Understanding Chiplets Today to Anticipate Future Integration Opportunities and Limits

Gabriel H. Loh
Advanced Micro Devices, Inc.
Bellevue, WA, USA
gabriel.loh@amd.com

Samuel Naffziger
Advanced Micro Devices, Inc.
Fort Collins, CO, USA
samuel.naffziger@amd.com

Kevin Lepak
Advanced Micro Devices, Inc.
Austin, TX, USA
kevin.lepak@amd.com

***Abstract*—Chiplet-based architectures have recently started attracting a lot of attention, and we are seeing real-world architectures utilizing chiplet technologies in high-volume commercial production in multiple mainstream markets. In this special session paper, we provide a technical overview of the current state of chiplet technology including its benefits and limitations. This provides background and grounding in the current state-of-the-art and also lays out a range of technical areas to consider for the remaining forward-looking papers in this special session. We discuss the benefits and costs of different approaches to splitting and modularizing a monolithic chip into chiplets. In particular, we cover supporting high bandwidth and low latency communication between the die, mixed integration of multiple process technology nodes, and silicon and IP reuse. We then explore future challenges for chiplet architectures looking into the next decade of innovation.**

Keywords—chiplets, integration, process technology

I. Introduction

Over the past several years, the continued slowing down of Moore's Law combined with the end of Dennard Scaling has created a variety of potential headwinds for the continued improvements of processor designs. In response to these challenges, the industry is increasingly looking toward advanced integration and packaging technologies to help keep processor capabilities moving forward [1].

While the general idea is not new [2], recent advancements in certain technologies have made "chiplets" a viable and effective technology to help fight against the slowing of Moore's Law. Traditionally, microprocessors are implemented as a single monolithic die of silicon. The chiplet approach takes a system-on-chip (SoC) design and repartitions it across multiple smaller chiplets. Coupled with integration technologies that allow high-speed communications between chiplets, the functionality and performance of a monolithic SoC can potentially be realized in a more cost-effective and scalable module with multiple chiplets.

II. Current State of Chiplet Technology

A. Motivation

The historical rate of Moore's Law delivered a doubling in the number of transistors per unit area every 18 to 24 months. In turn, processor designers have continually used the additional device count to construct more powerful microprocessors. Figure 1 plots the peak performance of the world's fastest supercomputers over the past several decades. Even before the slowdown of Moore's Law, the peak performance of these machines was increasing faster than the rate of the corresponding device densities. This trend does not appear to be slowing down as several recent exascale-class supercomputers have been announced [3][4]. The challenge for the industry is finding ways to continue to deliver ever increasing performance and capabilities in a world where the underlying silicon no longer provides the same historical rate of density improvements.

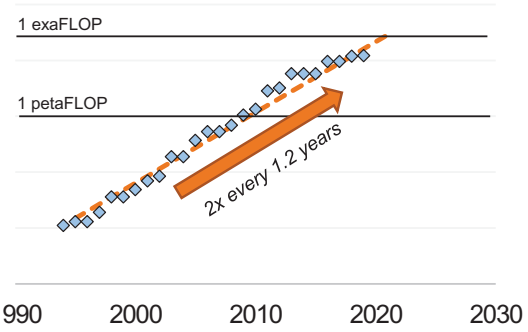

Figure 1. Peak performance of the world's fastest supercomputers.

B. Background

If the underlying silicon technology is not providing the increases in device density that one historically would expect, one possible path for increasing processor capability and performance is to build larger chips. As a hypothetical example, if a new technology node only provides a 1.5× increase in device density, then building a chip that is 33% larger can still provide an overall increase in total device count

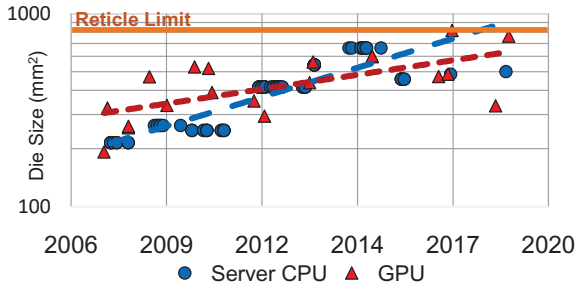

Figure 2. Microprocessor die sizes over time [1].

