Accelerating Flash Memory with the High Performance, Low Latency, OpenCAPI Interface

Allan Cantle, CTO & Founder, Nallatech/Molex
Marcy Byers, Processor Development, IBM
Nallatech at a Glance

Server qualified accelerator cards featuring FPGAs, network I/O and an open architecture software/firmware framework. Design Services/Application Optimisation

- Nallatech – a Molex company
- 25 years of FPGA heritage
- Energy-Efficient High Performance Heterogeneous Computing
- Real-time, low latency network and I/O processing
- Intel PSG (Altera) OpenCL partner
- Xilinx Alliance partner
- Server partners: Cray, DELL, HPE, IBM, Lenovo
- Application porting & optimization services
- Successfully deployed high volumes of FPGA accelerators
Hyperconvergence vs Disaggregation – An unavoidable Oxymoron?

Can we hyperconverge & disaggregate Flash Memory at the same time?

**Hyperconverged Architectures**
- CPU Centric Playbook
- Best Single Threaded Performance
- Tightest of CPU/Accelerator Coupling
  - Holy Grail = $\infty$ Bandwidth & 0 Latency
- Easier acceleration of Legacy Code
- PCIe is today’s convergence bus
  - E.g. NVMe SSDs

**Disaggregated Architectures**
- Data Centric Playbook
- Heterogeneous & Distributed compute
- Prioritize Application Dataflow needs
- Can put congestion back into the Network
- Latency managed, compute => data
- Ethernet is today’s disaggregation fabric
  - E.g. NVMe-oF
Hyperconverged accelerations quest for $\infty$ Bandwidth & 0 Latency

» Moving a single thread of data from the CPU to an accelerator can negate the acceleration benefit
  » therefore Acceleration > overhead of data movement to/from accelerator
» Partition code for minimum data movement and maximum acceleration

![Diagram](image.png)
Hyperconverged tightly Coupled FPGA Acceleration is not new.....

- Intel, Xilinx, Nallatech & ISI collaborated on FSB & QPI attached Accelerators
- Started Circa 2007
- Despite a decade of ongoing efforts, commercial reality has been elusive
Why are Tightly Coupled FPGA Accelerators so challenging?

» Tied to complex proprietary coherent busses
  » Rapid cadence of bus standards
  » Limited interface documentation
  » Onerous licensing terms

» Coherent busses not natively designed with FPGAs in mind
  » Pushes limits of FPGA’s capabilities

» Heavy burden on FPGA resources for interface IP
  » Impacts performance, in particular latency
  » Reduces resources available for acceleration

» Can drag down the performance of native CPUs using the same bus

OpenCAPI addresses these issues
OpenCoherent Accelerator Processor Interface

OpenCAPI™ Overview

Open Compute Project 2018
OpenCAPI Key Attributes

1. Architecture agnostic bus – Applicable with any system/microprocessor architecture
2. Optimized for High Bandwidth and Low Latency
3. High performance 25G interface design with zero ‘overhead’
4. Coherency - Attached devices operate natively within application’s user space and coherently with host microprocessor
5. Virtual addressing enables low overhead with no Kernel, hypervisor or firmware involvement
6. Wide range of Use Cases and access semantics
7. CPU coherent device memory (Home Agent Memory)
8. Architected for both Classic Memory and emerging Advanced Storage Class Memory
Comparison of Memory Paradigms

**Main Memory**
- Processor Chip
- DLx/Tlx
- DDR4/5

**Emerging Storage Class Memory**
- Processor Chip
- DLx/Tlx
- SCM

**Tiered Memory**
- Processor Chip
- DLx/Tlx
- DDR4/5
- SCM

---

**OpenCAPI 3.1 Architecture**
Ultra Low Latency ASIC buffer chip adding +5ns on top of native DDR direct connect!!

**Storage Class Memories** have the potential to be the next disruptive technology.... Examples include ReRAM, MRAM, Z-NAND...... All are racing to become the defacto

**Storage Class Memory tiered with traditional DDR Memory** all built upon OpenCAPI 3.1 & 3.0 architecture.
Still have the ability to use Load/Store Semantics
Acceleration Paradigms with Great Performance

Examples: Machine or Deep Learning such as Natural Language processing, sentiment analysis or other Actionable Intelligence using OpenCAPI attached memory

Examples: Encryption, Compression, Erasure prior to delivering data to the network or storage

Examples: Database searches, joins, intersections, merges Only the Needles are sent to the processor

