CHALLENGES WITH UCIE AS A CHIPLET INTERCONNECT

Practical case study from NEX on the use of UCIe as a standard chiplet interconnect

IICS 2024 TECHNICAL PAPER

Pat Fleming
Author

Praveen Mosur

Naveen Lakkakula

Venkidesh Iyer Krishna

Amar Deogharia

Subrahmanya Kumar Kuchibhotla

Chakravarthy Kalyana
Co-authors

> "The primary motivation is to enable a vibrant ecosystem supporting disaggregated die architectures which can be interconnected using UCIe"
>
> – UCIe Specification

Executive Summary

UCIe is intended to provide a standardized die-to-die interconnect that will enable end users to easily mix and match chiplet components from a multi-vendor ecosystem for System-on-Chip (SOC) construction, including customized SoCs. This paper will discuss the use of UCIe as a standard die-to-die interconnect and the challenges and trade-offs encountered in doing so. We will discuss the proposed solution which will be deployed for chiplet interconnect across multiple products but also highlight the challenges in using UCIe in a multi-vendor ecosystem.

Converged Edge SOC Architecture

Edge applications demand compact, ruggedized hardware with efficient power usage and competitive total cost of ownership. Accelerator integration within the package is essential in meeting the constraints of this Edge product. In such a diverse market achieving product scalability and versatility necessitates an approach based on various disaggregated dies. These dies can then be mixed and matched to build a product roadmap that meets Edge market demands.

The goal was to build the accelerator chiplets in a manner that allowed easy reuse by other products and even within a multi-vendor ecosystem. UCIe was already being used for die-to-die interconnect by the Xeon mainstream product line and as a standard for die-to-die interconnect made it our choice of interconnect to the accelerator chiplets.

UCIe is a layered architecture covering the physical layer, link layer and protocol layer. At its protocol layer it supports PCIe, CXL and streaming protocol. In working through the architecture definition, it became clear that UCIe alone does not encompass all the aspects necessary to build a standardized chiplet interconnect. It required the creation of custom protocols and flows beyond the standard to meet the SOC interconnect needs. UCIe standard defines a mainband interface and sideband interface. The challenges encountered on both mainband, and sideband are discussed in this paper.

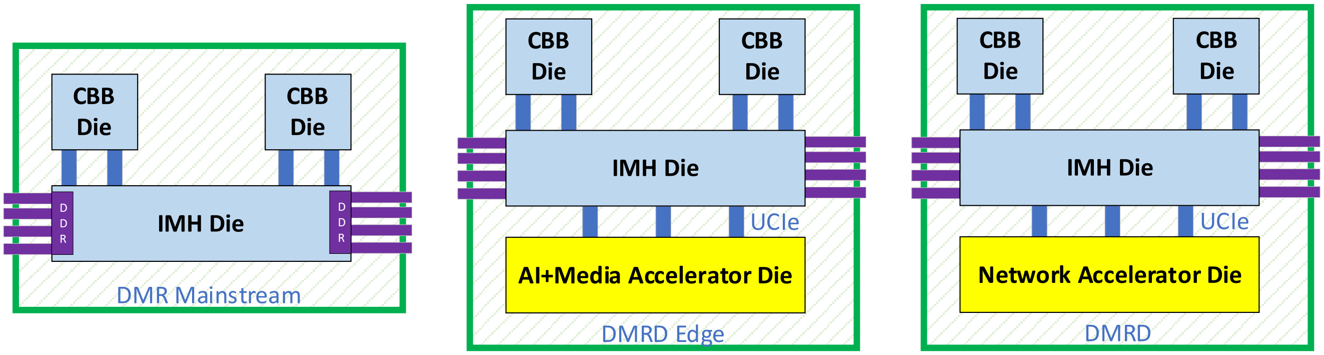

Figure 1: UCIe for accelerator chiplet interconnect across products

Mainband Architecture Challenges

The AI/Media acceleration die on DMR-Edge contains IP for performing AI inference and Media encode/decode tasks. Access to DDR and CPU cache from the accelerator die for this product is via the Integrated Memory Hub (IMH) die. This engine requires exposure to software as a single PCIe/CXL device but requires more bandwidth to the IMH than can be provisioned over a single PCIe/CXL Gen6 stack. To achieve the bandwidth additional UCIe links were required, referred to as data expansion links. This is not an issue specific to AI/Media acceleration but can be viewed as a more generic problem statement where a device requires access to more memory bandwidth than is provided within a single device IO stack such as PCIe or CXL.

