SEAMLESS ACCELERATION

An Introduction to CCIX™

WHITE PAPER

SYNOPSIS

Cache Coherent Interconnect for Accelerators or CCIX™ (pronounced 'see 6') is a chip–to-chip interconnect that enables two or more devices to share data in a cache coherent manner. Machine Learning and Big Data applications are fundamentally changing the way that the processing of data happens. Classic processor data flows are now being augmented with off-chip accelerators that can be customized for specific types of applications from compute accelerators to network traffic acceleration. This has driven an industry wide movement towards accelerators and heterogeneous compute. For many of today's compute tasks, accelerators can complete the needed functionality both faster and with lower power consumption than the processor working on its own. However, unmanaged heterogeneity can bring software complexity. CCIX is poised to optimize and simplify how heterogeneous systems are architected while at the same time increasing bandwidth and reducing latency in the systems built with devices processing via processors with different instruction set architectures (ISAs) or application specific accelerators. This white paper explains the key features of the CCIX standard and why it is set for fast adoption and long lasting support.

As Moore's law slows, the industry as a whole needs to find ways of increasing compute performance, while retaining low power consumption.

THE ACCELERATION CHALLENGE

Today, the high tech landscape is shaped by the innovations of new, large-scale consumer services, 5G, Cloud Computing, Internet of Things, Big Data and Autonomous Driving. Machine Learning and Artificial Intelligence applications are fundamentally changing consumer behavior. This has in turn led to evolving platform and solution architectures to address these new applications in an efficient and scalable fashion. The performance required by these applications cannot be reached with solutions based purely on CPU-centric server architectures. Instead, solutions based on highly efficient heterogeneous computing architecture are needed with the addition of accelerators such as Graphic Processing Units (GPUs), Field-Programmable Gate Arrays (FPGAs), Smart Network Interface Cards (Smart NICs) and many other domain-specific programmable devices. PCI Express™ (PCIe™) is currently the most common protocol for moving data between the processor and off-chip accelerators. While the PCIe protocol works well as an input output (IO) protocol, it does not enable IO devices to be elements within a seamless peer-processing model.

As an ever increasing number of applications for off-chip accelerators come on line, there are three overarching drivers for what a next-generation interconnect needs to achieve: high performance, low latency and ease of use.

CCIX BENEFITS

A high performance, low latency, chip-to-chip interface is a critical part of any system that relies on off-chip accelerators. In order for the industry to be successful, it is important that a non-proprietary standard is available to vendors of processors, accelerators and other peripheral devices to ensure seamless chip-to-chip communication. The strong demand for such a standard led to the formation of the CCIX Consortium, a group of like-minded companies looking to solve the interconnect needs of off-chip accelerators. By changing the paradigm of how memory and data movement is thought of in the accelerator, the new interconnect can achieve a critical role in enabling next-generation platforms and solutions.

CCIX uses two mechanisms to increase performance and reduce latency. The first mechanism is to use cache coherency to automatically keep the processor and accelerator caches coherent thus facilitating ease-of-use and reducing latency. The second mechanism is for CCIX to increase the raw bandwidth of the link. This is done by increasing the maximum link speed to 25GT/s. The CCIX Specification also defines how multiple CCIX Ports can be aggregated together to enable interfaces with performance beyond that of a single interface allowing the accelerator and memory expansion bandwidth to be matched.

CACHE COHERENCY AND SHARED VIRTUAL MEMORY

Multi-processor systems have already implemented the technology to ensure caches between different processors are kept up to date.

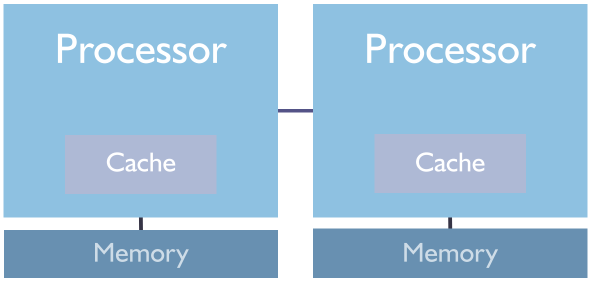

FIGURE 1 | *Multi-processor cache coherency*

By extending the basic premise of existing cache coherent interconnects to accelerators, application data can be autonomously moved between processor cache and accelerator cache without a software driver being involved in the data movement. In addition to cache memory, CCIX also enables expansion of the OS paged memory (system memory) to include CCIX device attached memory as well. The CCIX data sharing model is based on use of shared memory addressed with Virtual Addresses (VAs).

