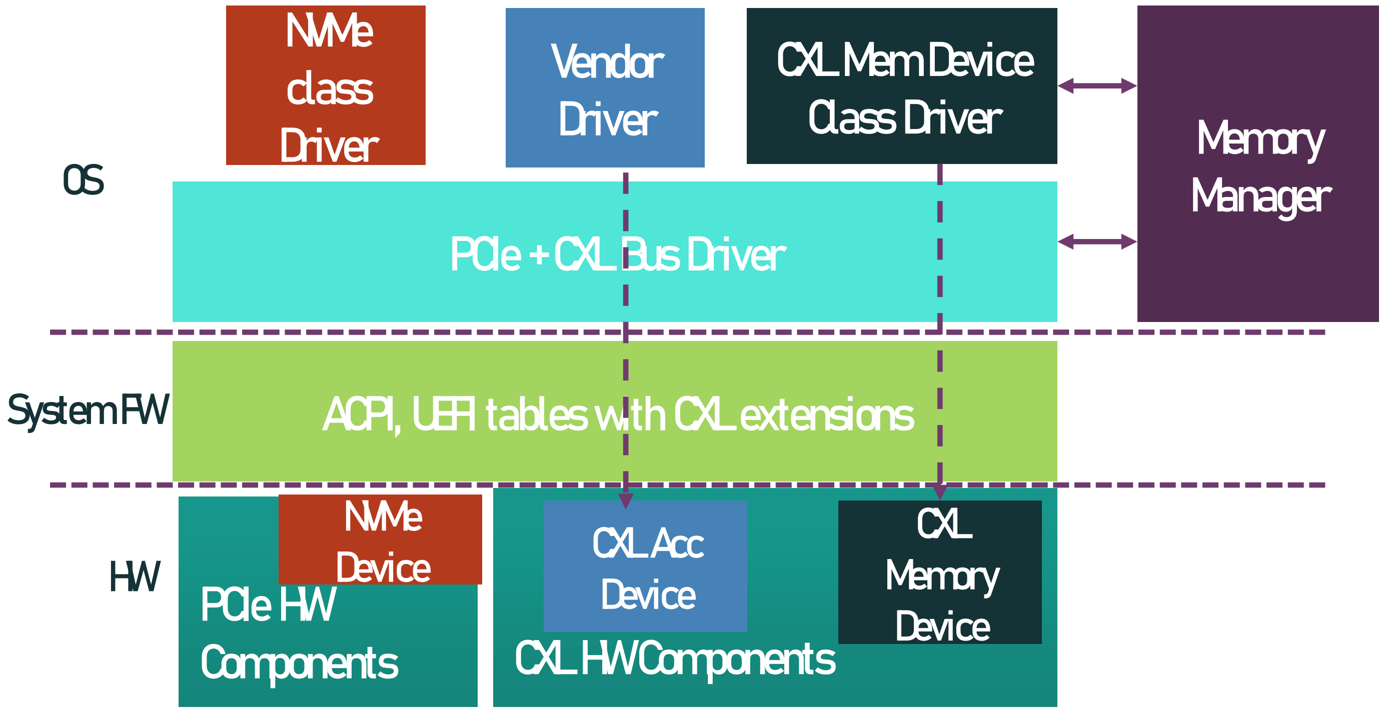


CXL Memory Device Config Interface

- Type 3 devices, especially persistent memory devices, rely on system software for provisioning and management
- CXL specification defines a standard register interface for managing CXL attached memory devices
- Allows OS to carry a class driver for these devices
- Architecture Elements
  - Defined as number of discoverable Capabilities
  - Capabilities includes Device Status and standard mailboxes, accessed via MMIO registers
  - Standardized mailbox commands
  - Allow Vendor specific extensions

Example of Software Stack

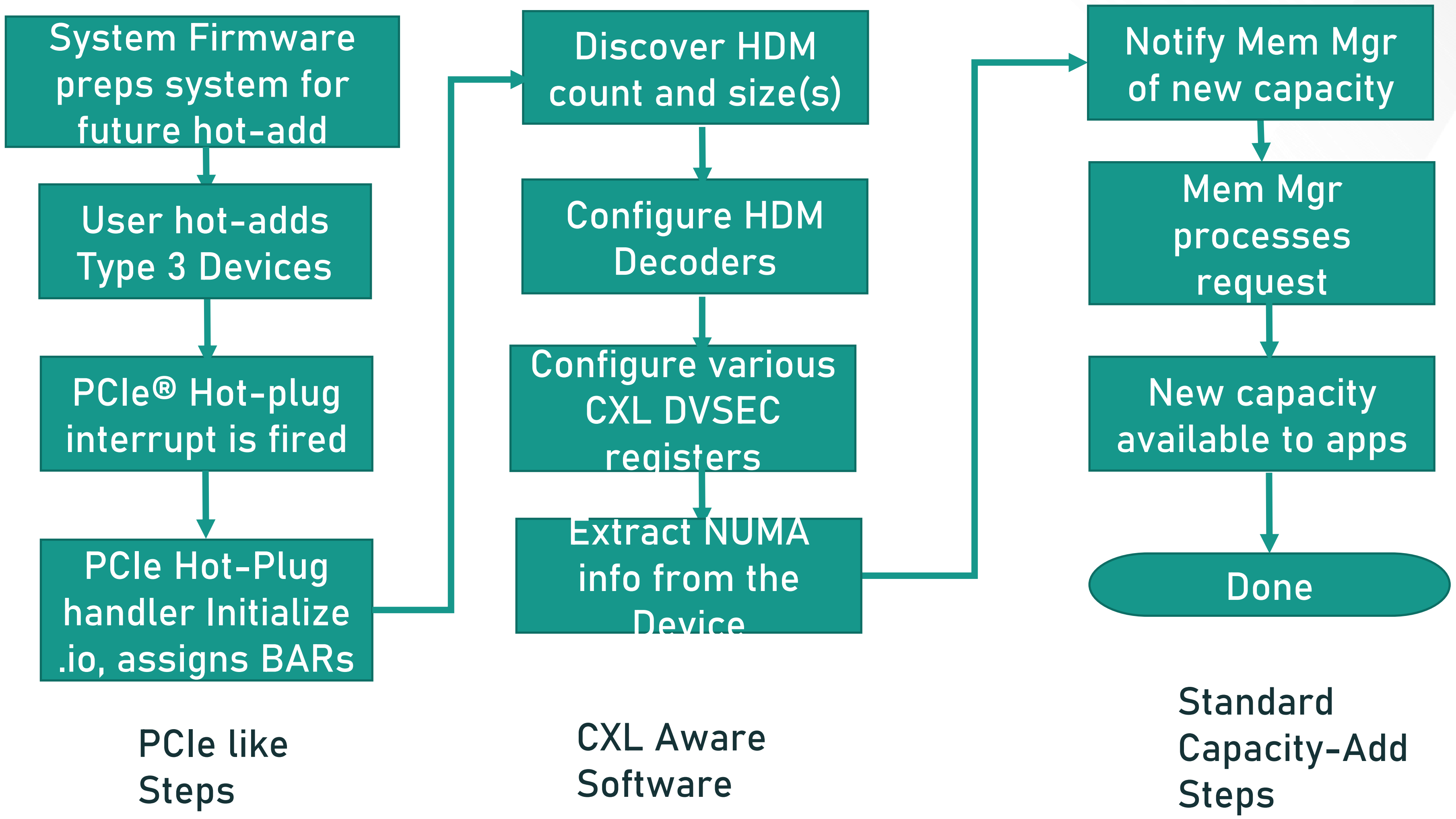
- CXL extends PCIe® and leverages ideas from PCIe world
- CXL aware System FW exposes host/system specific aspects of CXL via UEFI and ACPI tables
- PCIe bus driver may be enhanced to handle CXL enumeration
- CXL memory devices are Type 3 devices with standard reg i/f, managed by a class driver
  - Similar to how NVMe devices are managed by NVMe class driver
- Memory manager will typically manage CXL attached memory
- CXL accelerator drivers can be vendor supplied, similar to PCIe accelerators



Hot-Plug

- CXL Protocol supports hot-add, managed hot-remove and surprise hot-remove
  - Managed hot-remove implies software is able to prepare the system for removal of the device (quiesce accelerator, flush caches, offline pages mapped to HDM etc.)
  - Surprise removal handled via CXL Isolation capability. Requires Root Port support and SW changes, SW recovery would be best effort
- Leverages Hot-plug model and Hot-plug elements as defined in PCI Express® Specification and appropriate Form Factor Specifications

