Compute Express Link™ 2.0: A High-Performance Interconnect for Memory Pooling

Presented by Andy Rudoff, Persistent Memory SW Architect, Intel
Compute Express Link™

A New Class of Interconnect
CXL Consortium

CXL Board of Directors

Open Industry Standard for Cache Coherent Interconnect

160+ Member Companies
Compute Express Link

- **CXL 1.1**
  - June 2019
- **CXL 2.0**
  - Nov 2020
  - 1.1 Compatible
  - Adds Pooling
  - Adds PMem

---

**Introducing CXL**

- Open industry standard for high bandwidth, low-latency interconnect
- Connectivity between host processor and accelerators/ memory device/ smart NIC
- Addresses high-performance computational workloads across AI, ML, HPC, and Comms segments
  - Heterogeneous processing: scalar, vector, matrix, spatial architectures spanning CPU, GPU, FPGA
  - Memory device connectivity
  - PCIe PHY completely leveraged with additional latency optimization
  - Dynamic multiplexing of 3 protocols
- Based on PCie® 5.0 PHY infrastructure
  - Leverages channel, retimers, PHY, Logical, Protocols
  - CXL.io – I/O semantics, similar to PCIe - mandatory
  - CXL.cache – Caching Semantics – optional
  - CXL.memory – Memory semantics - optional

www.computeexpresslink.org
Representative CXL Usages

Caching Devices / Accelerators
- Usages:
  - PGAS NIC
  - NIC atomics
- Protocols:
  - CXL.io
  - CXL.cache

Accelerators with Memory
- Usages:
  - GPU
  - FPGA
  - Dense Computation
- Protocols:
  - CXL.io
  - CXL.cache
  - CXL.memory

Memory Buffers
- Usages:
  - Memory BW expansion
  - Memory capacity expansion
  - 2LM
- Protocols:
  - CXL.io
  - CXL.mem

(Type 1 Device)  (Type 2 Device)  (Type 3 Device)
CXL 2.0 Memory Pooling

Benefit of CXL 2.0 Switching

Pooling

Memory/Accelerator Pooling with Single Logical Devices

Memory Pooling with Multiple Logical Devices

CXL 2.0 Switch

Standardized CXL Fabric Manager

D1 D2 D3 D4 D# H1 H2 H3 H4 H# D1 D2 D3 D4 D# H1 H2 H3 H4 H#
Benefits of Persistent Memory

- Moves Persistent Memory from Controller to CXL
- Enables Standardized Management of the Memory and Interface
- Supports a Wide Variety of Industry Form Factors

Latency (nanoseconds):
- CPU: $10^0$
- DRAM CXL 1.1/1.0: $10^1$
- CXL 2.0: $10^2 - 10^3$
- Performance SSD: $10^4$
- Capacity SSD: $10^5$
- HDD: $10^6$

CXL + PM Fills the Gap!
Persistent Memory Today

Connecting to the Memory Bus
Connecting the Memory Bus
Intel's Approach for Optane PMem

![Diagram showing the memory bus architecture with Memory Controller, Core, Load/Store, PMem, and DDR.]
The ACPI “NVDIMM” Framework

The SNIA NVM Programming Model
The ACPI “NVDIMM” Framework

The SNIA NVM Programming Model

NVDIMM Firmware Interface Table (ACPI 6.0)
The ACPI “NVDIMM” Framework
The ACPI “NVDIMM” Framework
### Learnings from ACPI Based Approach

#### Pros
- ACPI NFIT Unified NVDIMM-N and Intel’s Optane PMem
  - Helped get enabling upstream early
- _DSMs allowed a generic kernel implementation
  - Differences abstracted away by _DSMs
- Mechanism has evolved gracefully
  - Fairly small additions, few errata

#### Cons
- ACPI dynamic support doesn’t scale
  - Hot plug challenging
  - Meant for small number of empty sockets
- _DSM complexity hard to maintain
  - Bug fixes, additions logistical challenges
  - Virtually impossible to support multiple devices
  - No generic BIOS
Adding PMem to CXL

The CXL 2.0 Specification
CXL 2.0 Changes for PMem

- Most changes should apply to all memory types
  - Minimize PMem-specific changes, rest apply to volatile memory too
- PCIe enumeration
  - NFIT isn’t used for CXL devices (they aren’t NVDIMMs!)
  - Leverage PCIe frameworks, including hot plug
- MMIO registers
  - Mailbox interface, etc.
- Command Interface
  - Was vendor-private for NVDIMMs
- SW Guide for Driver Writers
  - https://tinyurl.com/7eyje4pu
Mailbox Commands

- Was vendor private
- Standards are a double-edged sword
  - Generic Drivers
  - Committee visit for every change
- Learnings from NVDIMMs helped
  - Leverage what worked
  - Fix pain points
- Some commands apply to all CXL →
Memory Device Commands

- Most added commands →
  - DSMs are gone
    - OS uses mailbox directly
    - BIOS too
- Much complexity moved:
  - From BIOS to OS
- Allows generic:
  - BIOS
  - OS Drivers

