SCIENCE CHINA
Information Sciences

• REVIEW •

October 2024, Vol. 67, Iss. 10, 200401:1–200401:20
https://doi.org/10.1007/s11432-023-3926-8

Special Topic: AI Chips and Systems for Large Language Models

Review of chiplet-based design: system architecture and interconnection

Yafei LIU[1], Xiangyu LI[2*] & Shouyi YIN[1*]

[1]*School of Integrated Circuits, Tsinghua University, Beijing* 100084*, China;*
[2]*Laboratory of Integrated Circuits and Intelligence Systems, Research Institute of Tsinghua University in Shenzhen, Shenzhen* 518057*, China*

Received 24 June 2023/Revised 28 September 2023/Accepted 26 December 2023/Published online 19 July 2024

Abstract Chiplet-based design, which breaks a system into multiple smaller dice (or "chiplets") and reassembles them into a new system chip through advanced packaging, has received extensive attention in the post Moore's law era due to its advantages in terms of cost, performance, and agility. However, significant challenges arise in this implementation approach, including the mapping of functional components onto chiplets, co-optimization of package and architecture, handling the increased latency of communication across functions in different dies, the uncertainty problems of fragment communication subsystems, such as maintaining deadlock-free when independently designed chiplets are combined. Despite various design approaches that attempt to address these challenges, surveying these approaches one-after-another is not the most helpful way to offer a comparative viewpoint. Accordingly, in this paper, we present a more comprehensive and systematic strategy to survey the various approaches. First, we divide them into chiplet-based system architecture design and interconnection design, and further classify them based on different architectures and building blocks of interconnection. Then, we analyze and cross-compare each classification separately, and in addition, we present a topical discussion on the evolution of memory architectures, design automation, and other relevant topics in chiplet-based designs. Finally, some discussions on important topics are presented, emphasizing future needs and challenges in this rapidly evolving field.

Keywords chiplet-based design, package, architecture, interconnection, silicon interposer

1 Introduction

In recent years, the progression of Moore's law has slowed down while the demand for high-performance integrated circuits (ICs) and the diversification of their applications has continued to grow. The conventional monolithic chip design approach, though widely adopted, presents limitations in cost-effectiveness, performance, and agility [1, 2]. Fortunately, the chiplet-based design approach offers a promising solution by disintegrating the traditional monolithic silicon chip into multiple smaller chiplets and then reassembling them into a new system chip [3]. Chiplet refers to an IC block that has been specifically designed to communicate with other chiplets, to form larger more complex ICs [4]. To be emphasized, a chiplet is a reusable prefabricated component in the form of a die. As a promising technology, chiplet technology has received extensive attention in the post-Moore's law era [3, 5–8].

Chiplet-based design has emerged as a crucial solution to break limitations in increasing integration level, reducing time-to-market (TTM), and improving the energy efficiency of ICs. By breaking down the monolithic chip into smaller dice, the chiplet technology improves yields, reduces design complexity, and overcomes reticle limitations. As a result, multiple times, even one to two orders of magnitude larger scale chip systems become achievable. The chiplet technology facilitates the reuse of the silicon-proven fabricated dice because they can be assembled together to form a new system through a relatively short processing step. The TTM and non-recurring engineering (NRE) costs associated with the back-end design process and chip manufacturing can be diluted. As the chiplets in an integrated system are manufactured independently and can be assembled in various structures, heterogeneous integration is also enabled, as a result, additional optimization space is opened up and system iteration and upgrade are

[*] Corresponding author (email: xiangyuli@tsinghua.edu.cn, yinsy@tsinghua.edu.cn)

also simplified. Consequently, companies such as AMD, Intel, and TSMC have already used the chiplet technology in their products and solutions [9–12].

However, chiplet-based design faces many challenges, including die-to-die interconnection, packaging, power supply, thermal management, design methodologies, and electronic design automation (EDA) tools. Among them, a new special challenge in chiplet-based design is the architecture design, which is the top-level problem that governs the system performance and cost and is related to most of the issues mentioned above. Moreover, in chiplet-based design, the coupling between the packaging, system architecture, and interconnection becomes tighter than in the traditional monolithic system on chip (SoC). The trade-off between performance, power, area, and cost (PPAC) requires careful choices within the design space of packaging, system architecture, and interconnection. Unfortunately, no general and clear system-level architecture design method has been proposed. Therefore, it is valuable to summarize and analyze the state-of-the-art chiplet-based designs in order to provide designers with valuable design schemes for reference. This task involves reviewing the works in the aspects of function mapping, system floorplan, and interconnection fabric architecture.