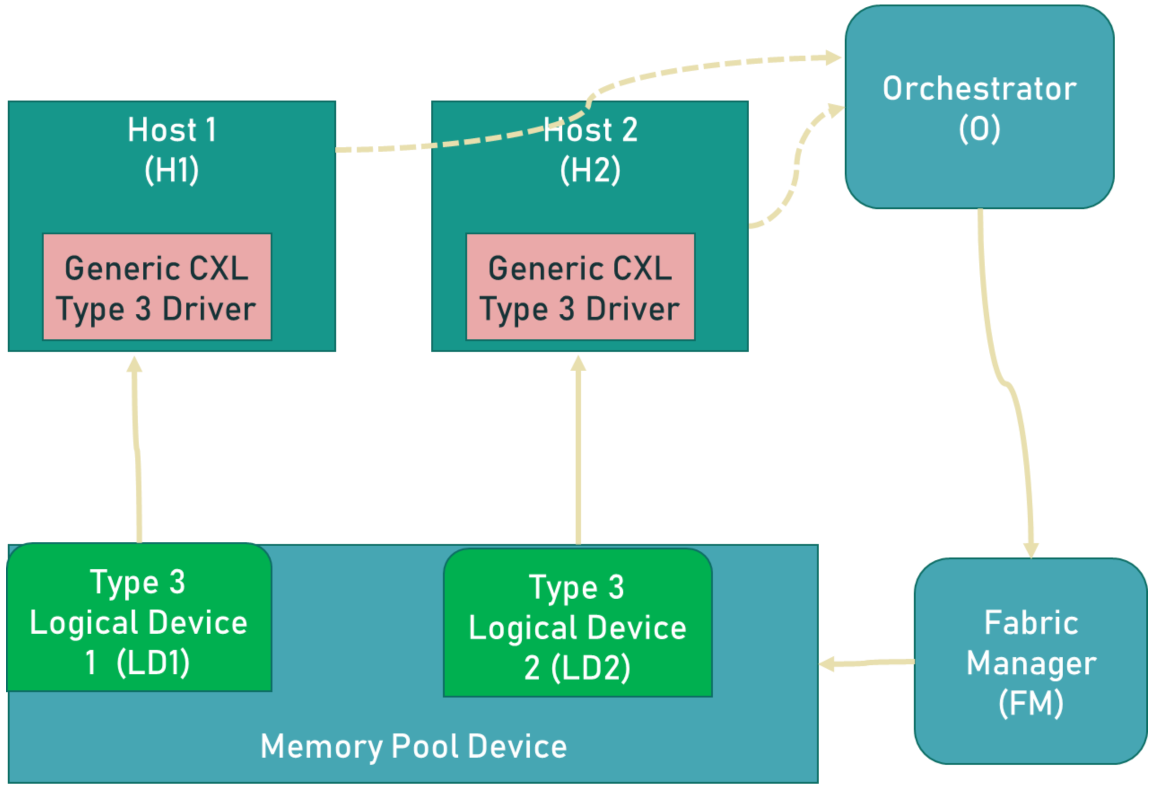
Type 3 Device Hot-add


System Firmware preps system for future hot-add
User hot-adds Type 3 Devices
PCIe® Hot-plug interrupt is fired
PCIe Hot-Plug handler Initialize .io, assigns BARs
PCIe like Steps
Discover HDM count and size(s)
Configure HDM Decoders
Configure various CXL DVSEC registers
Extract NUMA info from the Device
CXL Aware Software
Notify Mem Mgr of new capacity
Mem Mgr processes request
New capacity available to apps
Done
Standard Capacity-Add Steps

Dynamic Capacity Device (DCD)

- To exploit memory pooling, we need an abstraction that enables efficient page reassignment from one host to another
  - CXL hot-plug is a big hammer
- CXL defines
  - Events to notify host of capacity addition and removal requests
  - Acknowledgements from the host
  - Commands that allow FM to manage Pooling Device capacity and direct Pooling Device to add/remove capacity to/from a host
- A DCD can support up to 8 regions, each with different characteristics.
- Each region spans a DPA range.
- If a DCD supports shared memory region, this mechanism can be used to set up memory sharing across hosts connected to this device.
- A DCD can either be
  - Memory pooling device (Multi-logical device or Multi-headed device)
  - Fabric Attached device, specifically Global FAM Device (GFD)

DCD Sample Flow



Host 1
(H1)
Generic CXL Type 3 Driver
Host 2
(H2)
Generic CXL Type 3 Driver
Orchestrator
(O)
Type 3 Logical Device 1 (LD1)
Type 3 Logical Device 2 (LD2)
Memory Pool Device
Fabric Manager
(FM)

1. O decides to schedule Workload W onto H2. W requires 512 GB mem.
2. O notices H2 is using all of its mem, but H1 is not.
3. O asks FM to get 512 GB back from LD1
   - FM ask LD1 to release 512 GB mem
   - LD1 asks H1 to release 512 GB mem. H1 complies.
   - FM returns success
4. O asks FM to give 512 GB mem to LD2, with Tag=T_W . The tag value is opaque to FM and LD.
   - FM ask LD2 to offer 512 GB mem to associated host
   - LD2 offer H2 512 GB mem with Tag T. H2 accepts.
   - FM returns success
5. O schedules W onto H2. H2 assigns memory with Tag T_W to W.

CXL Performance Monitoring

- Today, Perf Tuning and Perf Debug activities rely on Counters located in different system components.
- CXL components can affect the overall system performance greatly
- CXL 3.0 spec standardize the register interface to CXL component performance counters and performance events,
- Allows generic software to be written and integrated with existing Performance Monitoring frameworks (e.g. Linux PMU)

CPMU (CXL Performance Monitoring Unit)

- A component can have multiple CPMU instances
- CPMU Registers are mapped in MMIO space
- CPMU interface consists of
  - CPMU capability register – number of counters, supported events etc.
  - Counter registers that count events
  - Counter configuration registers that control what is counted
  - Overflow status
- Counters can be configurable or fixed function
- Optional interrupting capability via MSI/MSI-X upon overflow
- CXL spec defines certain events and allows vendors to extend

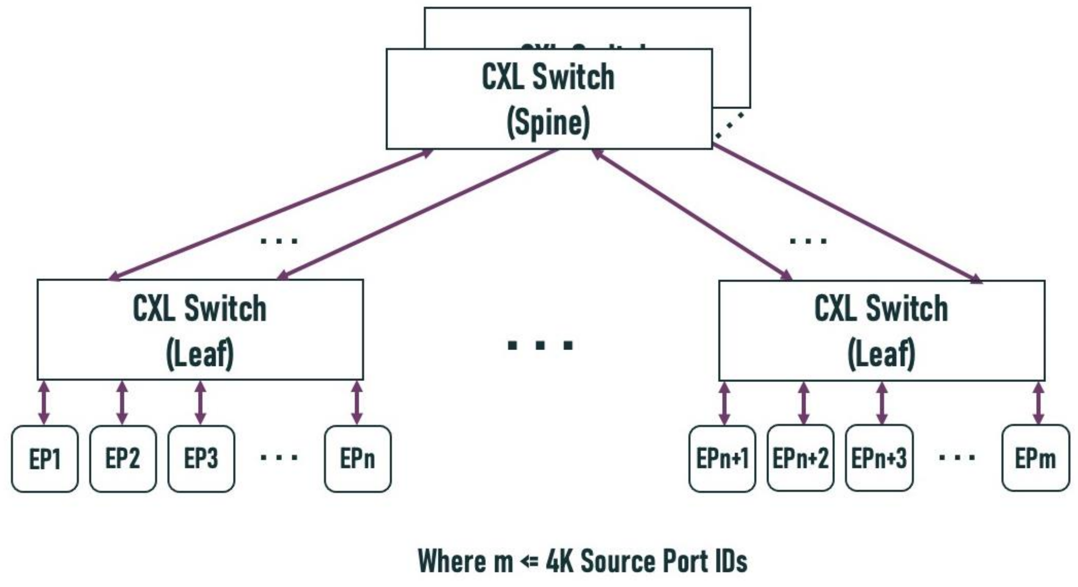
CXL Error Handling

- Protocol Errors
  - Errors are communicated between CXL devices and the host via messages over CXL.io
  - Errors are logged via PCIe® AER mechanisms as “Corrected Internal Error (CIE), or “Uncorrectable Internal Error (UIE)”.
- CXL memory device errors
  - Standardized access to error logs
    - Standard error log structure
    - Multiple error queues in the device, one per error severity
  - Standardized signaling
    - OS signaling via standard MSI/MSI-X
    - Firmware signaling via Error Firmware Notification VDM (MEFN)
    - WIP ECN adds alerting to BMC via MCTP