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Command Set</th>
<th>Command Block</th>
<th>Required</th>
<th>Input Payload Size (B)</th>
<th>Output Payload Size (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40h</td>
<td>Identify</td>
<td>00h</td>
<td>4000h</td>
<td>M 0</td>
<td>42h</td>
</tr>
<tr>
<td>00h</td>
<td>Get Partition Info</td>
<td>4100h</td>
<td>O 0</td>
<td>20h</td>
<td></td>
</tr>
<tr>
<td>01h</td>
<td>Set Partition Info</td>
<td>4101h</td>
<td>O 0An 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02h</td>
<td>Get LSA</td>
<td>4102h</td>
<td>PM 8 8 0+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03h</td>
<td>Set LSA</td>
<td>4103h</td>
<td>PM 4+ 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04h</td>
<td>Get Health Info</td>
<td>4200h</td>
<td>M 0</td>
<td>12h</td>
<td></td>
</tr>
<tr>
<td>05h</td>
<td>Get Alert Configuration</td>
<td>4201h</td>
<td>M 0</td>
<td>15h</td>
<td></td>
</tr>
<tr>
<td>06h</td>
<td>Get Shutdown State</td>
<td>4202h</td>
<td>M 0Ch 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07h</td>
<td>Set Shutdown State</td>
<td>4203h</td>
<td>PM 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08h</td>
<td>Get Poison List</td>
<td>4204h</td>
<td>PM 10h 20h+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09h</td>
<td>Get Scan Media Capabilities</td>
<td>4300h</td>
<td>O 8 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0ah</td>
<td>Clear Poison</td>
<td>4301h</td>
<td>O 8 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0bh</td>
<td>Get Scan Media Results</td>
<td>4302h</td>
<td>O 48h 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0ch</td>
<td>Get Scan Media</td>
<td>4303h</td>
<td>PM 10h 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0dh</td>
<td>Scan Media</td>
<td>4304h</td>
<td>PM 11h 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0eh</td>
<td>Get Scan Media Results</td>
<td>4305h</td>
<td>PM 0 20h+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0fh</td>
<td>Sanitize</td>
<td>4400h</td>
<td>O 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01h</td>
<td>Secure Erase</td>
<td>4401h</td>
<td>O 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Example: Identify Memory Device

### Identify Memory Device Output Payload

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10h</td>
<td><strong>FW Revision</strong>: Contains the revision of the active FW formatted as an ASCII string. This is the same information that may be retrieved with the Get FW Info command.</td>
</tr>
<tr>
<td>10h</td>
<td>8</td>
<td><strong>Total Capacity</strong>: This field indicates the total usable capacity of the device. Expressed in multiples of 256 MB. Total device usable capacity is divided between volatile only capacity, persistent only capacity, and capacity that can be either volatile or persistent. Total Capacity shall be greater than or equal to the sum of Volatile Only Capacity and Persistent Only Capacity.</td>
</tr>
<tr>
<td>18h</td>
<td>8</td>
<td><strong>Volatile Only Capacity</strong>: This field indicates the total usable capacity of the device that may only be used as volatile memory. Expressed in multiples of 256 MB.</td>
</tr>
<tr>
<td>20h</td>
<td>8</td>
<td><strong>Persistent Only Capacity</strong>: This field indicates the total usable capacity of the device that may only be used as persistent memory. Expressed in multiples of 256 MB.</td>
</tr>
<tr>
<td>28h</td>
<td>8</td>
<td><strong>Partition Alignment</strong>: If the device has capacity that may be used either as volatile memory or persistent memory, this field indicates the partition alignment size. Expressed in multiples of 256 MB. Partitionable capacity is equal to Total Capacity - Volatile Only Capacity - Persistent Only Capacity. If 0, the device doesn’t support partitioning the capacity into both volatile and persistent capacity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Informational Event Log Size</strong>: The number of events the device can record.</td>
</tr>
</tbody>
</table>
- CXL 2.0 Spec defines commands
  - Can be terse, spec language
- Platforms decide some of the details
  - Example: BIOS on Intel platforms
- Document flows, algorithms
Interleaving

- HDM Decoders
  - Allow interleaving across devices
  - New to PCIe: interleave sets
- Important concept for PMem
  - For volatile memory, changing the interleave may impact performance
  - For PMem, changing the interleave loses your data
- Label Storage Area
  - Defined in CXL 2.0 spec
  - Provides region (interleave set) and namespace configuration
Label Example

Host Bridge
2 way@4k

Switch
4 way@1k

Device 0
Device 1
Device 2
Device 3

Device 4
Device 5
Device 6
Device 7

Label | Label | Label | Label
--- | --- | --- | ---
N 8 | N 8 | N 8 | N 8
Pos 0 | Pos 1 | Pos 2 | Pos 3
IG 1k | IG 1k | IG 1k | IG 1k

Label | Label | Label | Label
--- | --- | --- | ---
N 8 | N 8 | N 8 | N 8
Pos 4 | Pos 5 | Pos 6 | Pos 7
IG 1k | IG 1k | IG 1k | IG 1k

SDC 2021 Conference
Hot Plug

- Handled by OS
  - No BIOS in flow
- Uses “windows”
  - BIOS provided
- Flow used for PMem
  - Except boot dev
Flush-on-fail To Persistence

- Global Persistent Flush (GPF)
  - Analogous to ADR or eADR with NVDIMMs
- Simple when it works
  - Application just relies on it
- More complex when it fails
  - Dirty Shutdown Count
  - Already part of the programming model
  - Code written for NVDIMM still works correctly
    - Provided the model was followed correctly
SW Enabling

- Preliminary generic Type 3 CXL Driver already upstream in Linux
  - Orchestrated by NVDIMM framework maintainer Dan Williams

- QEMU patches emulating CXL Type 3 devices posted
  - Written by Ben Widawsky, provided a development platform for the driver

- Cross-company, cross committee collaboration on specifications
  - Prevents messy collisions from different implementation decisions
Summary

- The programming model remains the same
  - Applications written to the SNIA programming model continue to work
- CXL offers:
  - Moving PMem off the memory bus
  - Scalability (all types of memory)
  - Flexibility
- PMem on CXL specified as of CXL 2.0, published last November
  - OS enabling is emerging
Please take a moment to rate this session.

Your feedback is important to us.