Several surveys have already focused on some key technologies and fundamental modules of chiplet-based design. Studies in [13, 14] have provided a comprehensive overview of chiplet-based designs from a packaging perspective. Similarly, in [14–16], the interconnect interfaces of chiplets have been summarized, with a focus mainly on the physical layer. However, further investigation that can provide guidance for system architecture design is desired in addition. The study [17] has overviewed the computing architecture of high-performance computing (HPC) systems. However, it mainly classifies and introduces architectures from the perspective of packaging technology rather than the perspective of design, lacking the analysis of the system architecture and especially the architecture of interconnection networks. Furthermore, the analysis is limited to computing systems for applications such as HPC, mobile, and personal computers (PC), while there are many other potential application fields for which the chiplet technology is valuable. Accordingly, this article investigates and analyzes chiplet-based design issues from the perspective of system architecture and interconnection design, with a wider range of application domains.

We declare the following contributions:

- We provide a high-level classification of the system architecture of existing chiplet-based designs from a design perspective, and highlight the challenges and evolution of the memory subsystem in chiplet-based designs.
- We demonstrate the significant trend of platformization in chiplet-based system design and provide a systematic analysis of existing solutions.
- We present a summary of emerging chiplet-based interconnect topologies and analyze the evolving optimization of the network topology on active silicon interposers.
- We analyze and summarize the state-of-the-art modular routing schemes in chiplet-based systems and provide a comparative analysis of centralized and distributed routing approaches.

The remainder of the paper is organized as follows: Section 1 is the introduction. Section 2 presents an overview of packaging technologies and highlights the design challenges faced by chiplet-based systems. Section 3 reviews chiplet-based designs from the perspectives of system architecture and interconnection. It includes analyses of typical chiplet-based system architectures and interconnections, as well as a discussion on emerging technologies. Section 4 explores important topics for future research, with a focus on challenges and needs. Section 5 is the summary of the purpose, content, and conclusion of this paper.

2 Packaging technologies and design challenges

2.1 Packaging technologies

In this paper, we use the definition and classification of advanced packaging from articles [18–20]. The packaging technologies used in chiplet-based systems are classified into 2D, 2.xD, and 3D [19], as shown in Figure 1 [18].

In 2D packaging, multichips are mounted in a single plane on a package substrate or fan-out redistribution-layer (RDL) substrate [20]. The 2D package offers lower interconnect density than that of the silicon die, but it is a mature technology with cost-effectiveness, high yield, and robustness.

In 2.xD (x=1, 3, 5) packaging, chiplets are integrated on a substrate by a fine pitch interposer,

https://engine.scichina.com/doi/10.1007/s11432-023-3926-8

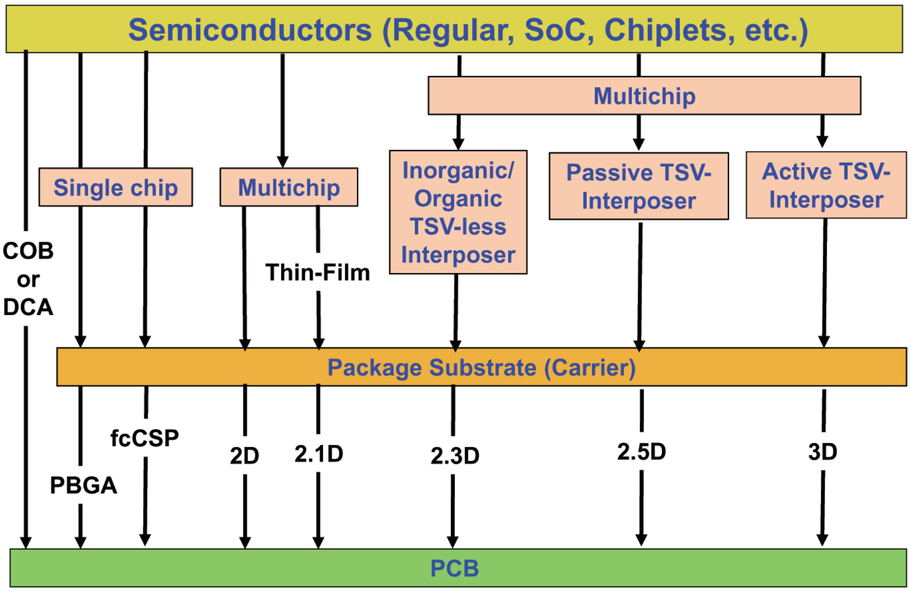


Figure 1 (Color online) Groups of advanced packaging: 2D, 2.xD, and 3D packaging [18] Copyright 2022 IEEE.

including thin-film layers or embedded bridges (2.1D), separately fabricated fine pitch through-silicon via (TSV)-less interposer (higher interconnect density than thin-film layer) (2.3D), and passive TSV silicon interposer (2.5D) [20]. In particular, the 2.5D package is a widely employed package type [21–24], in which the passive TSV silicon interposer is a piece of silicon with TSVs and RDLs [20], and a typical example is TSMC's chip-on-wafer-on-substrate (CoWoS) [25]. Compared with the 2D package, the 2.xD package offers much higher interconnect density, yet introduces higher package cost.