Fabrics and Fabric Management

CXL Fabrics – Motivation and Overview

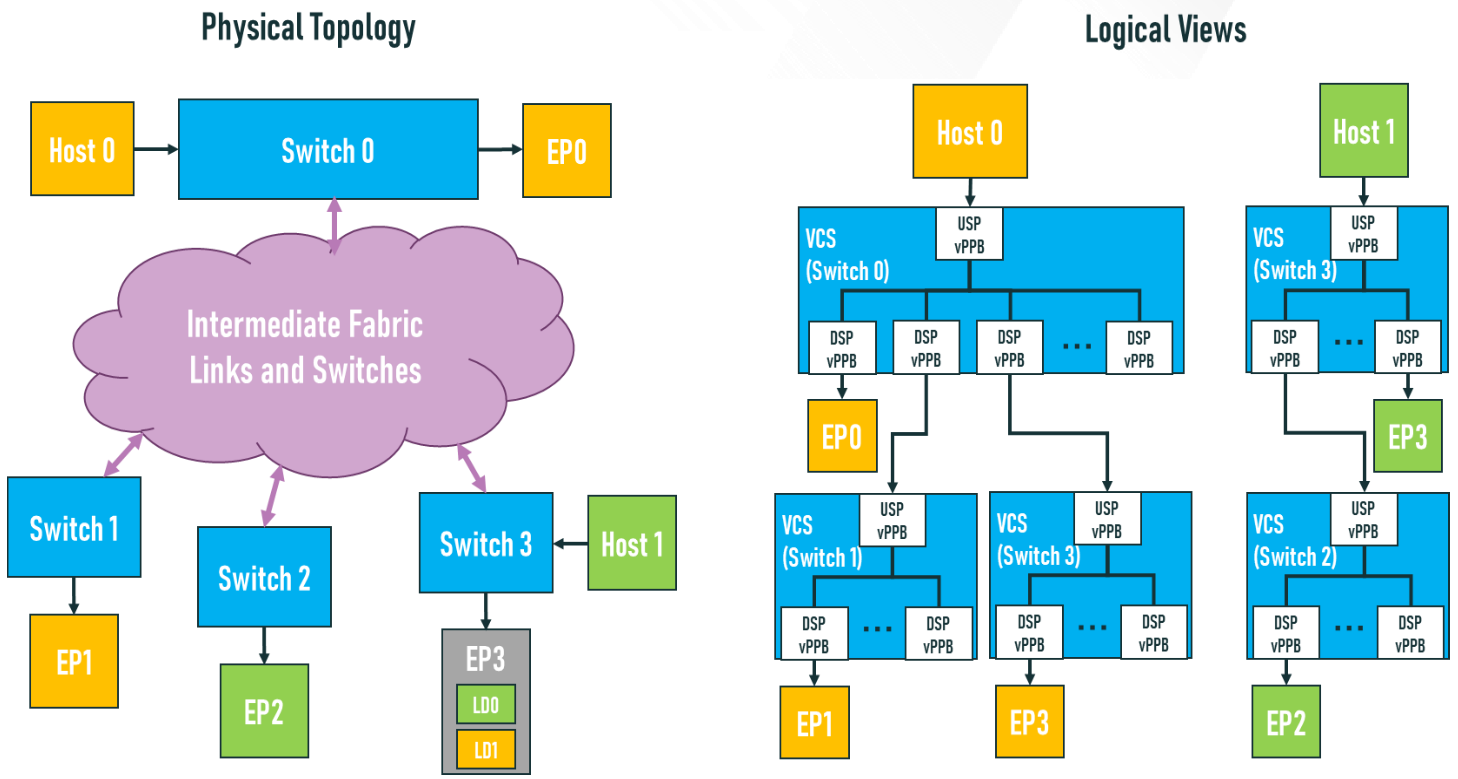
- Disaggregated, Composable Systems – Pooled Host, Device and Memory Resources
- Scale-out Systems – HPC/ML/Analytics
- Add capabilities to expand CXL from node to small number of racks
- Limited by 12b ID space (4k IDs)
- Scale beyond tree-based topologies
- Does not compromise node level properties



CXL Fabrics – Composability

EP binding from across fabric:

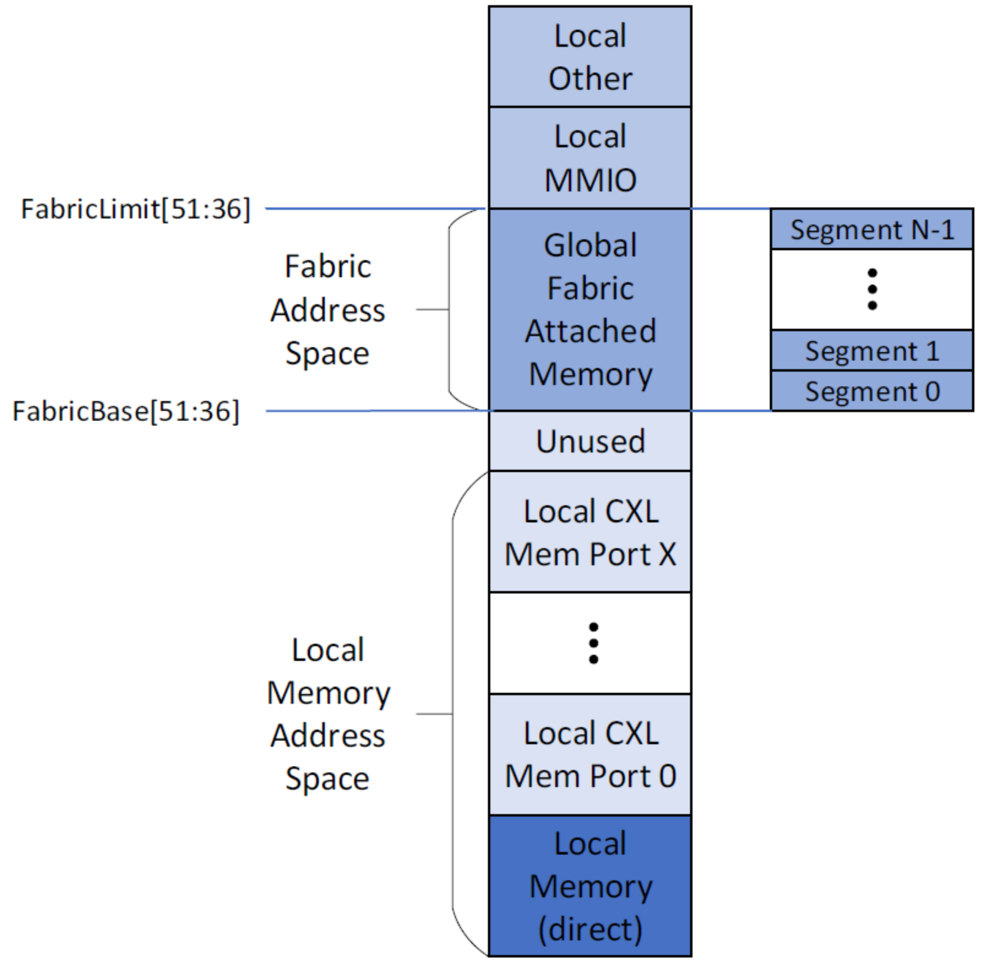
- Host sees up to 2 layers of standard switches: host edge and downstream edge
- Enables re-use of existing host SW



CXL Fabrics – Scale-Out

Global Fabric Attached Memory (G-FAM):

- Highly-scalable memory pool – e.g., 2000+ hosts accessing a memory pool of 2000+ G-FAM devices (GFDs)
- Accessible by all hosts through Fabric Address Segment Table (FAST)
- Hosts access HPA, GFD translates to DPA

