Industry Open Standard for High Speed Communications

180+ Member Companies
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
- **Q3 2022**: CXL 3.0 Specification Release
• **New breakthrough high-speed interconnect**
  - Enables a high-speed, efficient interconnect between CPU, memory and accelerators
  - Builds upon PCI Express® 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 with heterogeneous processing
    - Enables memory disaggregation, memory pooling and sharing, persistent memory and emerging memory media

• **Delivered as an open industry standard**
  - CXL Specification 3.0 is available now with full backward compatibility with CXL 2.0 and CXL 1.1
  - Future CXL Specification generations will continue to innovate to meet industry needs with backward compatibility
What is CXL?

- Alternate protocol that runs across the standard PCIe physical layer
- Uses a flexible processor port that can auto-negotiate to either the standard PCIe transaction protocol or the alternate CXL transaction protocols
- CXL 2.0 and CXL 1.1 align to 32 GT/s PCIe 5.0
- CXL 3.0 aligns to 64GT/s PCIe 6.0 and is backward compatible
The CXL transaction layer is compromised of three dynamically multiplexed sub-protocols on a single link:

- **CXL.io**
  Discovery, configuration, register access, interrupts, etc.

- **CXL.cache**
  Device access to processor memory

- **CXL.Memory**
  Processor access to device attached memory
• All 3 representative usages have latency critical elements:
  • CXL.cache
  • CXL.memory
  • CXL.io

• CXL cache and memory stack is optimized for latency:
  • Separate transaction and link layer from IO
  • Fixed message framing

• CXL.io flows pass through a stack that is largely identical a standard PCIe stack:
  • Dynamic framing
  • Transaction Layer Packet (TLP)/Data Link Layer Packet (DLLP) encapsulated in CXL flits
Representative CXL Usages

Caching Devices / Accelerators
- CXL.io
- CXL.cache

Accelerators with Memory
- CXL.io
- CXL.cache
- CXL.memory

Memory Buffers
- CXL.io
- CXL.memory

**PROTOCOLS**
- DDR
- Processor
- Accelerator
- Cache

**USAGES**
- PGAS NIC
- NIC atomics
- GP GPU
- Dense computation
- Memory BW expansion
- Memory capacity expansion
- Storage class memory
Industry trends

• Use cases driving need for higher bandwidth: e.g., high performance accelerators, system memory, SmartNIC etc.

• CPU capability requiring more memory capacity and bandwidth per core

• Efficient peer-to-peer resource sharing/messaging across multiple domains

• Memory bottlenecks due to CPU pin and thermal constraints needs to be overcome

CXL 3.0 introduces...

• Double the bandwidth
  • Zero added latency over CXL 2.0

• 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
  • Direct memory/Peer-to-Peer accesses by devices
  • New symmetric memory capabilities
  • Improved software capabilities

• Full backward compatibility with CXL 2.0, CXL 1.1, and CXL 1.0

CXL 3.0 is a huge step function with fabric capabilities while maintaining full backward compatibility with prior generations
# CXL 3.0 Spec Feature Summary

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

MEMORY POOLING

1. Device memory can be allocated across multiple hosts.

2. Multi Logical Devices allow for finer grain memory allocation.
RECAP: CXL 2.0 FEATURE SUMMARY

SWITCH CAPABILITY

CXL 2.0

- Supports single-level switching
- Enables memory expansion and resource allocation
CXL 3.0: MULTIPLE LEVEL SWITCHING, MULTIPLE TYPE-1/2 Devices

Each host’s root port can connect to more than one device type (up to 16 CXL.cache devices)

Multiple switch levels (aka cascade)
- Supports fanout of all device types
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 maintain 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
Device memory can be shared by all hosts to increase data flow efficiency and improve memory utilization.

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

CXL 3.0 defined mechanisms to enforce hardware cache coherency between copies.
CXL 3.0: POOLING & SHARING

- Expanded use case showing memory sharing and pooling
- CXL Fabric Manager is available to setup, deploy, and modify the environment
- Shared Coherent Memory across hosts using hardware coherency (directory + Back-Invalid. 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
Nodes can be any combination:
- Hosts
- Type 1 – Device with cache
  - Example: Smart NIC
- Type 2 – Device with cache and memory
  - Example: AI Accelerator
- Type 3 – Device with memory
  - Example: memory expander
- PCIe Device
CXL 3.0: GLOBAL FABRIC ATTACHED MEMORY (GFAM) DEVICE

- CXL 3.0 enables Global Fabric Attached Memory (GFAM) architecture which differs from traditional processor centric architecture by disaggregating the memory from the processing unit and implements a shared large memory pool.

- Memory can be of the same type or different types which can be accessed by multiple processors directly connected to GFAM or through a CXL switch.
CXL 3.0: FABRICS EXAMPLE USE CASE

Machine Learning Accelerator and GFAM Device in a Fabric Architecture

GFAM enables multiple media types, i.e. DRAM, Flash, future memory types.
CXL 3.0: FABRICS EXAMPLE USE CASE

HPC/Analytics

Sharing memory and networking devices to reduce cost and improve efficiency
CXL 3.0: FABRICS EXAMPLE USE CASE

Composable Systems with Spine/Leaf Architecture

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
CXL 3.0 Summary

- **CXL 3.0 features**
  - Enhanced memory pooling and enables new memory usage models
  - Multi-level switching with multiple host and fabric capabilities and enhanced fabric management
  - New symmetric coherency capabilities
  - Improved software capabilities

- **CXL 3.0 introduces new usage models**
  - Delivers industry needs for higher bandwidth
  - Optimized system level flows with advanced switching, efficient peer-to-peer and fine-grained resource sharing across multiple domains

- **Call to Action**
  - Join CXL Consortium
  - Follow us on Twitter and LinkedIn for updates!