Industry Collaboration and Innovation

OpenCAPI

A data-centric approach to server design
OpenCAPI Topics

- Industry Background
- Technology Overview
- Design Enablement
- OpenCAPI Consortium
Industry Landscape

- Key changes occurring in our industry
  - Historical microprocessor technology continues to deliver far less than the historical rate of cost/performance improvement per generation – Running out of steam
  - New advanced memory technologies changing the economics of computing

- Companies realizing need for **accelerated computing** with a coherent **high performance bus** to meet today’s computational demand
Attributes driving Accelerators

- Introduction of device **coherency requirements** (IBM’s introduction in 2013)
- Emergence of **complex storage and memory solutions**
- Growing demand for **network performance**
- Various form factors (e.g., GPUs, FPGAs, ASICs, etc.)

Driving factors for a high performance bus - Consider the environment

- Increased industry dependence on hardware acceleration for performance
- **Hyperscale datacenters and HPC** are driving need for much higher network bandwidth
- **Deep learning** and HPC require more bandwidth between accelerators and memory
- New **memory/storage technologies** are increasing the need for bandwidth with low latency
Two Bus Challenges

1. Coherent high performance bus needed
   • Hardware acceleration will become commonplace
   • But, if you are going to use Accelerators, you need to get data in/out very quickly
   • Today’s system interfaces are insufficient to address this requirement
     • Traditional I/O architecture results in very high CPU overhead when applications communicate with I/O or Accelerator devices
     • Systems must be able to integrate multiple memory technologies with different access methods, coherency and performance attributes

2. These challenges must be addressed in an open architecture allowing full industry participation
   • Need to be architecture agnostic to enable the ecosystem growth and adaption
   • Establish sufficient volume base to drive cost down
   • Support broad ecosystem of software and attached devices
OpenCAPI Approach

- Two bus challenges
  1. Coherent high performance bus
     → Needed to define a new technology
  2. A need to make this ‘open’
     → Needed to establish open community

- Approach taken to define this new technology
  - Bottom’s up design approach for advanced capabilities and performance
    - Clean sheet of paper with no incumbent overhead
    - Architecture and electrical view point

- **OpenCAPI** bus architecture was Born
OpenCAPI Topics

- Industry Background
- Technology Overview
- Design Enablement
- OpenCAPI Consortium
Use Cases – A truly heterogeneous architecture built upon OpenCAPI

1. Accelerators: The performance, virtual addressing and coherence capabilities allow FPGA and ASIC accelerators to behave as if they were integrated into a custom microprocessor.

2. Coherent Network Controller: OpenCAPI provides the bandwidth that will be needed to support rapidly increasing network speeds. Network controllers based on virtual addressing and eliminate software overhead without the programming complexity usually associated with user-level networking protocols.

3. Advanced Memory: OpenCAPI allows system designers to take full advantage of emerging memory technologies to change the economics of the datacenter.

4. Coherent Storage Controller: OpenCAPI allows storage controllers to bypass kernel software overhead, enabling extreme IOPS performance without wasting valuable CPU cycles.

OpenCAPI specifications are downloadable from the website at [www.opencapi.org](http://www.opencapi.org)
- Register
- Download
**OpenCAPI Key Attributes**

1. Architecture agnostic bus – Applicable with any system/microprocessor architecture
2. Optimized for High Bandwidth and Low Latency
3. High performance 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. Supports a wide range of use cases and access semantics
7. CPU coherent device memory (Home Agent Memory)
8. Advanced Memories and Classic Memory

---

**Storage/Compute/Network etc**
**ASIC/FPGA/FFSA**
**FPGA, SOC, GPU Accelerator**
**Load/Store or Block Access**
Virtual Addressing and Benefits

- An OpenCAPI device operates in the virtual address spaces of the applications that it supports
  - Eliminates kernel and device driver software overhead
  - Allows device to operate on application memory without kernel-level data copies/pinned pages
  - Simplifies programming effort to integrate accelerators into applications
  - Improves accelerator performance

