CCIX: Interconnect for Seamless Acceleration

Presented By

Name: Millind Mittal
Title: Sr. Director of Architecture
Date: Oct 1, 18
Key Drivers for Interconnect Technology

- Decline of Moore’s law forcing more heterogeneous compute
- General Purpose processors are not power efficient or cost effective for a class of workloads
- 5G wireless applications requiring 10x more bandwidth, 10x lower latency by 2021
- Increase in distributed data forcing more network intelligence at faster data rates (10GbE -> 100GbE -> 400GbE)
- New classes of applications require exponential growth in computation needs
  - Requires moving beyond only General Purpose Processor based processing
  - Key enabler for wide adaption of acceleration technologies is high performance interconnect with seamless data sharing model
CCIX Consortium Effort

- Advance IO Interconnect to enable seamless expansion of compute and memory resources beyond processor SoCs
  - Accelerator SoCs to be like a NUMA node from Data Sharing perspective.
CCIX Ecosystem Status Update

> CCIX Base Spec 1.0 available.

> Complete Ecosystem with CCIX-enabled Host, Accelerator devices and SCM memory expansion products rolling out

> CCIX Hosts:
  >> ARM/Cadence/Xilinx collaboration – A 7nm test Processor SoC providing CCIX interface
  >> Other Hosts with caching and Slave Agent / Memory Expansion support coming soon

> CCIX Accelerator / EP:
  >> Xilinx VU3xP family CCIX-enabled FPGAs silicon available

> CCIX Memory Expansion
  >> Leading SCM memory vendor driving CCIX Slave Agent / Memory Expansion use case
CCIX Roadmap and Milestones Update

> **CCIX 1.1**
  > Support 32GT/s, can use PCIe Gen5 switch for fan-out and other CCIX topologies.
  > Protocol enhancements to increase performance and reduce latency further
  > Target 4Q2018 to 1Q2019

> **CCIX 2.0**
  > Expands Seamless coherent data sharing and load/store access to across Multiple Nodes
  > Support 56GT/s and higher
  > Target End’2019
Use Case 1: Virtualized, Coherent Accelerators

- Reduced data transfer latency
- Improved fine grain data sharing
- Simplified software dev., eliminates difficult debug issues
- Seamless offload of threads from general-purpose processors to accelerators
  - Preserves shared data-structures between the host and accelerator
  - No need to re-architect any shared data structures

Improved efficiency with true peer-processing
Use Case 2: Memory Expansion

- Multiple use cases evolving for external interconnect attached memory
  - Larger DRAM/SCM capacity with-in a “box”
  - LD/ST to remote memory via bridging to a scale-out fabric
    - Opportunity for value-add functionality via external card solutions for remote memory
    - Overtime there is need for choices in the scale-out fabric for carrying native LD/ST

- Supports Memory Atomics over CCIX interface
Hardware Architecture
CCIX Layered Architecture

Protocol Layer
- Coherency protocol, memory read & write flows
- Full feature protocol
- Port aggregation for higher BW

Link Layer
- Formats CCIX messages for target transport
- Adds ability to pack and chain multiple messages to achieve higher efficiency

Transaction Layer
- Adds optimized packets, manages credit based flow control

Physical Layer
- Dual mode PHY to support extended data rates
CCIX coherency layer architecture model

- Portable protocol to other transports
- Support for port aggregation, multiple link agents
- CCIX agent types:
  - Request Agent (RA) - single (implementation specific) function or proxy for multiple functions
  - Home Agent (HA) - point of coherency for a given address
  - Slave Agent (SA) - used for memory expansion
  - Error Agent (EA) – receives and processes protocol error messages
System Topology Examples

Direct attached, daisy chain, mesh and switched topologies
Software Architecture
Towards a True Driverless Model

> Driver or OS involvement in Data Movement adds latency and processing overhead
  >> Move to driverless model for data movement

> Traditional DMA approach is to provide a special kernel driver for every unique accelerator
  >> Requires skilled kernel developers (a driver for each accelerator), failure mode is catastrophic (system crash/downtime)

> CCIX capable devices behave similarly to nodes in existing NUMA systems
  >> Memory based approach leverages existing Operating System capabilities
  >> Enabled by coherent shared virtual memory – it's all “just memory”

> OS enablement required, mostly limited to kernel infrastructure
  >> e.g. OS driver for power management, error handling, etc.
  >> Lightweight OS impact for individual accelerator drivers
Management

> Runs management interface over standard PCIe interface
> Leverage PCIe support for address translation service
  >> CCIX adds extension to carry additional memory attributes and to address translation invalidation performance issue
> Leverages PCIe signaling mechanism
CCIX Consortium SW activities

> CCIX capability discovery and configuration

> UEFI updates to support Peripheral attached memory and heterogeneous NUMA platforms

> ACPI extensions

> CCIX common management driver

> Management flows – Hot plug, Power management

> RAS – error reporting and handling – its integration into Kernel level

> Kernel enhancements
  > Memory management
  > Long term – scheduling accelerator resource
Use Case Demos
KVS Seamless Acceleration

> CCIX Value Proposition

  >> Leave ‘Control’ operations (set, delete, ..) on the host- offload “fast-path” operations (Get) to the accelerator
  >> Leverage contention data-structure as-it-is between host and accelerator
    - High through-put Independent of the size of the request
    - In presence of longer latency memory pool (e.g. use of SCM or DRAM expansion through peripheral device.
    - Tolerant to Address Translation latency overhead in presence of TLB misses
    - Future work- support for memory expansion
      - Hash table with the accelerator (host can share same table)
      - Zero-overhead Zero-copy Tx
      - Seamless offload of Linux networking (self hosted NIC with fast-path termination)
  - Total throughput benefit in the range of 2x-10x

> Demo Performance Data

  >> Measured reduction in CPU utilization for application processing due to all of Get-op offload – 75% CPU reduction
  >> Increase throughput for multi-gets with almost increase in CPU utilization
    - Show-cased 2x increase in throughput without any increase in CPU utilization for application layer threads when number of Get-ops is increased from 1x to 4x
Successful Hardware Demos

> ISC 18 - Seamless Acceleration of KVS
> SC17 – Accelerated OVS

CCIX 25G Demo

Future 56G Demo
Xilinx Devices with CCIX Support

16nm Ultrascale + (4th Gen) ES Sample: May, 2018

- 4th Gen 3D IC
  - 3 16nm FPGA die
  - 2 HBM2 Stacks
- 4 PCIe-Gen3x16 controllers with CCIX transport support; Each also works as PCIe-Gen4x8
- Cache is implemented in Soft IP
  - upto 4MB of cache; upto 8 accelerator functions.
  - IP for Slave Agent (memory expansion) support

7nm (5th Gen) ES

- Hardened Coherency Blocks
Summary

> CCIX enables broader use of acceleration technologies
> CCIX Base specification is available
> CCIX is supported by broad eco-system- both host and accelerator devices in under development with ES becoming available in near future
> Active work underway to enable SW eco-system and showcase use cases
> Go to www.ccixconsortium.com for learn more about CCIX and to join CCIX eco-system.