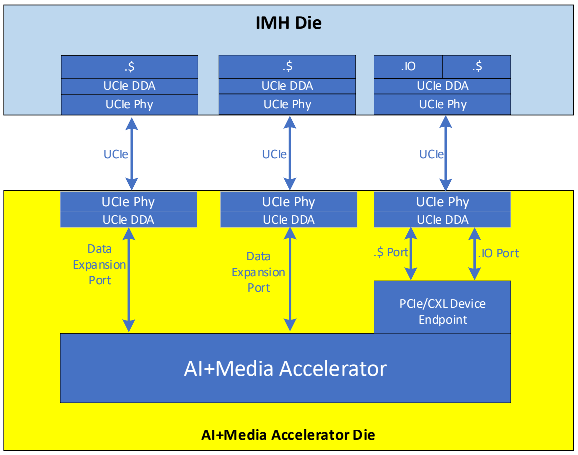

Figure 2: Data bandwidth expansion ports

The .IO port shown in Figure 2 exposes the PCIe/CXL device to SW and allows access to MMIO space within the device as well as accommodating upstream ATS requests, this .IO port is of relatively low bandwidth. The other ports will all issue host physical addresses for direct access to DDR, the most suitable UCIe transport protocol for access to memory across these high bandwidth ports is to use CXL.$. UCIe specification does not currently accommodate this need for UCIe ports without the corresponding PCIe/CXL device hierarchy (root port + endpoint) existing on each of these ports. UCIe would require each one of these ports to have a corresponding PCIe/CXL endpoint device within the accelerator die. These constraints create significant issues for software to manage since the single accelerator would then be exposed via three separate PCIe/CXL endpoint devices.

The proposed architectural solution for this product was to utilize CXL.$ as an encapsulation format over UCIe to provide access to memory via these expansion ports. This required modification of the IO stack in the IMH such that the CXL.$ root port controller could be utilized in the absence of an associated CXL.IO controller and therefore without the need of a downstream CXL endpoint device. These data expansion links are transparent to software and provide a scalable method to provision access to further bandwidth across the die-to-die interconnect.

Sideband Architecture Challenges

The usage of sideband link as defined in UCIe specification is for link training and parameter exchange with the remote link partner. The global flows including reset, boot, firmware download, configuration, power management and security require a sideband definition beyond that of the UCIe sideband physical and link layer definition to ensure die-to-die compatibility. A transport layer and protocol encapsulation layer on top of UCIe sideband link is required to orchestrate global flows agnostic to the on-die bus protocol. The proposed architectural solution was to define a UCIe management transport packet mechanism over UCIe sideband links to provide notification of events corresponding to various global flows.