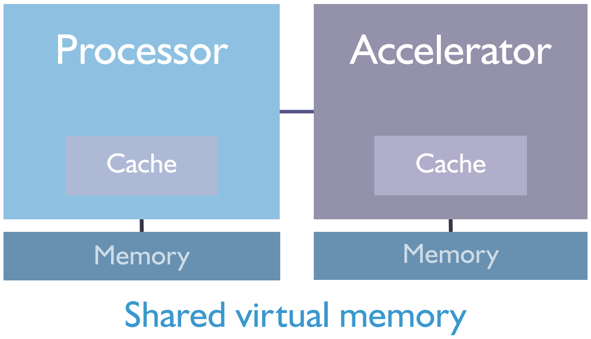

FIGURE 2 | *Share Virtual Memory with cache coherency*

The cache and/or memory between the processor and accelerator is automatically kept up to date using the CCIX protocol. Because the data is automatically synchronized by the CCIX protocol, only data pointers need to be passed rather than relying on complex direct memory access (DMA) drivers. This automatic synchronization leads to reduced data latency and improved application performance. It also simplifies the burden on the software developer allowing them to focus on their application rather than the underlying mechanics of moving data between the accelerator and the host processor.

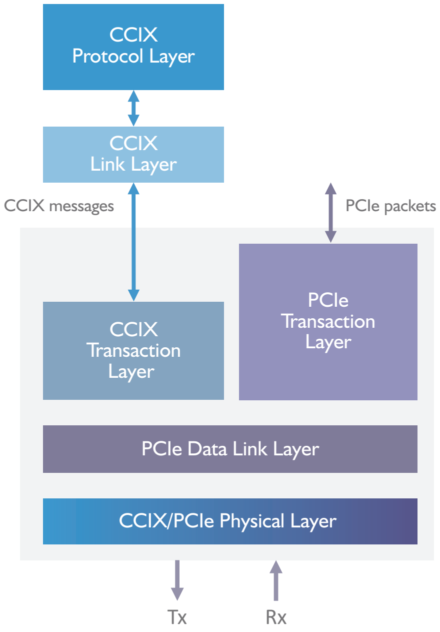

FIGURE 3 | *CCIX Layered Architecture*

CCIX LAYERED ARCHITECTURE

The CCIX architecture is a layer based architecture that expands on the base PCI Express architecture. Generally speaking, CCIX can be thought of as two main specifications that further break down into protocol layers. The CCIX Protocol Specification covers the CCIX Protocol and CCIX Link Layers. These layers define the cache coherent protocol, messaging and flow control and CCIX transport portions of the protocol.

The CCIX Transport Specification, includes the CCIX and PCIe Transaction Layers, the PCIe Data Link Layer and the CCIX Physical Layer. These layers generally take care of the physical link between devices including speed and width negotiation, packet error checking and retry, as well as the initial packet decode protocol.

CCIX Protocol Layer

At the very top of the CCIX stack resides the CCIX Protocol Layer. This layer is responsible for the coherency protocol, including memory read and write flows. The layer provides a simple mapping for on-chip coherency protocols such as the Arm AMBA CHI. The cache states defined in this layer allow hardware to determine the state of the memory. For instance, hardware can determine if the data is unique and clean or if it is shared and dirty.

CCIX Link Layer

The layer below the CCIX Protocol Layer is the CCIX Link Layer. This layer is responsible for formatting CCIX traffic for the target transport. Currently that is PCIe, but with the layered architecture, CCIX could be mapped over a different transport layer in the future. In addition, this layer manages port aggregation, allowing multiple ports to be aggregated together to increase bandwidth.