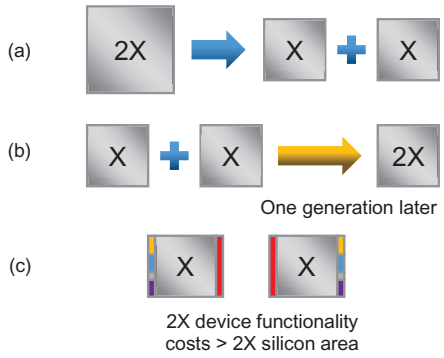

Figure 3. (a) A silicon chip decomposed into two chiplets, (b) the functionality of two chips historically could be achieved with a technology shrink, and (c) illustration that dividing a chip into chiplets can incur additional silicon area overheads.

of 2.0×. The sizes of microprocessors have slowly increased over the years, as shown in Figure 2, but the sizes of the chips are now also running up against the lithographic reticle limit beyond which further increases in die sizes are no longer practical. Furthermore, such reticle-scale silicon die are very expensive to manufacture because the larger chip area greatly increases the probability of incurring one or more manufacturing defects, thereby causing lower yields and consequently higher costs.

The idea behind chiplets is that instead of building a single, large, monolithic chip, multiple smaller chips can be manufactured and then combined to implement what is logically a single microprocessor. Figure 3(a) illustrates a chip with 2*X* devices decomposed into two smaller chiplets with *X* devices each. The smaller size of the chiplets improves their yields, such that in some scenarios the combined cost of the two smaller chiplets can still be less than the cost of the single larger chiplet. This approach could have been utilized in the past; however, while Moore's Law was still delivering steady technology improvements generation after generation, there was far less incentive to use such a chiplet approach. As shown in Figure 3(b), by waiting for the next technology node, the doubling in device count was enough to provide the increased functionality. Another reason why chiplets historically may have been less attractive is that the "chipletization" of an SoC does not come for free. As shown in Figure 3(c), there is a variety of functionality that must be instantiated on a per-chiplet basis (e.g., clocking circuitry, power management, testing, debug) as well as the area required for new inter-chiplet communication interfaces that would otherwise not be needed in a monolithic design. The result is that the 2*X* devices' worth of functionality would require more than 2*X* devices' worth of total silicon area.

However, with the slowing of Moore's Law and the rising costs of leading-edge process technologies, along with the maturation in the necessary integration and packaging technologies, chiplets have now become an attractive approach to building high-performance microprocessors.

C. The Chiplet Approach

An illustrative example of the manufacture and assembly of conventional monolithic chips is shown in Figure 4(a). Multiple die are manufactured on a single silicon wafer. The individual die undergoes functional testing to determine which ones are functional and can be packaged and sold, and

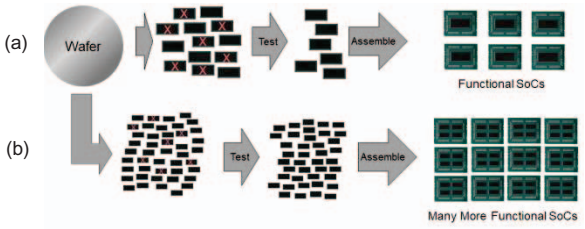

Figure 4. (a) Manufacture, test, and assembly of SoCs from monolithic silicon die, and (b) the same for SoCs assembled from multiple chiplets.

which ones are faulty and must be discarded. The number of sellable die per wafer depends heavily on the die size. The larger a silicon die is, the more likely a manufacturing fault may occur. To put it another way, for the same number of total faults per wafer, a larger die size results in more discarded silicon per fault. A secondary effect is that due to the round shape of silicon wafers and the rectangular shape of individual die, there is a basic geometric packing challenge in that larger die result in more wasted silicon around the periphery of the wafer, resulting in even fewer yielded die per wafer. Finally, the die that pass testing can be packaged and sold as complete SoCs.