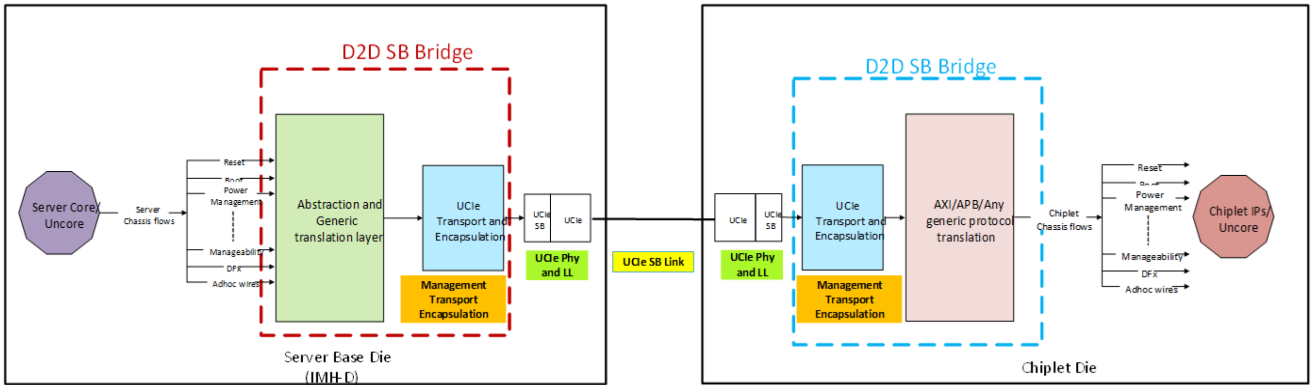

Figure 3 Proposed UCIe sideband transport and protocol translation layers.

Figure 3 captures the abstraction layers proposed for this solution. Each chiplet comes with its own on-die bus fabric and chassis. A generic protocol translation layer is defined to capture various global flow events such as firmware download request. The captured event is communicated using UCIe management transport protocol layer to the other chiplet where it can be captured in a shadow register. The shadow register generates the corresponding global event within the chiplet. The action that is performed based on these events are specific to the chiplet bus protocol and chassis implementation. This abstraction provides a generic view of communicating global flow events which is agnostic to chiplet specific implementation and provides support for the global flows.

Provisioning and access to the chiplet can then be viewed across three memory regions:

1. Application space. Access to this space can be provided as part of standard PCIe/CXL device and protocol over the UCIe mainband interface.
2. Configuration space. This is used for IP configuration such as enabling features, interrupts etc. Access to this space can be provided as part of standard PCIe/CXL device and protocol over the UCIe mainband interface.
3. Private space, which is not visible to BIOS/SW/OS and can be accessed by specific agents only to facilitate specific global flows like firmware download, firmware authorizations, secure policy attestation etc.

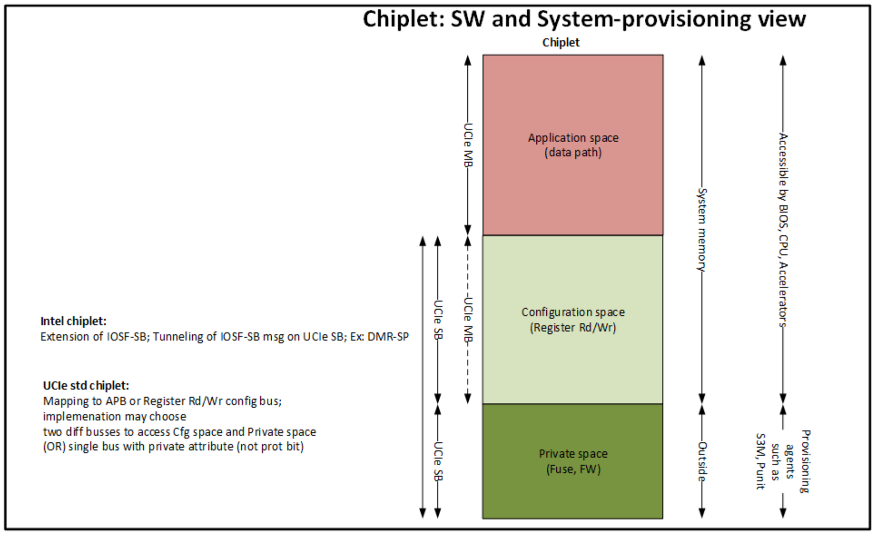
Chiplet: SW and System-provisioning view
Chiplet
Application space (data path)
Configuration space (Register Rd/Wr)
Private space (Fuse, FW)
UCIe MB
UCIe MB
UCIe SB
UCIe SB
System memory
Outside
Accessible by BIOS, CPU, Accelerators
Provisioning agents such as S3M, Punit
Intel chiplet:
Extension of IOSF-SB; Tunneling of IOSF-SB msg on UCIe SB; Ex: DMR-SP
UCIe std chiplet:
Mapping to APB or Register Rd/Wr config bus;
implemenation may choose
two diff busses to access Cfg space and Private space
(OR) single bus with private attribute (not prot bit)

Figure 4: Chiplet provisioning and access view.