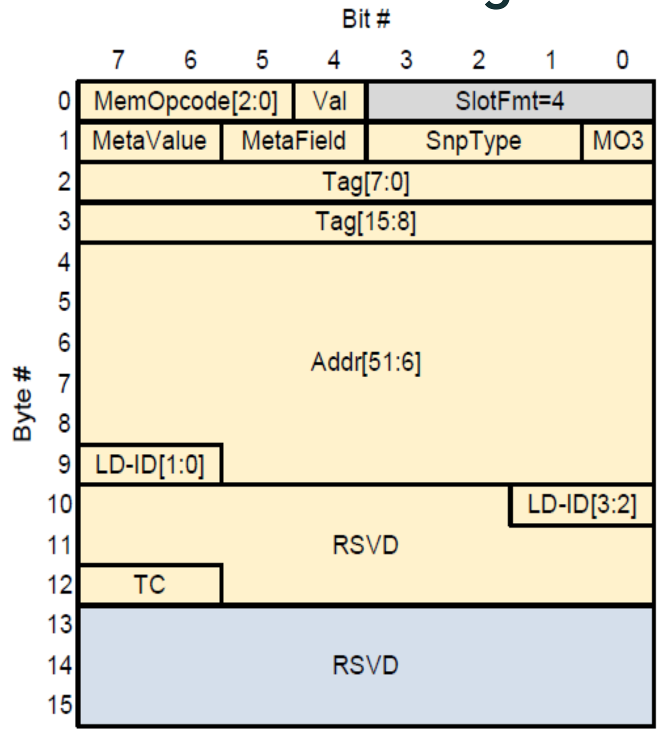


Transport Level Details

Transport Level Details

- Host-Based Routing (HBR)
- Covers CXL 1.1/2.0 transport protocol definitions
- Reads and writes requests are address routed
- Restricts links to a single VH – cannot resolve routing otherwise

256B Packing: G4/H4/HS4 HBR Message

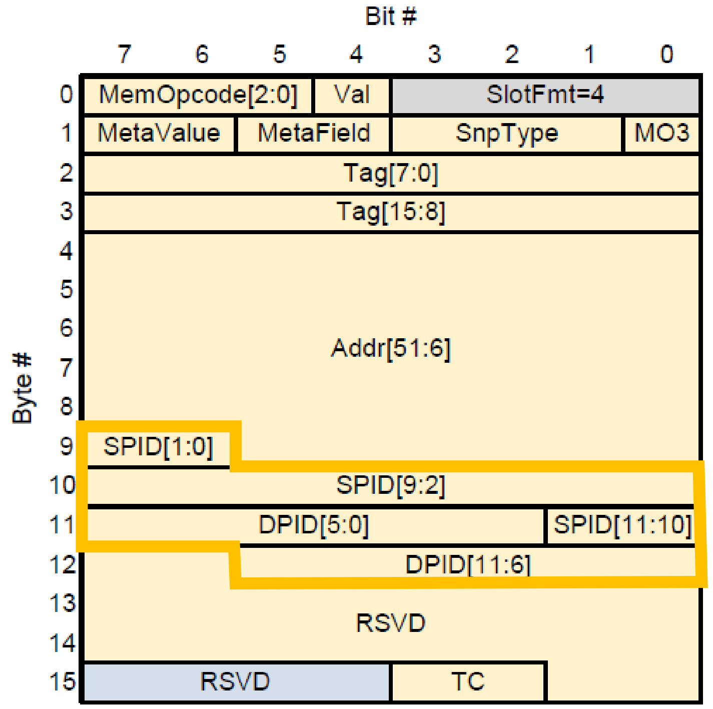


Transport Level Details

Port-Based Routing (PBR)

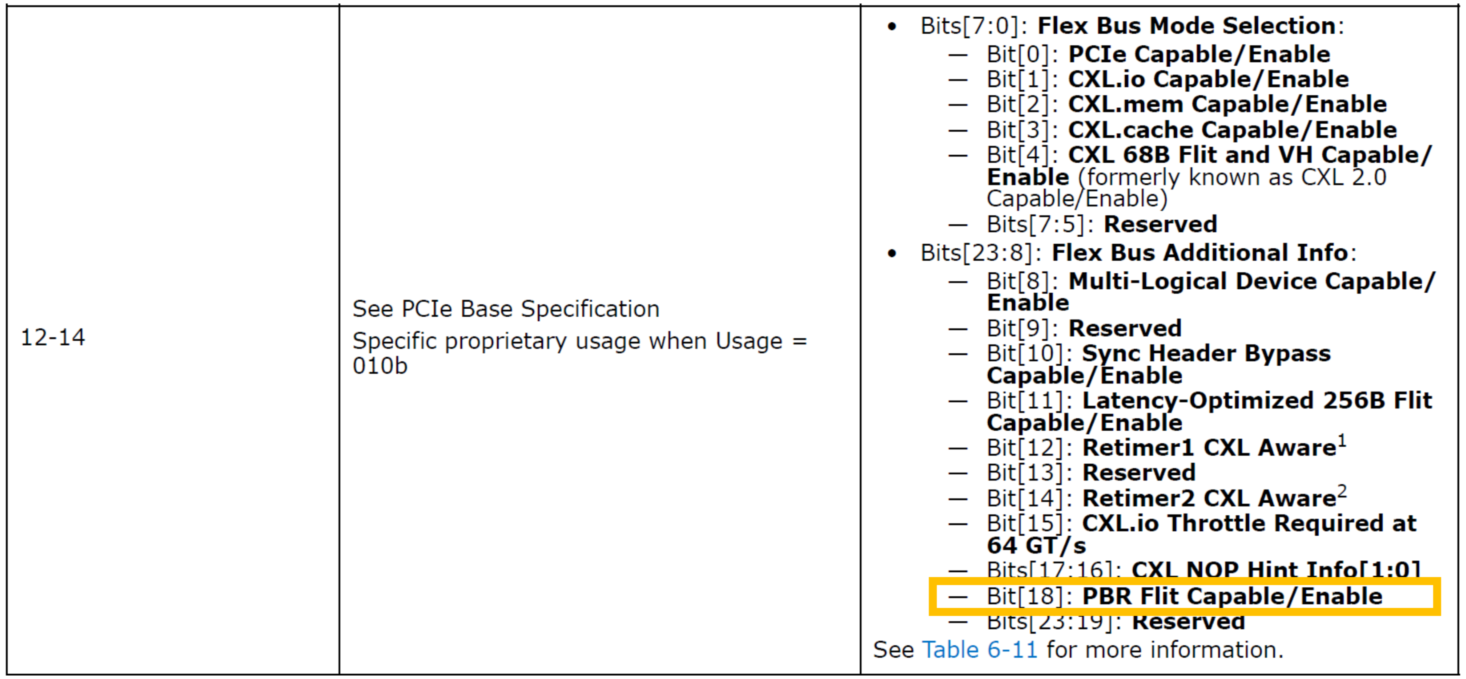
- Brand new flit mode in 3.0
- Transactions routed by PBR-ID:
  - Destination PBR-ID (DPID) carried by all transactions
  - Source PBR-ID (SPID) carried by select transactions as needed
- Inter-switch links can carry traffic from multiple VHs
- Supported only in 256B flit mode

256B Packing: G4/H4 PBR Message



Transport Level Details

PBR mode negotiated during Alternate Protocol Negotiation based on capabilities advertised in Symbols 12-14:

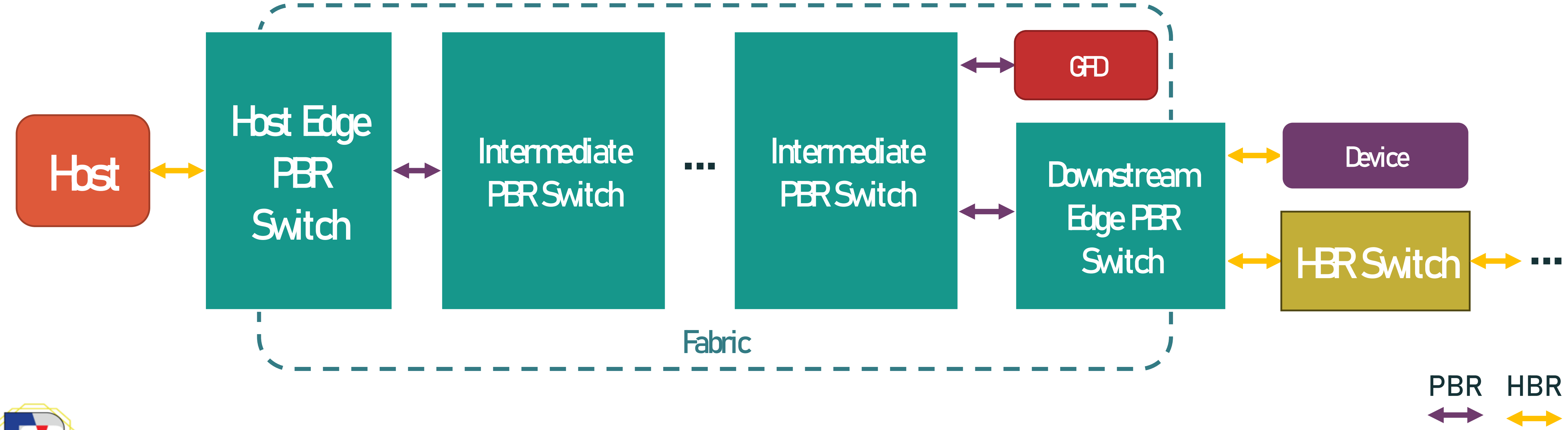


12-14	See PCIe Base Specification Specific proprietary usage when Usage = 010b	• Bits[7:0]: **Flex Bus Mode Selection**: — Bit[0]: **PCIe Capable/Enable** — Bit[1]: **CXL.io Capable/Enable** — Bit[2]: **CXL.mem Capable/Enable** — Bit[3]: **CXL.cache Capable/Enable** — Bit[4]: **CXL 68B Flit and VH Capable/ Enable** (formerly known as CXL 2.0 Capable/Enable) — Bits[7:5]: **Reserved** • Bits[23:8]: **Flex Bus Additional Info**: — Bit[8]: **Multi-Logical Device Capable/ Enable** — Bit[9]: **Reserved** — Bit[10]: **Sync Header Bypass Capable/Enable** — Bit[11]: **Latency-Optimized 256B Flit Capable/Enable** — Bit[12]: **Retimer1 CXL Aware**[1] — Bit[13]: **Reserved** — Bit[14]: **Retimer2 CXL Aware**[2] — Bit[15]: **CXL.io Throttle Required at 64 GT/s** — Bits[17:16]: **CXL NOP Hint Info[1:0]** — Bit[18]: **PBR Flit Capable/Enable** — Bits[23:19]: **Reserved** See Table 6-11 for more information.

Routing Model

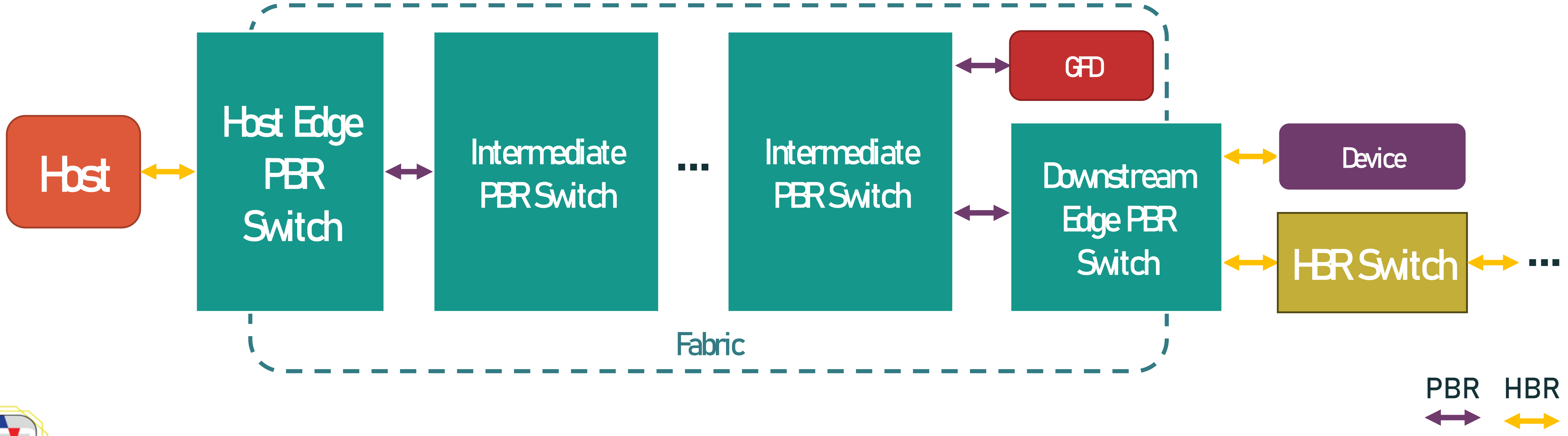
Routing Model

- Host requests begin in HBR format
- Host edge PBR switch converts to PBR format



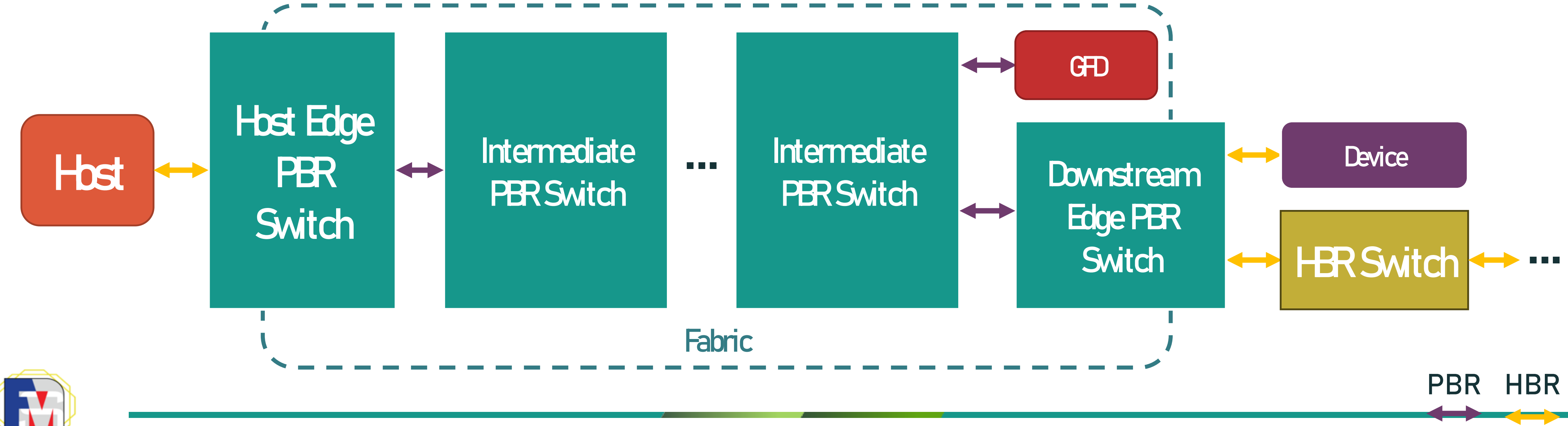
Routing Model

- Any intermediate switches route by DPID
- GFDs support PBR flit mode



Routing Model

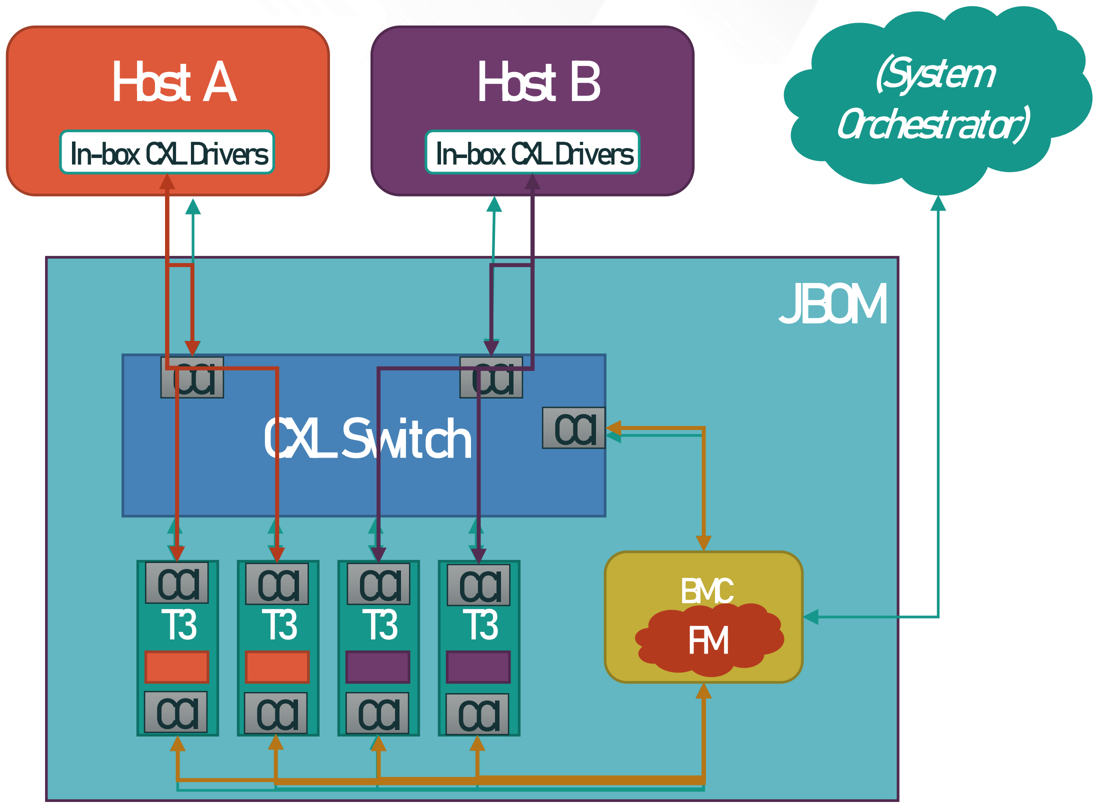
- All other EP types connected to Downstream edge switch
- Downstream edge converts to HBR



