# Whitepaper

## CXL 1.1 vs. CXL 2.0: What's the Difference

June 2022

# CXL 1.1 vs. CXL 2.0: What's the Difference?

By Elad Shliselberg, System Architect, and Ronen Hyatt, CEO, UniFabriX Ltd.

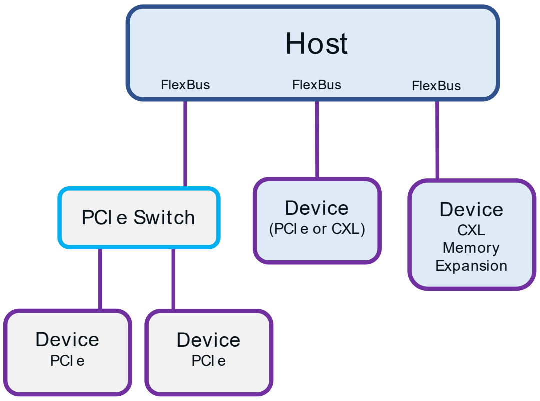
## Intro to CXL

Compute Express Link (CXL) is a cache-coherent interconnect, designed to be an industry-open standard interface, between platform functions, such as processors, accelerators, and memory. CXL 1.1 is the first productized version of CXL. It brings forward a world of possibilities and opportunities to improve upon the many strong features that exist in the PCIe arsenal. To name a few, CXL 1.1 introduces the concept of memory expansion, coherent co-processing via accelerator cache, and device-host memory sharing. The rich set of CXL semantics goes much beyond the familiar **cxl.io** (PCIe with enhancements), to offer also **cxl.cache**, and **cxl.mem**. These semantics are groups into **Device Types**: 1 (cxl.io/cxl.cache), 2 (cxl.io/cxl.cache/cxl.mem) and 3 (cxl.io/cxl.mem). Given the disruptive nature of CXL, its true value and potential ecosystem of applications are yet to be realized once it is deployed at scale. As the standard evolves, CXL 2.0 builds upon CXL 1.1 and uncovers new opportunities to further strengthen the robustness and scalability of the technology, while being **fully backwards compatible** with CXL 1.1.

In this white paper we will explore the fundamental capabilities of CXL and highlight the primary differences between CXL 2.0 vs. CXL 1.1 and the enhancements made as the protocol natively evolves.

## Topology Structure: From Direct Attachment to Switching

CXL 1.1 uses a simple topology structure of direct attachment between **Host** (such as CPU or GPU) and **CXL Device** (such as CXL accelerator or CXL-attached memory). **FlexBus** ports on the Host Root Complex use standard PCIe electricals and can support native **PCIe Devices** (optionally over a switched PCIe topology) or **CXL Devices** (direct attached only).