Security Architecture Challenges

The chiplets integrated into the package with UCIe interconnects share platform IOs and resources with the SOC. Different from a typical discrete add-in-card, an integrated chiplet with UCIe will rely on the platform and attached SOC for following functionalities.

- For any firmware operated microcontrollers in the chiplet, it will share the SPI/Flash interface with the rest of the SOC and use the platform flash to store, access and download the firmware image.
- Chiplet will leverage the SOC root of trust such as S3M for any firmware image authentication and secure policy attestation.
- The functionality implemented in the chiplet may require it to have secured or isolated memory provisioned by the SOC. Chiplet would need to have a secure access to that provisioned memory
- Chiplet would need to support additional secure features such as TDX connect and secure address translation services and will need to have secure interface with SOC for enforcing policies associated with these features.

To implement the chiplet features outlined in the section above, an interface architecture definition beyond the current UCIe standard definition was required. For certain features this would require choice of an alternate mainband protocol. The challenges are outlined in detail below.

- For sharing the platform SPI, accessing the firmware for the download and to establish the secure attestation and authentication flows with the SOC root of trust, custom operations need to be defined and implemented using UCIe sideband interface.
- To provision secure memory region and enforce the access policies, the chiplet accelerator had to be implemented as a root complex integrated end point with a mainband protocol that can carry the security attributes. Mainband PCie and CXL.IO protocols on UCIe did not support this capability.
- TDX implementation of end points require link encryption that is not covered in the UCie standard definition.

Summary

The UCIe standard forms the basis of a very efficient solution for die-to-die interconnect. Through projects like the one highlighted in this paper Intel can continue to learn and drive the standardization to meet the goals of defining an interconnect that will operate across a broad multi-vendor eco system. The final solution in this case utilizes a UCIe physical layer with customizations across the link and protocol layers to meet the needs of the chiplet interconnect. The following challenges were identified in the UCIe standard:

1. UCIe does not provision for the bandwidth needs to a device beyond that of a single PCIe/CXL IO stack. The recommendation from this work is that a single PCIe/CXL device should be capable of expanding its bandwidth with additional UCIe links.
2. The current UCIe standard does not define a transport or encapsulation protocol for die-to-die communications on UCIe sideband. As a result, vendors are required to define their own custom flows and can thereby break die-to-die compatibility. A transport and protocol encapsulation layer on top of UCIe sideband link is required to orchestrate global flows agnostic to the on-die bus protocols.
3. UCIe does not currently provide support for root complex integrated endpoints, TDX-connect and its

need for link encryption and it does not provide a means to carry additional information such as security access policy hints.

Working around the gaps in the UCIe standard highlighted above led to customizations on top of UCIe operating in streaming mode. The challenges highlighted above, and associated customizations meant that the solution would not meet the requirements needed to operate within a multi-vendor eco system. However, the solution allowed the propagation of security attributes, the support of a root complex integrated endpoint and software transparent expansion of bandwidth to meet the needs of the converged edge SOC architecture.

Next Steps and Opportunities for Intel

In the current environment of disaggregated dies and custom SOCs as well as Intel's ambition to build a foundry services business including ingredient technologies, the availability of a standardized die-to-die interconnect that enables a multi-vendor ecosystem becomes critical. Intel can realize real efficiencies and cost saving opportunities by building upon and maximizing this initial architecture of accelerator dies using UCIe. Taking on board the learnings in this paper across our products can enable the ability to mix and match chiplets across the product roadmap. Progressing the UCIe standard to encompass these learnings presents the opportunity to enable a vibrant multi-vendor, custom product ecosystem.