Fabric Management Architecture

New Management Challenges and Opportunities

- Leverage existing DRAM management
  - Minimize SW/FW development requirements
  - UEFI CDAT
  - SPD, PMIC, etc.
- New component architectures require extended topology reporting
  - PLDM Type 2 PDRs
  - Redfish models
- Standardization is key to deployment
  - Parallel efforts from CXL Consortium, JEDEC, and DMTF



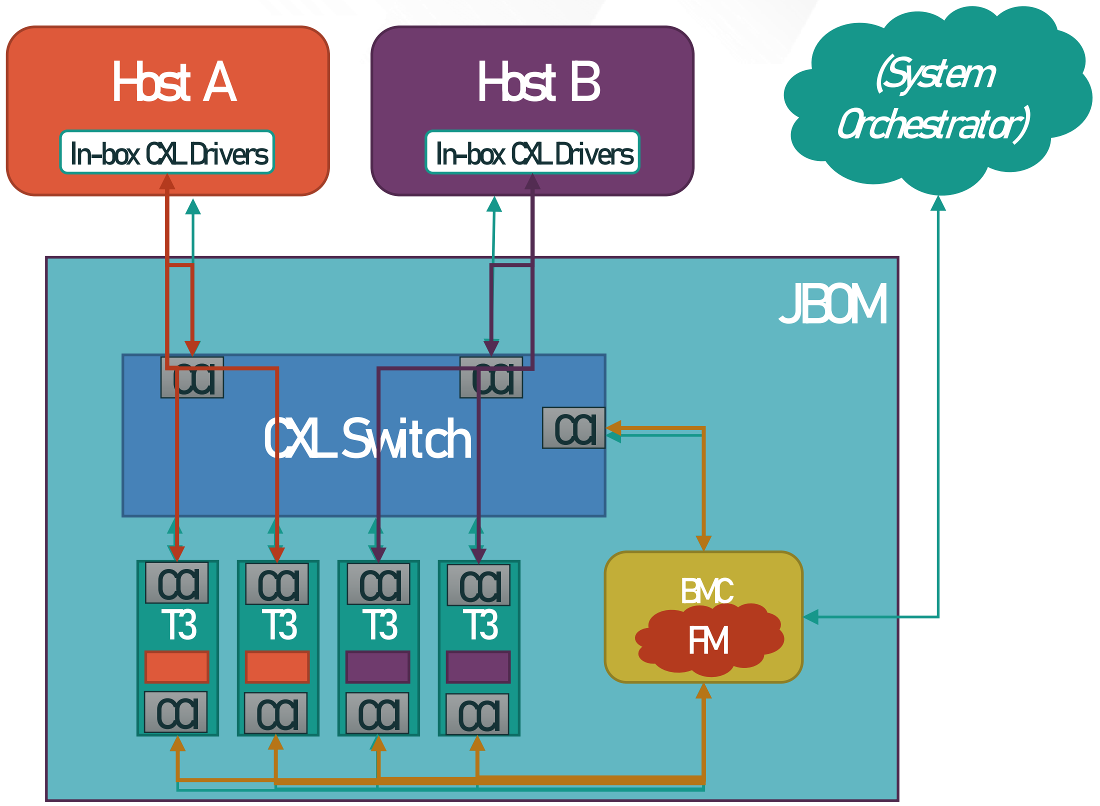
What is a Fabric Manager?

Fabric Manager (FM) is a conceptual term

Refers to the application-specific logic composing systems, allocating pooled resources, managing platforms, etc.

Can take many forms:

- BMC in a rack-mount appliance
- Management software running in a host
- Embedded FW in a CXL Switch



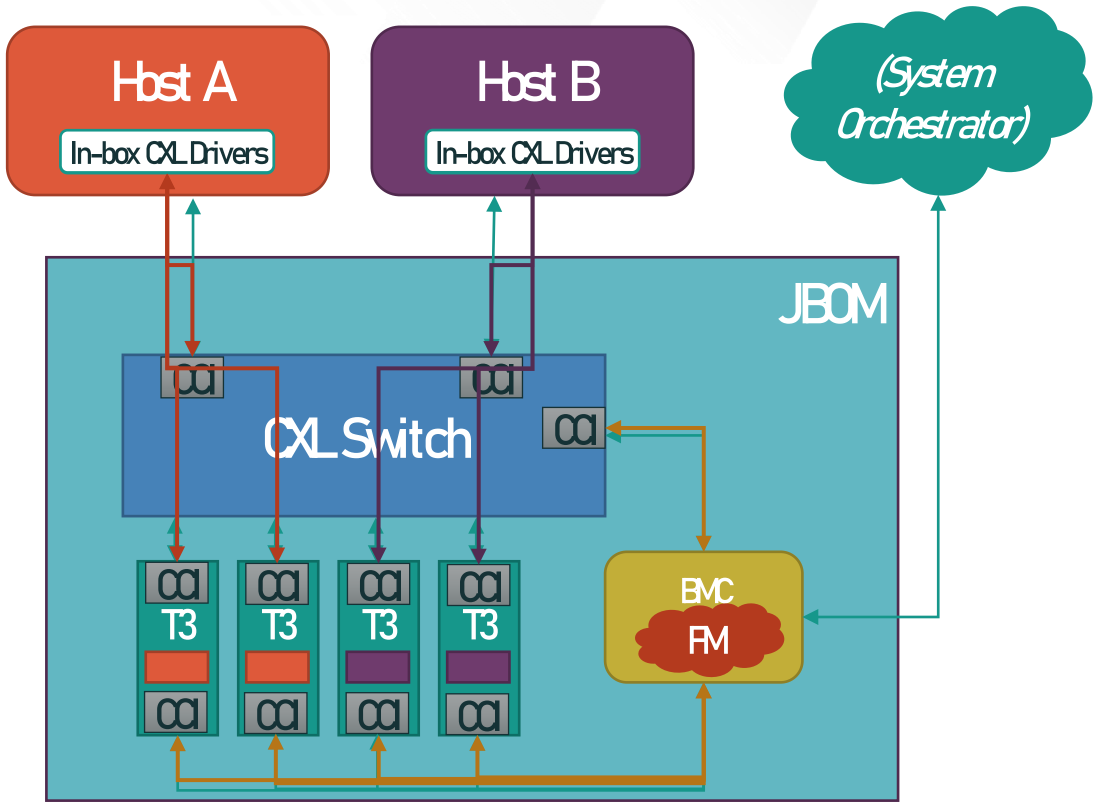
What is a Fabric Manager?