- The Virtual-to-Physical Address Translation occurs in the host CPU
  - Reduces design complexity of OpenCAPI-attached devices
  - Makes it easier to ensure interoperability between OpenCAPI devices and different CPU architectures
  - Security - Since the OpenCAPI device never has access to a physical address, this eliminates the possibility of a defective or malicious device accessing memory locations belonging to the kernel or other applications that it is not authorized to access
OpenCAPI Advantages for Memory

- Open standard interface enables to attach wide range of devices
- OpenCAPI protocol was architected to minimize latency
  - Especially advantageous for classic DRAM memory
- Extreme bandwidth beyond classical DDR memory interface
- Agnostic interface allows extension to evolving memory technologies in the future (e.g., compute-in-memory)
- Ability to handle a memory buffer to decouple raw memory and host interfaces to optimize power, cost and performance
- Common physical interface between non-memory and memory devices
Asymmetric Demands of the Device and Host

- **ISA of Host Architecture**: Need to hide difference in Coherence, Memory Model, Address Translation, etc.

- **Serdes frequency ratio**: As device frequency is reduced on the attached device, the ratio between the interface and internal clock is different.

- **Design schedule**: The design schedule of a high performance CPU host is typically on the order of multiple years, conversely, accelerator devices have much shorter development cycles, typically less than a year.

- **Timing Corner**: Related to the design schedule, accelerator designs in both ASIC and FPGA technologies run at lower frequencies and timing optimization as CPUs.

- **Plurality of devices**: Attached devices have an inherently larger number of device implementations compared to hosts. Effort in the host, both IP and circuit resource, have a multiplicative.

- **Trust**: Attached devices are susceptible to both intentional and unintentional trust violations. The accelerated design schedule of accelerators makes validation difficult.

- **Cache coherence**: Hosts have high variability in protocol. Host cannot trust attached device to obey rules.
Comparison of Memory Paradigms

Emerging Storage Class Memory
- Processor Chip
- DL/TLx
- SCM

Tiered Memory
- Processor Chip
- DL/TLx
- DDR4/5
- SCM

Main Memory
- Processor Chip
- DL/TLx
- DDR4/5

Example: Basic DDR attach
OpenCAPI Attached Memory

- Tiered emerging memory technologies can now provide a hybrid memory solution that uses economies of scale
  - Defined in the OpenCAPI 3.0 Transaction Layer Specification
  - Also memory buffer chips can leverage OpenCAPI talking to DDR4/5, GDDR, etc.
    - Defined in the Transaction Layer OpenCAPI 3.1 specification

- Both the Standard and Advanced Memory is coherent with the processor’s caches
- Hybrid memory is also part of coherent domain
- Application uses same Load/Store Model to address this memory just like System Memory
- Economy of Scale Memory Footprint. Buy less system memory and instead use advanced memory technologies
OpenCAPI Features and Programming Ease

Higher performance programming model
1. Virtual addressing enables low overhead with no Kernel, hypervisor or firmware involvement
2. CPU <-> Accelerator
   - Low latency event notification
   - OpenCAPI removes PCIe layering
   - Fast CPU → Accelerator load/store
3. CPU coherent device memory
   - Shared coherent data structures and pointers
4. Traditional thread level programming
5. Atomic operations for barriers, locks, semaphores, etc.
**Acceleration Paradigms with Great Performance**

- **Memory Transform**
  - Example: Basic work offload
  - Processor Chip \(\rightarrow\) Data \(\rightarrow\) Acc
  - Processor Chip \(\leftarrow\) Data \(\leftarrow\) Acc

  - Examples: Machine or Deep Learning potentially using OpenCAPI attached memory

- **Egress Transform**
  - Processor Chip \(\rightarrow\) Data \(\rightarrow\) Acc
  - DLx/TLx

  - Examples: Encryption, Compression, Erasure prior to network or storage

- **Needle-In-A-Haystack Engine**
  - Processor Chip \(\rightarrow\) Data \(\rightarrow\) Acc
  - DLx/TLx

  - Examples: Database searches, joins, intersections, merges

- **Ingress Transform**
  - Processor Chip \(\leftarrow\) Data \(\leftarrow\) Acc
  - DLx/TLx