The chiplet approach, shown in Figure 4(b) starts off similarly but with smaller die. This yields a larger number of die per wafer, and then a larger number of chiplets (compared to the monolithic case) likewise remain after testing. These individual "known good die" (KGD) can then be packaged together to yield a larger number of sellable SoCs. The increased cost of leading-edge silicon manufacturing combined with the maturation in in-package interconnect technologies has made the chiplet approach a commercially viable strategy for constructing high-performance SoCs.

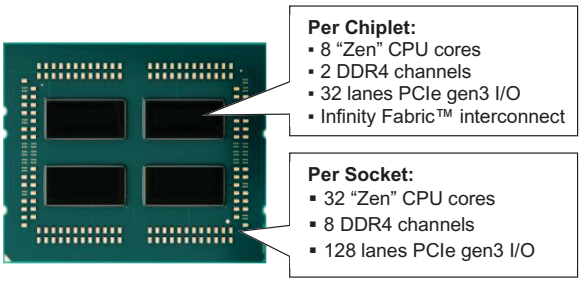

Figure 5. Quad-chiplet first-generation AMD EPYC™ processor.

D. Chiplets for Cost and Yield

AMD's initial foray into a chiplet-based microprocessor design was embodied in the first-generation AMD EPYC™ CPU processor, launched in 2017 and shown in Figure 5. This processor consisted of four identical chiplets each implemented in a 14nm process technology. Each individual chiplet delivers eight first-generation "Zen" CPU cores, two DDR4 memory channels, 32 lanes of PCIe® gen3 I/O, and AMD Infinity Fabric™ interconnect to provide inter-chiplet communication. A single package consists of four chiplets, providing a total of 32 CPU cores, eight DDR4 memory channels, and 128 lanes of PCIe I/O. High-speed interconnects across the package substrate provide the links between the chiplets. Because the inter-chiplet distances are relatively short within the package, highly-optimized high-

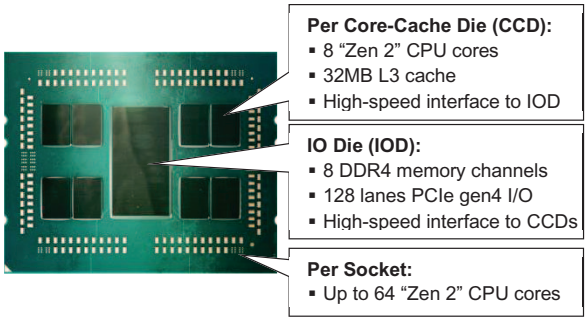

Figure 6. Mixed process node chiplet organization for the second-generation AMD EPYC processor.

bandwidth SerDes could be deployed without the use of more expensive solutions such as silicon interposers [5].

This approach was effective for multiple reasons. First, the use of chiplets enabled the construction of an SoC that utilized more total silicon area than could normally be manufactured due to lithographic reticle size limitations (i.e., the total silicon of the four chiplets exceeds the area of the reticle limit). The availability of this much silicon enables the instantiation of more total CPU compute than could be attained by conventional monolithic single-chip designs. Second, putting aside the reticle limit, even if an equivalent monolithic SoC with the same number of cores, memory channels, and I/O could hypothetically be built, our cost estimates indicate that utilizing chiplets could be 41% less expensive to manufacture.

E. Chiplets for Mixed Process Nodes

While the first-generation EPYC™ processor approach was very effective in delivering a high-performance and cost-effective processor solution, there remained other opportunities that an advanced chiplet approach could leverage. Figure 6 shows the organization of the second-generation EPYC™ processor. Whereas the first-generation approach utilized four identical chiplets, the second-generation architecture uses two different die. In the center of the package is the I/O die (IOD), which provides all of the DDR memory controllers and physical interfaces (PHYs), PCIe, other I/O (e.g., USB, SATA), the scalable data fabric (also known as a network on chip), system coherence tracking, and a variety of other functionality (e.g., power management, security, test). Surrounding the IOD are up to eight core-cache dies (CCDs), where each CCD provides eight "Zen 2" CPU cores, the accompanying cache hierarchy, and a high-speed interface to the IOD.