CCIX and PCIe Transaction Layers

The CCIX and PCIe transaction layers are responsible for handling their respective packets. The PCIe protocol allows for the implementation of Virtual Channels allowing different data streams to travel across a single PCIe link. By splitting CCIX traffic into one Virtual Channel and PCIe traffic into a second Virtual Channel, both CCIX and PCIe traffic can share the same link.

CCIX has the ability to work with either standard PCIe packets or to use optimized CCIX packets. Optimized CCIX packets eliminate several unneeded fields that exist in PCIe packets. When using PCIe packets, existing PCIe switches can be used to transport packets. If optimized CCIX packets are enabled, these optimized packets eliminate the PCIe overhead resulting in a much smaller and more efficient packet for communicating coherency.

PCIe Data Link Layer

The PCIe Data Link Layer performs all of the normal functions of the data link layer. Some examples of these functions are CRC error checking, packet acknowledgment and timeout checking, and credit initialization and exchange.

CCIX/PCIe Physical Layer

The basis for the CCIX/PCIe Physical layer is the PCIe physical layer. Additionally, CCIX extends the Physical Layer to support 25GT/s. This faster speed, known as ESM (Extended Speed Mode), is automatically detected when two ESM capable devices are connected, resulting in a 56% increase over 16GT/s.

CCIX SYSTEM TOPOLOGY EXAMPLES

Because of the layered architecture, CCIX enables a number of flexible topologies. The most common topology will be direct attached shared virtual memory. But other topologies such as switches, daisy chains and meshes can be easily created and supported.

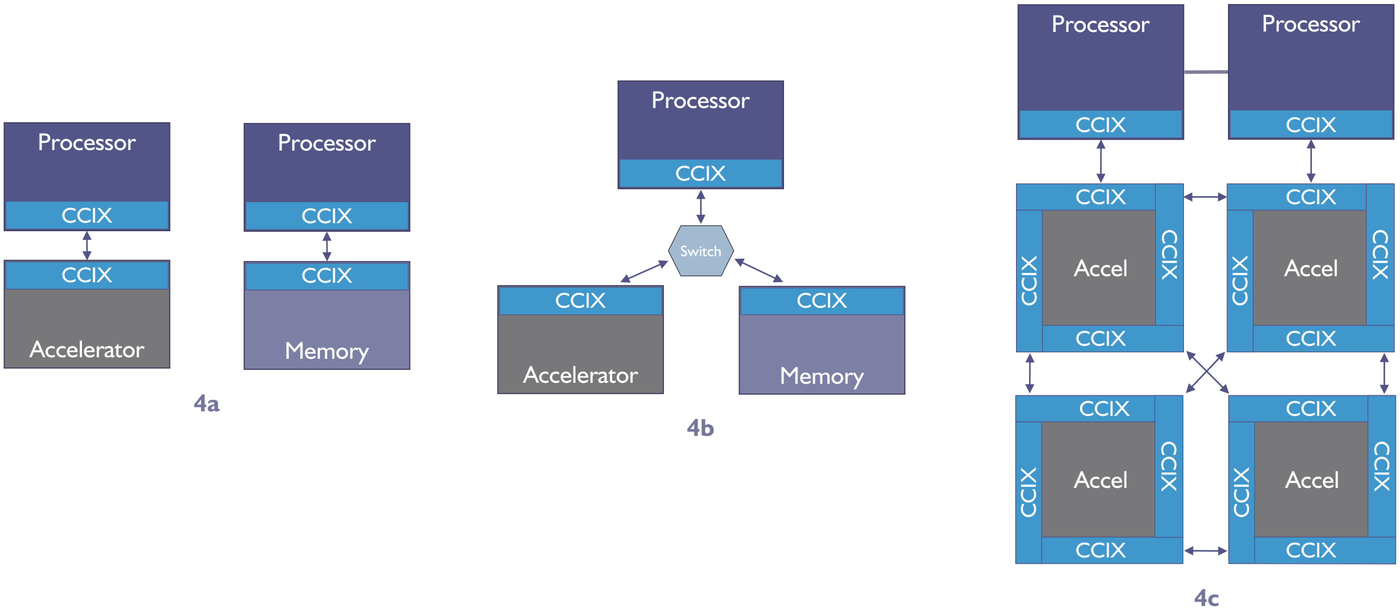

FIGURE 4 | *CCIX Example Topologies*
(4a-Direct Attached; 4b-Switched Topologies; 4c-Hybrid Daisy Chain)