Framework is flexible by design to enable a wide variety of applications (embedded, automotive, hyperscale…)

Most management capabilities are optional

FM is required for advanced system operations:

- Composable systems – FM is responsible for assigning resources to hosts (e.g., binding ports/LDs to a VCS)
- Memory pooling – FM is responsible for allocating/deallocating memory, statically and dynamically



Component Command Interface

Component Command Interface

- Commands are processed by a Component Command Interface (CCI)
- Two types
  - Mailbox CCI – presented through memory registers
  - MCTP-based CCI – presented as an MCTP EP
- Not a queued interface
- Lengthy operations run as “Background Operations”
- A component may support multiple CCIs, with varying capabilities

Mailbox CCI

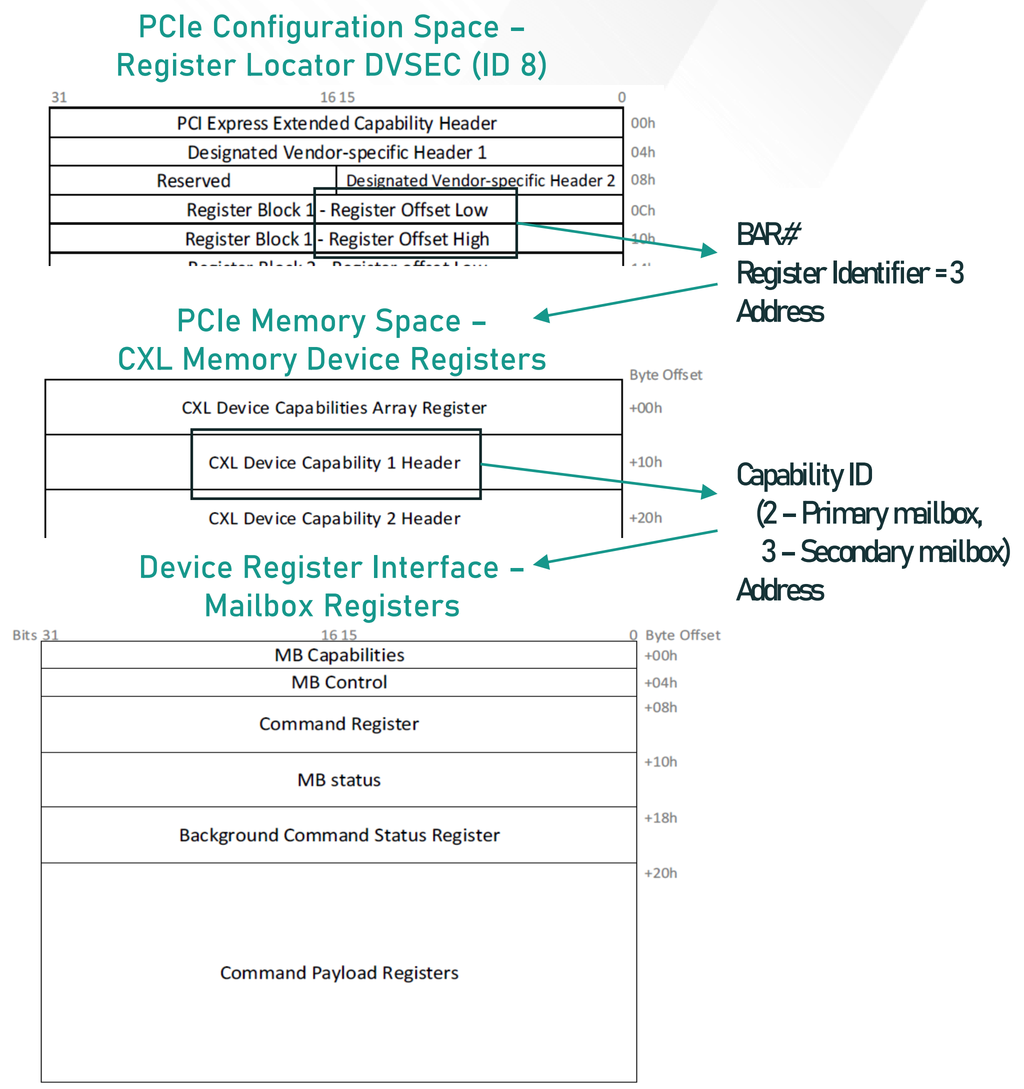
Located in PCIe® MMIO Space

Two types of mailbox:

- Primary – designed for use by driver; intended for privileged operations
- Secondary (optional) –designed for log/event record access; no interrupt or background operation support

Command inputs written to Command Payload Registers; outputs read from same region

Optionally generates MSI/MSI-X interrupts



PCI Express Extended Capability Header	00h
Designated Vendor-specific Header 1	04h
Reserved / Designated Vendor-specific Header 2	08h
Register Block 1 - Register Offset Low	0Ch
Register Block 1 - Register Offset High	10h

CXL Device Capabilities Array Register	+00h
CXL Device Capability 1 Header	+10h
CXL Device Capability 2 Header	+20h

MB Capabilities	+00h
MB Control	+04h
Command Register	+08h
MB status	+10h
Background Command Status Register	+18h
Command Payload Registers	+20h

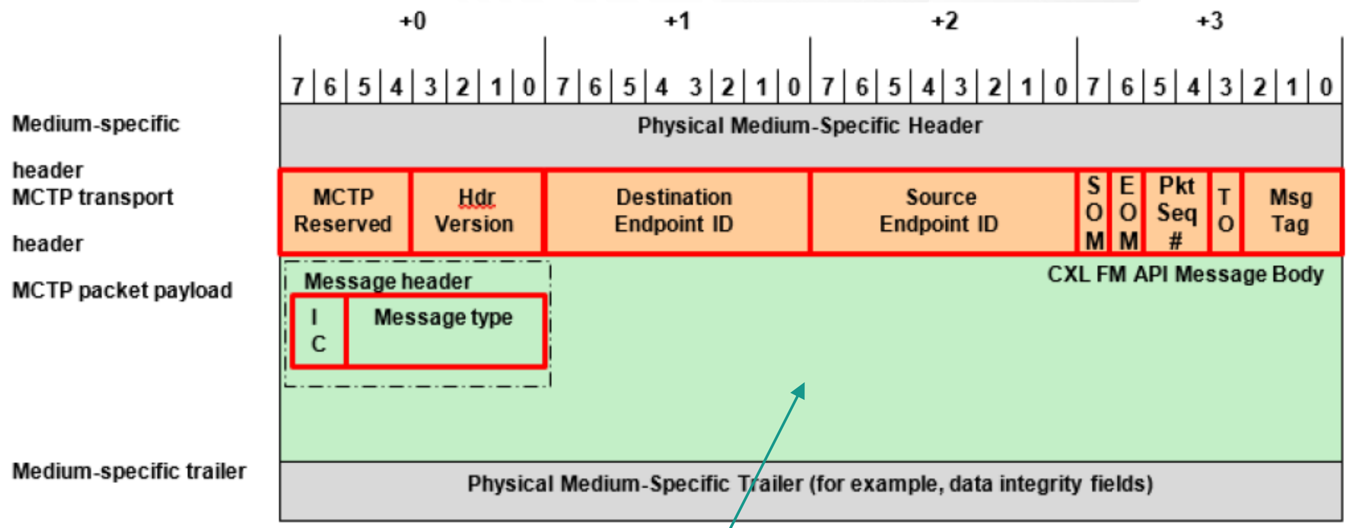
MCTP-based CCI

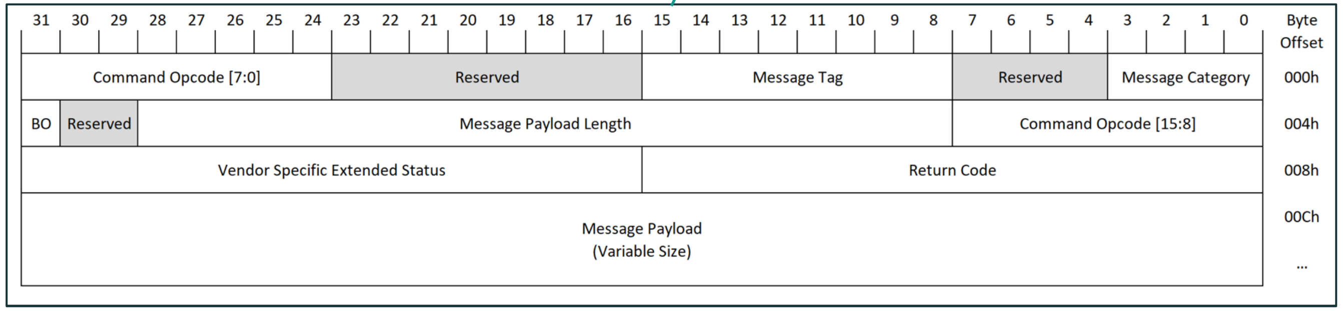
FM will first discover all MCTP EPs using MCTP spec-defined discovery