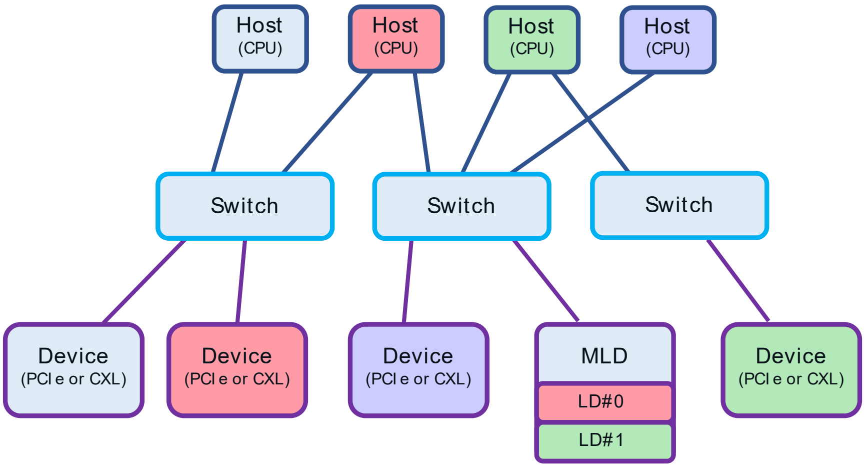


*CXL 1.1 Topology Example - Direct Attachment of CXL Devices & Memory Expansion*

CXL 2.0 further elaborates the topology structure by introducing a new single-level **CXL Switch** entity that provides a topology fanout of multiple Devices (CXL or PCIe) connected to multiple Hosts in a "mix and match" fashion for further expansion. This CXL Switch can be configured using an **FM (Fabric Manager)** that allows for each Host to get a different subset of CXL Devices. Whereas with CXL 1.1 each Device was able to connect only to a single Host, with CXL 2.0 a CXL Switch allows each Device to connect to multiple Hosts allowing for partitioning or demand-based provisioning of resources. This provides the ability to create completely separate virtual CXL hierarchies where each host sees a virtual CXL Switch beneath it and is led to believe that all the devices it sees are his alone.

## CXL 2.0 LDs (Logical Devices): SLD and MLD Resources

Now that with CXL 2.0 multiple Hosts can be connected to a single CXL Device, it may often times make sense to split the resources of such Device unto multiple Host recipients. In order to do this, a supporting CXL **MLD (Multi Logical Device)** can be split into up to **16 x LDs (Logical Devices)** identified by separate **LD-IDs**. Each of the LDs can then belong to a separate **VH (Virtual Hierarchy)**, have separate reset controls, and handed out by a CXL Switch to a particular Host as though it was a separate resource. For example, a CXL MLD memory expander that has 64GB of DRAM and 256GB of PMEM could be split to up to 16 isolated LDs, with some LDs exposing DRAM resources and other LDs exposing PMEM resources. In this way, a Host requiring additional memory could connect to LDs providing either DRAM or PMEM as needed. CXL 2.0 supports only Type 3 MLD components, that may represent memory resources or any other acceleration resources that are split and provisioned over time and into sets of capabilities.



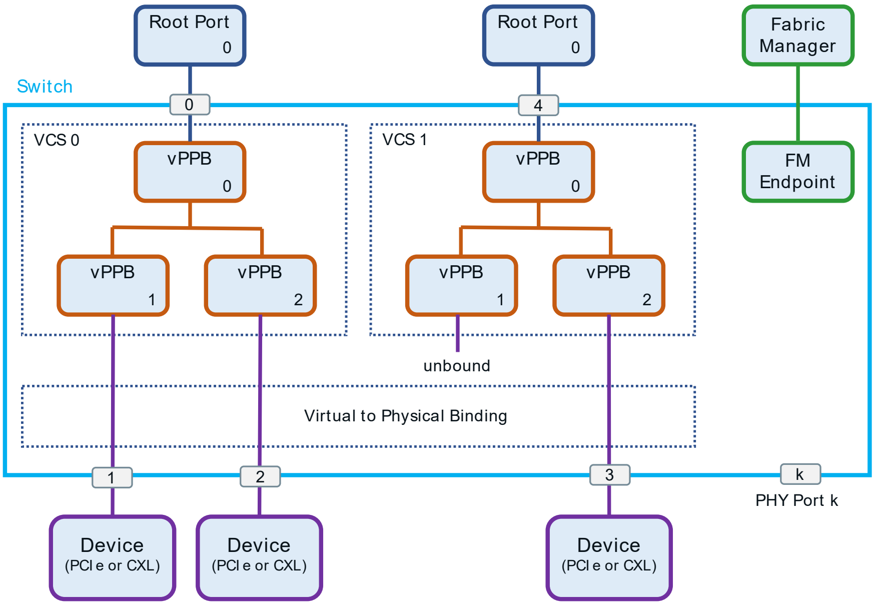
*CXL 2.0 Topology Example - Switches and an MLD (Multi Logic Device)*

## Fabric Management: Controlling the Elaborated Topology of CXL 2.0

The elaborated topology structure of CXL 2.0 that introduces CXL Switches and MLDs (Multi Logic Devices) makes a lot of sense when controlled by an **FM (Fabric Manager)**. The FM sits external to the CXL Switches and configures the internal CXL hierarchies within them. Each CXL hierarchy is defined by a **Virtual CXL Switch (VCS)** within the Switch so that a Host is completely isolated from all other

Hosts, by configuring internal **virtual PCIe-to-PCIe Bridges (vPPB)** and binding them to physical **PCIe-to-PCIe Bridges (PPB)** that attach directly to CXL Devices.