OpenCAPI WINS due to Bandwidth to/from accelerators, best of breed latency, and flexibility of an Open architecture

Examples: Video Analytics, Network Security, Deep Packet Inspection, Data Plane Accelerator, Video Encoding (H.265), High Frequency Trading, etc

Examples: NoSQL such as Neo4J with Graph Node Traversals, etc
TLx and DLx Reference Designs in an FPGA

- TLx and DLx will be provided as reference designs to OpenCAPI consortium members
  - Associated reference design specifications for TLx and DLx will also be delivered along with RTL
- TLx and DLx are not symmetric with OTL and ODL that are on the host processor
- Designed to operate at 400MHz
- Xilinx Vivado 2017.1 TLx and DLx Statistics on **VU3P** Device

<table>
<thead>
<tr>
<th>VU3P Resources</th>
<th>CLB FlipFlops</th>
<th>LUT as Logic</th>
<th>LUT Memory</th>
<th>Block Ram Tile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DLx</strong></td>
<td>9392/788160</td>
<td>19026/394080</td>
<td>0/197280</td>
<td>7.5/720</td>
</tr>
<tr>
<td></td>
<td>(1.19%)</td>
<td>(4.82%)</td>
<td>(0%)</td>
<td>(1.0%)</td>
</tr>
<tr>
<td><strong>TLx</strong></td>
<td>13806/788160</td>
<td>8463/394080</td>
<td>2156/197280</td>
<td>0/720</td>
</tr>
<tr>
<td></td>
<td>(1.75%)</td>
<td>(2.14%)</td>
<td>(1.09%)</td>
<td>(0%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Total kLUTs</th>
<th>Fabric Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VU3P</strong></td>
<td>394</td>
<td>8.1%</td>
</tr>
<tr>
<td><strong>KU15P</strong></td>
<td>523</td>
<td>6.1%</td>
</tr>
<tr>
<td><strong>VU9P</strong></td>
<td>1182</td>
<td>2.7%</td>
</tr>
</tbody>
</table>

OpenCAPI IP Floorplan on a VU3P
## CAPI and OpenCAPI Performance

<table>
<thead>
<tr>
<th></th>
<th>CAPI 1.0 PCIE Gen3 x8 Measured Bandwidth @8Gb/s</th>
<th>CAPI 2.0 PCIE Gen4 x8 Measured Bandwidth @16Gb/s</th>
<th>OpenCAPI 3.0 25 Gb/s x8 Measured Bandwidth @25Gb/s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>128B DMA Read</strong></td>
<td>3.81 GB/s</td>
<td>12.57 GB/s</td>
<td>22.1 GB/s</td>
</tr>
<tr>
<td><strong>128B DMA Write</strong></td>
<td>4.16 GB/s</td>
<td>11.85 GB/s</td>
<td>21.6 GB/s</td>
</tr>
<tr>
<td><strong>256B DMA Read</strong></td>
<td>N/A</td>
<td>13.94 GB/s</td>
<td>22.1 GB/s</td>
</tr>
<tr>
<td><strong>256B DMA Write</strong></td>
<td>N/A</td>
<td>14.04 GB/s</td>
<td>22.0 GB/s</td>
</tr>
</tbody>
</table>

**POWER8 CAPI 1.0**

**POWER9 CAPI 2.0**

**POWER9 Open Architecture with a Clean Slate Focused on Bandwidth and Latency**

*POWER8 Introduction in 2013*
Latency Ping-Pong Test

- Simple workload created to simulate communication between system and attached FPGA

- Bus traffic recorded with protocol analyzer and PowerBus traces

- Response times and statistics calculated

**Host Code**
1. Copy 512B from cache to FPGA
2. Poll on incoming 128B cache injection
3. Reset poll location
4. Repeat

**FPGA Code**
1. Poll on 512B received from host
2. Reset poll location
3. DMA write 128B for cache injection
4. Repeat

*HIP refers to hardened IP*
Latency Test Results

OpenCAPI Link

P9 OpenCAPI
3.9GHz Core, 3.4GHz Nest

298ns‡
2ns Jitter

Total Latency

Xilinx FPGA VU3P

TLx, DLx, PHYx (80ns‖)

PCIe G4 Link

P9 PCIe Gen4

est. <555ns§ Total Latency

Xilinx PCIe HIP (218ns¶)