  - Examples: Video Analytics, HFT, VPN/IPsec/SSL, Deep Packet Inspection (DPI), Data Plane Accelerator (DPA), Video Encoding (H.265), etc

- **Bi-Directional Transform**
  - Processor Chip \(\leftrightarrow\) Data \(\leftrightarrow\) Acc

  - Examples: NoSQL such as Neo4J with Graph Node Traversals, etc

---

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

- Industry Background
- Technology Overview
- Design Enablement
- OpenCAPI Consortium
Transaction Layer (TL) specifies the control and response packets between a host and an endpoint OpenCAPI device.

**TL On the host side converts:**
- Host specific protocol requests into transaction layer defined commands
- TLx commands into host specific protocol requests.
- Responses

**Data Link layer supports a 25Gbps serial data rate per lane connecting a processor to an accelerator device:** DL and DLX
- Configuration supports lanes running at 25.78125 GHz for a 25 GB/s data rate

**TLx:** On the endpoint OpenCAPI device, the transaction layer converts:
- AFU protocol requests into transaction layer commands
- TL commands into AFU protocol requests.
- Responses

---

**The full TL/DL specification can be downloaded at opencapi.org and registering under the technical -> specifications pull down menu.**
FPGA and ASIC/Structured Array Reference Designs

Reference Designs

- The OpenCAPI architecture is truly agnostic to any vendor technology.
- The TLx and DLx reference RTL is written for the Xilinx FPGA Vivado toolchain and statistics are provided using that flow in this deck (note: BlockRAM (BRAM), Distributed RAM are specific Xilinx constructs).
- To convert to an ASIC vendor or a structured array technology would be a very minimal exercise to port.
  - Discussions held to take our reference design RTL and harden it into a structured array technology.
  - Master Definition is underway between IBM and a structured array provider.
- Economy of scale, performance and NRE are considerations that partners need to make in deciding whether to go the FPGA, ASIC, or structured array route to market.
Exerciser Examples – Provided to OCC Members

- **MemCopy**
  - The MemCopy example is a data mover from source address -> destination address using Virtual Addressing and includes these features
    - Configuration and MMIO Register Space
    - acTag Table used for Bus/Device/Function and Process ID identification
    - 512 processes/contexts and 32 engines supporting up to 2K transfers using 64B, 128B, or 256B operations

- **Memory Home Agent**
  - The Memory Home Agent example implements memory off the endpoint OpenCAPI accelerator to act as a coherent extension to the host processor memory
  - The Memory Home Agent example includes these features
    - Configuration and MMIO Register Space
    - Individual and pipelined operation for memory loads and stores
    - Interrupts, with error details reported to software through MMIO registers
    - Sparse Address Mapping feature to extend 1 MB of real space to 4 TB of address

- Potentially other exercisers to be defined and contributed later
Validation to Date

• POWER9 is in the lab
• 25Gbit/sec signal integrity work completed
• Memcopy, Home Agent Memory and AFP exercisers are all up and running cleanly using the Alpha Data 9V3 card with Xilinx VU3P FPGA and the Mellanox Innova2 Programmable Adapter Card
• Bandwidth measurements of ~22GB/s achieved with streaming read and streaming writes
• Latency measurements can be obtained under NDA with partner
OpenCAPI Topics

- Industry Background
- Technology Overview
- Design Enablement
- OpenCAPI Consortium
Cross Industry Collaboration and Innovation

Welcoming new members in all areas of the ecosystem
## Current Members

### Board Membership level
- AMD
- Google
- IBM
- Mellanox Technologies
- Micron
- NVIDIA
- WD
- Xilinx

### Contributor Membership level
- Amphenol AssembleTech
- molex
- Parade
- SK hynix
- TE Connectivity
- Microsemi

### Observing Membership Level
- DELL EMC
- achronix Semiconductors
- Applied Materials
- Synology
- Rambus
- Roche

### Academic Membership Level
- ELI beamlines
- NGCodec
- SmartDV
- Universidad de Córdoba
JOIN TODAY!

www.opencapi.org