CCIX COHERENCY LAYER ARCHITECTURE

The CCIX protocol defines the memory access protocol in terms of CCIX building block components. All CCIX devices will be composed of at least one port with a CCIX link. A CCIX Port is associated with a set of physical pins and is used to connect to another CCIX Port to exchange information between two or more distinct chips.

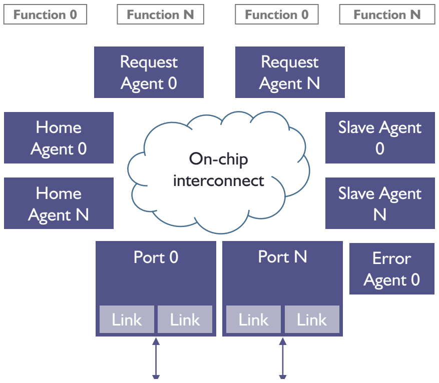

FIGURE 5
CCIX Coherency Layer Architecture Model

There are also a number of different agent types defined and which agents are on a device will depend on the functionality of that device. The agent types that are defined are: Request Agent, Home Agent, Slave Agent, and Error Agent. Request Agents, Home Agents, Slave Agents, Error Agents, Ports and Links in the system are collectively called CCIX components.

An Agent is identified within the protocol using an Agent ID value. A brief description of each agent type is given below.

Request Agent – A Request Agent performs read and write transactions to different addresses within the system. A Request Agent can cache locations that it has accessed. Each CCIX Request Agent may have one or more processing functions acting as internal initiators whose requests are then carried out from a CCIX architected Request Agent. Essentially a CCIX Request Agent provides an interface for accelerators, or CCIX enabled IO masters, to coherent system memory. In addition, the Request Agent enables caching in the accelerators to be coherent and therefore transparent to the programmer.

Home Agent - A Home Agent is responsible for managing coherency and access to memory for a pre-determined address range. It manages coherency by sending snoop transactions to the required Request Agents when a cache state change for a cache line is required.

Slave Agent – CCIX enables expanding system memory to include memory attached to peripheral or external devices. This scenario occurs when the Home Agent resides on one chip and some or all of the physical memory associated with the Home Agent resides on a separate chip. This architectural component (the expansion memory) is referred to as Slave Agent. A Slave Agent is never accessed directly by a Request Agent. A Request Agent always accesses a Home Agent, which in turn accesses the Slave Agent.

Error Agent – An Error Agent receives and processes protocol error messages. The protocol error messages are sent from CCIX components.

EXAMPLE CCIX DATA FLOWS

With the knowledge of agent types above, some common use cases expected for CCIX can now be explained.