Total Latency

Kaby Lake PCIe Gen3*
3.9GHz Core, 2.4GHz Nest

776ns§ Total Latency

Intel Core i7 7700 Quad-Core 3.6GHz (4.2GHz Turbo Boost)

Altera FPGA Stratix V

376ns
31ns Jitter

PCIe G3 Link

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen3
3.9GHz Core, 2.4GHz Nest

737ns§ Total Latency

337ns
7ns Jitter

PCIe G3 Link

Altera FPGA Stratix V

"PCIe Stack"

P9 PCIe Gen4

est. <337ns

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen4

Xilinx FPGA VU3P

378ns† Total Latency

376ns

"PCIe Stack"

Xilinx PCIe HIP (218ns¶)

"PCIe Stack"

Xilinx FPGA VU3P

337ns

7ns Jitter

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

Kaby Lake PCIe Gen3*
3.9GHz Core, 2.4GHz Nest

376ns
31ns Jitter

PCIe G3 Link

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen3
3.9GHz Core, 2.4GHz Nest

737ns§ Total Latency

337ns
7ns Jitter

PCIe G3 Link

Altera FPGA Stratix V

"PCIe Stack"

P9 PCIe Gen4

est. <337ns

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen4

Xilinx FPGA VU3P

378ns† Total Latency

376ns

"PCIe Stack"

Xilinx PCIe HIP (218ns¶)

"PCIe Stack"

Xilinx FPGA VU3P

337ns

7ns Jitter

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

Kaby Lake PCIe Gen3*
3.9GHz Core, 2.4GHz Nest

376ns
31ns Jitter

PCIe G3 Link

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen3
3.9GHz Core, 2.4GHz Nest

737ns§ Total Latency

337ns
7ns Jitter

PCIe G3 Link

Altera FPGA Stratix V

"PCIe Stack"

P9 PCIe Gen4

est. <337ns

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen4

Xilinx FPGA VU3P

378ns† Total Latency

376ns

"PCIe Stack"

Xilinx PCIe HIP (218ns¶)

"PCIe Stack"

Xilinx FPGA VU3P

337ns

7ns Jitter

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

Kaby Lake PCIe Gen3*
3.9GHz Core, 2.4GHz Nest

376ns
31ns Jitter

PCIe G3 Link

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen3
3.9GHz Core, 2.4GHz Nest

737ns§ Total Latency

337ns
7ns Jitter

PCIe G3 Link

Altera FPGA Stratix V

"PCIe Stack"

P9 PCIe Gen4

est. <337ns

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen4

Xilinx FPGA VU3P

378ns† Total Latency

376ns

"PCIe Stack"

Xilinx PCIe HIP (218ns¶)

"PCIe Stack"

Xilinx FPGA VU3P

337ns

7ns Jitter

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

Kaby Lake PCIe Gen3*
3.9GHz Core, 2.4GHz Nest

376ns
31ns Jitter

PCIe G3 Link

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen3
3.9GHz Core, 2.4GHz Nest

737ns§ Total Latency

337ns
7ns Jitter

PCIe G3 Link

Altera FPGA Stratix V

"PCIe Stack"

P9 PCIe Gen4

est. <337ns

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

P9 PCIe Gen4

Xilinx FPGA VU3P

378ns† Total Latency

376ns

"PCIe Stack"

Xilinx PCIe HIP (218ns¶)

"PCIe Stack"

Xilinx FPGA VU3P

337ns

7ns Jitter

"PCIe Stack"

Altera PCIe HIP (400ns§)

"PCIe Stack"

Kaby Lake PCIe Gen3*
3.9GHz Core, 2.4GHz Nest

376ns
31ns Jitter

PCIe G3 Link

Altera PCIe HIP (400ns§)
Nallatech / Molex ASG – CAPI Flash Acceleration Product Timeline

CAPI 1.0 Bridge to IBM’s Flash Drawer Power8 PCIe Gen 3 Storage Acceleration

Altera 5SGXA7 FPGA to Fiber Channel Interface
Introduced 2014

CAPI 1.0 FlashGT Power8 PCIe Gen 3 Storage Acceleration

Xilinx KU060 FPGA + 2x 1TByte M.2 NVMe SSDs
Introduced 2016

CAPI 2.0 FlashGT+ Power9 PCIe Gen 4 Storage Acceleration

Xilinx KU15P FPGA + 4x 1TByte M.2 NVMe SSDs Or 4x cabled U.2 NVMe SSDs
Introduced 2017