The second-generation EPYC™ processor utilizes a mix of different silicon process technologies. The CCDs are implemented in a 7nm process node, which provided the highest density and performance technology at the time this product was designed for. The higher device density is an important factor in enabling the second-generation EPYC™ processor to deliver twice the total number of cores compared to its first-generation counterpart (i.e., 64 versus 32 cores) despite using the same size package. On the other hand, the IOD is implemented in an older 12nm process node. Many of the analog components of the IOD (e.g., DDR4 and PCIe PHYs) take up a lot of area and their circuit sizes do not scale well with shrinking device sizes. That is, implementing these analog circuits in an advanced process node neither

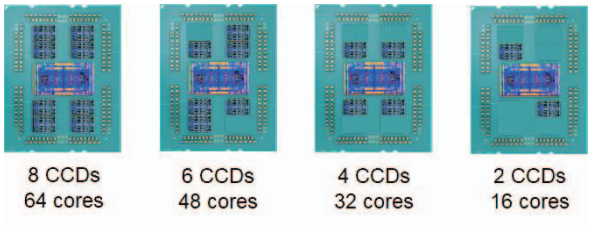

Figure 7. AMD second-generation EPYC processor in 64, 48, 32, and 16 core offerings achieved by varying the number of CCDs.

appreciably reduces the die size nor delivers any additional functionality or end-user benefit (e.g., a DDR4 interface implemented in 12nm versus 7nm externally delivers the same functionality).

The separation of the CPU cores into CCDs provides a new level of flexibility and configurability in providing a complete product stack. For conventional SoCs based on monolithic designs, a processor vendor that wanted to offer N different products with varying core counts would have to create N separate SoCs. While some engineering effort and IP reuse can be leveraged across the designs, many other costs increase on a per-SoC basis. This includes manufacturing-related overheads such as mask sets as well as per-SoC engineering work including a lot of the physical design (e.g., power distribution, clocking). Figure 7 shows how the approach used for the second-generation EPYC™ processor utilizes two unique chips (i.e., the CCD and IOD) to provide an entire line up of processors from 16 cores up to 64 cores. In a conventional non-chiplet methodology, these four processors could have required up to four separate tapeouts.

This approach of utilizing a single IOD for all configurations also provides additional benefits. Regardless of how many CCDs are populated for a given part, the entire complement of DDR4 channels and PCIe lanes can be available. This provides the customer with a "fully featured" solution independent of the core-count requirements for their use case. For example, a server optimized for storage may desire a large number of PCIe I/O lanes to maximize the number of the connected storage devices, but as this application is typically I/O bound, a relatively smaller number of CPU cores may be more than sufficient. With conventional monolithic design approaches, the larger I/O capabilities were often limited to the higher core-count products, which forces customers to pay a premium for the additional cores that their application cannot even utilize.

III. Future Challenges

While chiplet-based designs have already enabled some very exciting architectures for AMD, the industry as a whole is still in the early days of chiplet design. In this section, we discuss some of the research challenges that lay ahead for expanding chiplets to a broader range of potential use cases and designs.

A. Chiplet Generality versus Optimization

The first challenge is in striking an effective balance between generality and optimization. The idea of a generic chiplet infrastructure is not new [2], where some form of a common substrate can act like the equivalent of an in-package circuit board, and then arbitrary combinations of chiplets can be integrated upon this substrate to quickly deploy new

solutions. However, generalization can cause trade-offs in performance, cost, power, or other important factors. For example, chiplets implementing different IP may have different bandwidth requirements. By designing an interface for the most demanding chiplets (e.g., higher pin counts, higher signaling rates), this can cause other chiplets to have to pay a higher area cost to implement the same common interface. Instead, if multiple different interfaces are defined (e.g., a high-performance and a more modest-performance interface), then the generality and flexibility of the underlying substrate may be compromised, as not all chiplet-mounting sites on the substrate may support both types of interfaces, or the cost of the substrate may increase if all sites are to support both types of interfaces. The same type of trade-off argument can be made for other aspects of the chiplet interface, such as whether all chiplets need to support cache coherency, virtualization, security functions, and more.