CCIs will advertise support for CXL Message Types

- Type 07h for FM API commands
- Type 08h for General and Memory Device Commands

Supported over any physical interface for which an MCTP binding spec is defined





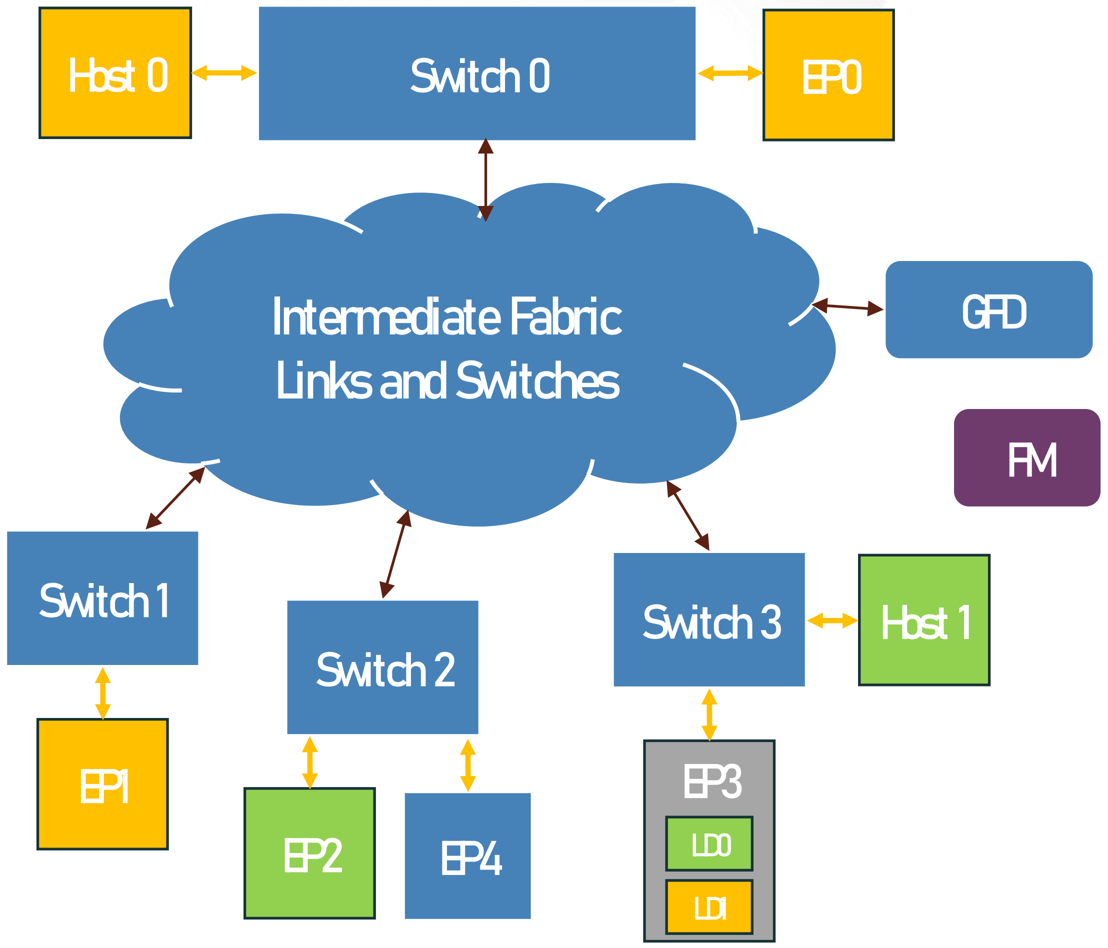
31–24	23–16	15–8	7–4	3–0	Byte Offset
Command Opcode [7:0]	Reserved	Message Tag	Reserved	Message Category	000h
BO / Reserved / Message Payload Length			Command Opcode [15:8]		004h
Vendor Specific Extended Status		Return Code			008h
Message Payload (Variable Size)					00Ch
					...

Fabric Management Architecture

Fabric Manager (FM) is responsible for:

- Fabric discovery and initialization
- Composition (binding)
- Inter-switch link management
- GFD Management
- Unbound EPs

Host manages its edge link and bound EPs



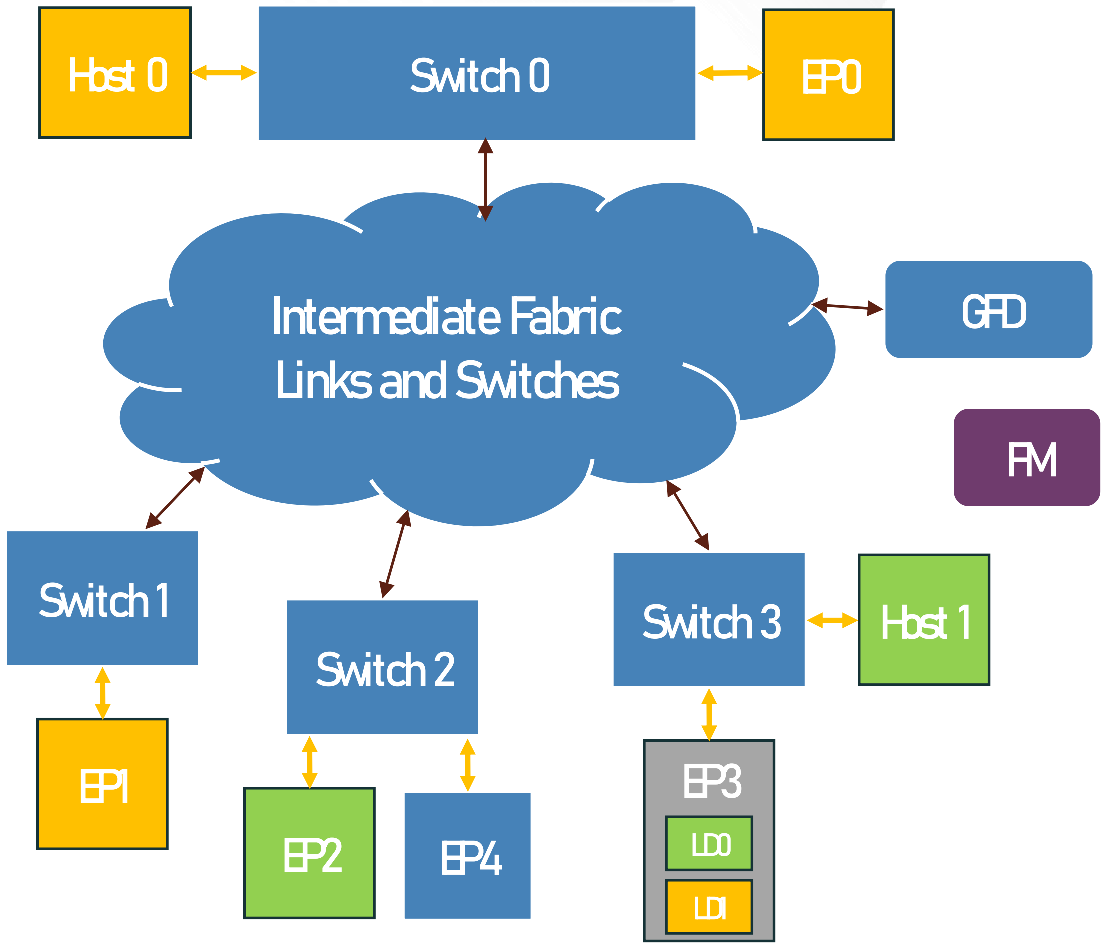
Fabric Management Architecture

Fabric Resource Management:

- Errors and scrub
- FW Update
- Binding/access control

Access Protection:

- Switch ingress
- Routing path
- Target (GFDs only)



Specification Roadmap

Specification Roadmap

Items to be defined in future specification release:

- CXL Fabric Management Specification
  - PBR Switch Management
  - GFD Management
- Host-to-Host communication
- Device-to-Device communication
- Cross-domain traffic

Conclusions

- CXL 3.0 features
  - Full fabric capabilities and fabric management
  - Expanded switching topologies
  - Symmetric coherency capabilities
  - Peer-to-peer resource sharing
  - Double the bandwidth and zero added latency compared to CXL 2.0
  - Full backward compatibility with CXL 2.0, CXL 1.1, and CXL 1.0

- Enabling new usage models
  - Memory sharing between hosts and peer devices
  - Support for multi-headed devices
  - Expanded support for Type-1 and Type-2 devices
  - GFAM provides expansion capabilities for current and future memory

- Call to Action
  - Download the CXL 3.0 specification
  - Support future specification development by joining the CXL Consortium
  - Follow us on Twitter and LinkedIn for updates!

CXL Compute Express Link™

Thank You

Flash Memory Summit