OpenCAPI Hyperconverged & Disaggregatable Flash Storage Accelerator for Zaius/Barreleye-G2 OCP Power9 platforms

OpenCAPI, CAPI 2.0, PCIe 250-SoC Generic Storage Acceleration

Xilinx ZU19P MPSoC FPGA + 8x 2TByte M.2 NVMe SSDs + 50GB/s Dataplane Fabric IO
Introduced 2018

Xilinx ZU19P MPSoC FPGA + 4x PCIe G4x8 cabled Storage IO 50GB/s Dataplane Fabric IO
Introduced 2018

CAPI 2.0 Bridge to IBM’s Flash Drawer Power8 PCIe Gen 3 Storage Acceleration

CAPI 1.0 Bridge to IBM’s Flash Drawer Power8 PCIe Gen 3 Storage Acceleration

Altera 5SGXA7 FPGA to Fiber Channel Interface
Introduced 2014

Xilinx KU060 FPGA + 2x 1TByte M.2 NVMe SSDs
Introduced 2016

Xilinx KU15P FPGA + 4x 1TByte M.2 NVMe SSDs Or 4x cabled U.2 NVMe SSDs
Introduced 2017

Xilinx ZU19P MPSoC FPGA + 8x 2TByte M.2 NVMe SSDs + 50GB/s Dataplane Fabric IO
Introduced 2018

Xilinx ZU19P MPSoC FPGA + 4x PCIe G4x8 cabled Storage IO 50GB/s Dataplane Fabric IO
Introduced 2018
Leveraging OpenCAPI as a Bridge to a Data Centric World
Data Centric Architectures - Fundamental Principles

1. Consume Zero Power when Data is Idle
2. Don’t Move the Data unless you absolutely have to
3. When Data has to Move, Move it as efficiently as possible

Which Translates to :-

1. Use Non Volatile Memory where possible
2. Move the compute to the data
3. Leverage independent power efficient Dataplanes
Data Center Architectures, blending evolutionary with revolutionary

Zero Power, High Bandwidth, Low Latency, Non Coherent, Streaming Data Plane, e.g GenZ

Emerging Data Centric Enhancements

Existing Data Center Infrastructure

Traditional ToR Packet switched Network & Control Plane
» Leverage Google & Rackspace’s OCP Zaius/Barreleye G2 platform
» Reconfigurable FPGA Fabric with Balanced Bandwidth to CPU, Storage & Data Plane Network
» OpenCAPI provides Low Latency & coherent Accelerator / Processor Interface
» GenZ Memory-Semantic Fabric provides Addressable shared memory up to 32 Zetabytes
Molex ASG Flash Storage Accelerator, FSA, in Barreleye-G2 OCP Server

» Xilinx Zynq US+ 0.5OU High Storage Accelerator Blade
» 4 FSAs in 2OU Barreleye-G2 OCP Storage drawer deliver:
  » 152 GByte/s PFD* Bandwidth to 1TB of DDR4 Memory
  » 256 GByte/s PFD* Bandwidth to 64TB of Flash
  » 200 GByte/s PFD* Bandwidth through the OpenCAPI channels
  » 200 GByte/s PFD* Bandwidth through the GenZ Fabric IO
» Open Architecture software/firmware framework

*PFD = Peak Full Duplex

OpenCAPI Interface

Data Plane I/O

PCIe Gen 3 Switch

Zynq US+ ZU19EG FFVC1760

MPSoC

PCIe x16 G3

8x PCIe x4 G3

M.2 22110 SSD

100GbE QSFP28

8GByte DDR4

M.2 22110 SSD

Xilinx Zynq US+ 0.5OU High Storage Accelerator Blade

Molex ASG Flash Storage Accelerator, FSA, in Barreleye-G2 OCP Server

128GByte DDR4 RDIMM

PCIe G2 x 4

Control Plane Interface

Nallatech a moxel company
Summary

- The OpenCAPI interface standard is a perfect compliment to the OCP Initiative bringing best in class features including:
  - Coherency
  - Lowest Latency
  - Highest Bandwidth
  - Open Standard
  - Perfect Bridge to blend CPU Centric & Data Centric Architectures

- Simultaneous Hyperconverged & Dissagregatable Flash Memory solutions can be built without performance compromise

- OCP, OpenCAPI & FPGA Acceleration are now bringing highly optimized Data Centric server architectures closer to reality
JOIN TODAY!

www.opencapi.org

Come see us in the Expo Hall

OpenCAPI Booth C5