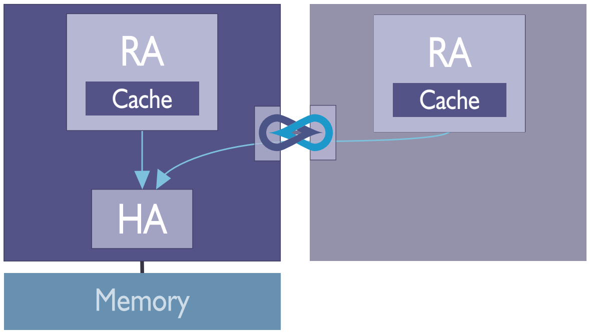

FIGURE 6 | *Accelerator sharing processor memory over CCIX*

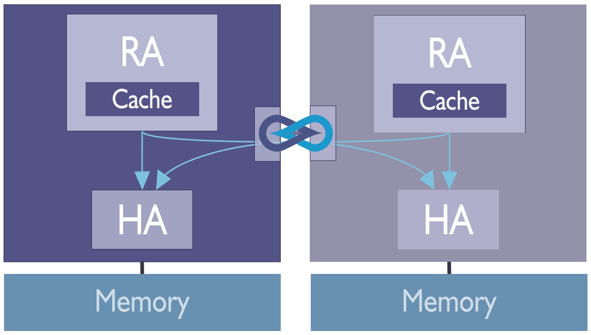

FIGURE 7 | *Shared memory between processor and accelerator*

Accelerator Shares Processor Memory

As CCIX is adopted and deployed, the most common initial use case will be for the processor and the accelerator to share host memory and expansion of host memory to include CCIX device attached memory. In this case, there will be two Request Agents, each managing their own caches. The Home Agent will reside in the processor and manages access to the memory that is connected to the processor.

Shared Processor and Accelerator Memory

The next most common model will likely be the case where the processor and accelerator can share virtual memory. In this case, the memory for both the accelerator and the processor are part of a shared virtual memory pool. The processor can simply pass address pointers to the accelerator indicating the data that needs to be worked on, rather than require a complex PCIe DMA and driver to move data between processor and accelerator memory. In this case, there are two Request Agents managing the respective caches and two Home Agents managing memory. By eliminating software driver development and overhead, system performance and software simplicity can be greatly improved.

Beyond the Basic Layout

Due to the very flexible nature of CCIX, it has the ability to work beyond the basic data flows shown above. From direct attached accelerators to mesh and star networks, CCIX has a very appealing set of options to enable a large variety of topologies.

CCIX SOFTWARE

While the hardware enhancements for CCIX will greatly benefit off-chip accelerators, one of the key benefits of the CCIX methodology is the ability to allow a driverless data-movement option for data sharing between hosts and accelerators. Traditional PCIe accelerators require a driver to be used for data movement to and from the accelerator that adds latency and processing overhead. With the driverless data-movement option, CCIX also allows for the expansion of system memory beyond the host attached memory.

With CCIX, each CCIX-capable device behaves similarly to nodes in an existing NUMA (non-uniform memory access) operating system. This memory-based approach leverages existing operating system capabilities. In this mode, all data structures for data sharing are placed in shared memory accessible by both processors and accelerators. This data sharing model enables the possibility of eliminating the accelerator specific control and management driver allowing scheduling of the accelerators resources to be used by a long running task from a central scheduler. This scheduler may be part of the OS scheduler or work in collaboration with the OS scheduler. This enables simple software libraries for applications running with Virtual Machines or Containers, allowing developers to write regular applications software in any language with full tool support.

CONCLUSION

The CCIX Consortium is an open industry standards body that is developing the non-proprietary CCIX Standard that is independent of any processor architecture. The CCIX Standard is designed for high performance, low latency, chip to chip interconnect for memory expansion devices and accelerators. It is specifically designed for tomorrow's toughest challenges in data center, cloud computing, Big Data and any other application where heterogeneous computing is required. The CCIX Standard will be a revolutionary step forward that extends the benefits of open, heterogeneous architecture and cache coherent shared memory model to meet the evolving demand of future data centers. To learn more about CCIX, go to **www.ccixconsortium.com**.

ABOUT CCIX™ CONSORTIUM, INC.

CCIX™ Consortium was founded to enable a new class of interconnect focused on emerging acceleration applications such as machine learning, network processing, storage off-load, in-memory database and 4G/5G wireless technology. The standard allows processors based on different instruction set architectures to extend the benefits of cache coherent, peer processing to a number of acceleration devices including FPGAs, GPUs, network/storage adapters, intelligent networks, and custom ASICs. CCIX simplifies the development and adoption by extending well-established data center hardware and software infrastructure. This ultimately allows system designers to seamlessly integrate the right combination of heterogeneous components to address their specific system needs.

For more information, please visit https://www.ccixconsortium.com/.

The CCIX Specification is available to all CCIX members. For membership information, please visit https://www.ccixconsortium.com/join.