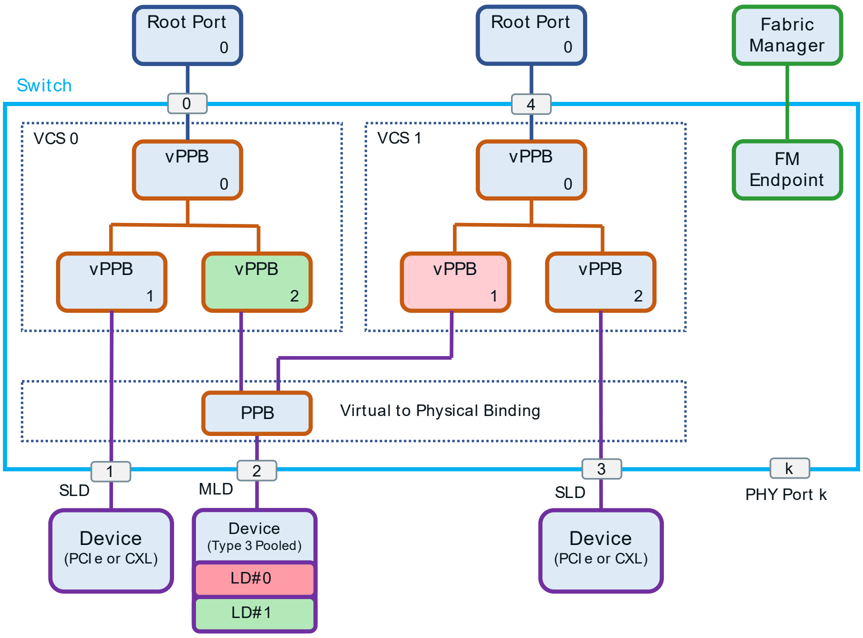
In favor of simplicity, CXL 2.0 also provides a static configuration that does not require an FM in the case where only SLDs are present.



*CXL 2.0 Switch with Multiple Internal VCSs (Virtual CXL Switches) Controlled by an FM (Fabric Manager)*

The FM can boot alongside the hosts or before them and execute changes to the bindings of physical devices to VCS in real time.

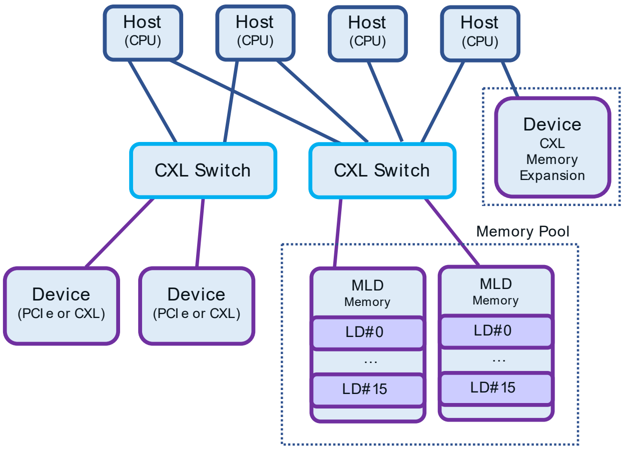
The FM can take any form, including software running on a Host machine, embedded software running on a **BMC (Baseboard Management Controller)**, embedded firmware running on another CXL Device or CXL Switch, or a state machine running within the CXL Device itself. Using a standardized API, the FM can send commands to LDs gathering error information, QoS and bandwidth information.