The debate on generality versus optimization also extends to the physical design of the overall package. Power must be delivered to all of the chiplets, and if an arbitrary set of chiplets could potentially be integrated into a system, the package may end up overprovisioned if it is designed to deliver enough power for the worst case where every chiplet integration site is populated with the maximum power-consuming chiplet. This can result in a package that is over-designed for systems implemented with a less demanding set of chiplets. This extends to more than just power delivery, but also to the handling of worst-case instantaneous current demands (di/dt), long-term current loads (electromigration), required decoupling capacitors, and more.

B. Architectural Partitioning

For a given system, there can be many possible ways to partition its functions across multiple chiplets. The first two generations of AMD EPYC™ processors showed two different approaches with different benefits related to cost, performance, flexibility, engineering effort, and the need to address different market requirements. The overall design process can involve a highly complex multi-dimensional optimization challenge. Even for the design of the second-generation EPYC™ processor, the general CCD and IOD approach could still have been done many other ways. For example, the system could have been implemented with a few 16-core CCDs or a larger number of four-core CCDs. The question of architectural partitioning can be a tricky one when deciding how to partition a specific SoC. The challenge becomes even more difficult when trying to decompose and partition multiple chiplets to support multiple simultaneous designs which could include chiplet combinations not yet even imagined.

C. Memory System Concerns

If future design methodologies utilize a diverse set of chiplets, especially if some chiplets might be sourced from third-party providers, then a variety of memory-related issues must be worked through. What is the right cache coherence protocol for a platform that may include a wide range of chiplet types? What is the minimum required coherency support that each chiplet must implement so they can work correctly with the larger overall system? What should the overall memory consistency model be so that programmers can meaningfully reason about the behavior of the memory system in the context of multiple diverse chiplets? Can different chiplets support different memory consistency models appropriate for their uses, and how would they all interact in the global memory system?

D. Security

Integrating multiple chiplets, including third-party IP, can provide a highly productive ecosystem for creating new products. However, this vision of a future chiplet-based platform could also provide myriad opportunities for security concerns. Some situations may be unintended or malicious, such as a chiplet issuing a large number of memory requests that effectively causes a denial-of-service for the other chiplets in the system. It is conceivable that a chiplet may be able to glean sensitive information from the rest of the system by snooping traffic on the shared interconnect or through coherence traffic. Being directly integrated in the same package could provide opportunities for malicious chiplets or software running on some of the chiplets to find and exploit new side-channel attacks into the rest of the system.

These are a handful of potential research areas that could be fruitful for the academic community to explore. While these have been presented in the context of chiplets, many of these issues are fundamental to any type of future integration technology that leverages multi-chip designs.

IV. Conclusions

The chiplet era has begun, and AMD has already successfully utilized this approach through two generations of high-performance EPYC™ processors. Chiplets can provide many benefits in the context of processor design in a post-Moore's Law world, but effective utilization of chiplets still requires careful engineering and optimization along many different and sometimes conflicting dimensions. As future integration technologies continue to improve, the opportunities to utilize chiplets will likely continue to expand, but there remain many open research challenges to tackle before a truly expansive and general chiplet methodology and broader ecosystem can really become ubiquitous.

Acknowledgments

Special thanks to the talented AMD design teams from around the world for the AMD EPYC™ server processor engineering achievements.

References

[1] Lisa T. Su, Samuel Naffziger, Mark Papermaster, "Multi-Chip Technologies to Unleash Computing Performance Gains over the Next Decade", IEDM Conference 2017.

[2] Bryan Black, "Die Stacking is Happening", keynote at the 46th International Symposium on Microarchitecture, December 2013.

[3] Oak Ridge National Laboratory, "U.S. Department of Energy and Cray to Deliver Record-Setting Frontier Supercomputer at ORNL", *press release*, May 7, 2019.

[4] Lawrence Livermore National Laboratory, "LLNL and HPE to Partner with AMD on El Capitan, Projected as World's Fastest Supercomputer", *press release*, March 5, 2020.

[5] Natalie Enright Jerger, Ajaykumar Kannan, Zimo Li, Gabriel H. Loh, "NoC Architectures for Silicon Interposer Systems", in the 47th International Symposium on Microarchitecture, December 2014.