*1CXL 2.0 Topology - MLD Connected to Multiple VCS instances within a Switch*

## Resource Pooling Becomes a Reality with CXL 2.0

CXL 1.1 introduces **Memory Expansion** in the form of a CXL memory Device (normally **Type 3** but can be **Type 2** as well) directly attached to a particular Host. With CXL Switching and MLDs in CXL 2.0, come an entirely new world of **Resource Pooling** possibilities, such as **Memory Pooling**. With that, ideally every Host can access all the memory that it needs, dynamically on-demand, from a centralized large pool or from a set of pools. The Memory Pool may be composed under a CXL Switch spanning multiple CXL Devices, each may be an MLD providing different memory resources. Similarly, multiple Hosts may allocate accelerators on-demand from an Accelerator Pool.



*2CXL 2.0 Memory Pooling vs. CXL 1.1 way of Memory Expansion*

## CXL 2.0 Managed Hot Plug: Enabling On-Demand Provisioning

Large systems with multiple Hosts, Switches and Devices, that hold resources that can be provisioned on-demand, require methods of connecting and disconnecting these resources. It does not make sense to turn off the system every time a new device is added as it may require turning off a whole fleet of systems. 'Hot-Plug'ability has been an optional feature in the last few PCIe generations but has never really taken hold. Along with CXL Switches and FMs, CXL 2.0 introduces the concepts of **Hot-Add** and **Managed Hot-Removal**.

The Hot-Add model leverages the baseline scheme defined in PCIe with modifications to allow for all the new CXL capabilities. But unlike classic PCIe systems, the addition of **cache coherency** and **memory** requires special attention, so that device removal needs to be done using software request. This gives the Host the ability to flush any modifications and read-back or propagate memory.

### Memory QoS Telemetry

CXL provides QoS telemetry (DevLoad) methods, some optional and some mandatory, that allow the host to throttle and balance work request rates based on returned load telemetry. This opens the door to dynamic load balancing and intelligent memory mapping. With the addition of MLDs in CXL 2.0, QoS can be associated with LD and allow for memory devices to share load levels per resource.

### CXL 2.0 Speculative Memory Reads

Speculative memory read is a new command introduced in CXL 2.0 to support latency saving. The command hints a memory supporting device to prefetch a data for a possibly imminent memory read from the Host.

### Memory Interleaving

CXL 1.1 supports a basic interleaving method on multi-headed devices, when a cxl.mem device is connected to a single Host CPU via multiple FlexBus links. CXL 2.0 enables interleaving across multiple devices using physical address bits 14-8, allowing for **IG (Interleaving Granularity)** of different sizes. Every interleave set contains 1, 2, 4 or 8 devices.

CXL 2.0 also introduces **LSA (Label Storage Area)** which allows both interleave and namespace configuration details to be stored persistently on the device to preserve geometry configurations in a similar fashion to RAID arrays.

## Security Enhancements

While CXL 1.1 allows for proprietary security measures defined by Hosts and Devices, CXL 2.0 takes a more constructive approach. Leaning on the **SPDM (Security Protocol and Data Module)** defined by **DMTF (Distributed Management Task Force)**, CXL 2.0 adopts a comprehensive set of rules leveraged from **PCIe IDE** for a secure connection relying upon **AES-GCM** cryptography.

## Wrapup: What's the Difference?

CXL is a new frontier, enabling a fundamental wave of innovation in datacenter-related technologies. Whereas CXL 1.1 focuses on enhancements within the Server platform, such as **Memory Expansion**, CXL 2.0 goes out and beyond the server platform to define system-wide solutions such as **Memory Pooling** that serve multiple server platforms. CXL 2.0 further strengthens the **RAS (Reliability, Availability and Serviceability)** capabilities of the protocol that become even more important as the system span of CXL expands to larger diameters with CXL 2.0.

The CXL Consortium is working diligently to augment and strengthen the CXL ecosystem and bring new capabilities that will enable higher scalability, performance, and efficiency. Stay tuned!