In 3D packaging, or vertical integration, multiple layers of dies stack vertically, interconnected by TSVs bounded with micro-bump (μbump) or bumpless hybrid bonding [20]. Remarkably, the widely adopted active interposer-based designs, such as the AMD 3D V-cache stack die [26] and Intel Lakefield [27] with Foveros technology [28], are typical of 3D packages [20]. Compared to the 2.5D package, the 3D package offers even higher performance, lower energy per bit transition, and a smaller form factor. However, it suffers from high technology complexity and high cost. Moreover, in 3D packaging, the thermal problem is more challenging compared to 2D/2.5D packaging. In 2D/2.5D packaging heat extraction can be performed from the top package face. However, in the case of die stacking, thermal dissipation is a more significant challenge as multiple layers of compute dies need to dissipate their heat on top of themselves particularly when the logic part has high power consumption.

2.2 Design challenges

The main system-level technology issues in chiplet-based design include the following interrelated aspects: system architecture which is mainly chiplet partition, interconnection design and package scheme. In detail, they respectively plan the mapping of the computing and processing functions, their physical floorplan, and the interconnect fabric between them across multiple chiplets.

System architecture. The primary challenge in the aspect of system architecture is to choose the chiplet partition scheme while addressing the potential performance degradation that may result from the split. as the bandwidth and latency between chiplets generally tend to be inferior compared to those within the same chip. Moreover, the challenges faced by systems with different computing architectures, such as homogeneous multicore processors and heterogeneous multiprocessor SoCs (MPSoCs), are quite different. In addition to the memory access bottlenecks and reticle limitation, the former mainly concerns scalability and inconsistencies in access latency due to splitting, while the latter may be more concerned with compute efficiency. Moreover, additional costs such as additional silicon area of interface circuits, additional package cost, and additional masks, and power consumption overhead associated with the chiplet interface must also be taken into account. Furthermore, the reusability of the chiplets is also an important factor to consider, which greatly determines the cost of the system.

Interconnection design. There are two differences between the interconnection in chiplet-based

https://engine.scichina.com/doi/10.1007/s11432-023-3926-8

systems and in monolithic chip systems, and these differences create design challenges. First, the characteristics of interconnect links in chiplet-based systems are not uniform. Inter-chiplet interconnects have lower bandwidth and higher latency compared to intra-chiplet interconnects due to physical limitations, such as limited input-output (IO) ports and additional latency from drivers and receivers. Second, the chiplet-based approach leads to the problem of fragmented network-on-chip (NoC), where each chiplet contains only a piece of the overall network and these pieces are usually designed and validated independently. This leads to additional communication hops and introduces more uncertainty in traffic, such as new deadlocks that may occur when different chiplets are assembled together. Therefore, deadlock-free routing technology is crucial in chiplet-based design.

Both the system architecture and the interconnection design are heavily dependent on the packaging scheme. The bandwidth of a die-to-die interface and the latency over its link are dominated by the package scheme for a large scale chiplet. Additionally, the cost of the package is also non-negligible under the current conditions. Therefore, co-optimization of package-architecture is crucial for achieving optimal PPAC in chiplet-based systems.

3 Literature review

3.1 System architecture

The crucial aspect of system architecture design is the mapping of system functions to chiplets. Previous studies [5, 29] classify these mapping scenarios into two categories: package aggregation and SoC disaggregation. In the former case, the systems are mainly based on the aggregation of board design into a package, including the aggregation of processors with memory and the aggregation of processors of different architectures. In the latter case, it represents the partitioning of a monolithic SoC into two or more chiplets [29]. Following this line, we will first review the existing literature from an architectural perspective and then analyze and discuss each of them.

3.1.1 *Package aggregation*

In the context of chiplet, existing package aggregation chips include scenarios such as aggregating compute processor with memory, CPU with other heterogeneous processors, such as graphics processing unit (GPU).

To address the increasing demands for processor-memory bandwidth, a viable solution is to physically bring the memory chips closer to the compute processor, as summarized in [30, 31]. Moreover, emerging processing-in-memory (PIM) architectures based on nonvolatile memory (NVM) are introduced in [32,33]. Another way to improve processor-memory access is to use embedded memory or integrate on-package memory (i.e., using chiplet technology), as discussed in the study [34]. An example of this is Intel's Haswell processor [35], which integrates embedded dynamic random access memory (eDRAM) next to the CPU in the same package connected by on-package-IO (OPIO). The eDRAM serves as a fourth-level (L4) cache that is dynamically shared between the CPU and the on-die graphics, providing low-power high-bandwidth memory access to meet the needs of high-performance graphics segments [36]. This on-package approach delivers 3× bandwidth at 1/10 power compared to its double data rate 3 (DDR3) counterpart [37]. GPU design can also benefit from the on-package memory. For AMD's Radeon$^{\text{TM}}$ Fury [38], the entire GPU memory system is integrated into the package using a silicon interposer. Specifically, the GPU die is located in the center and surrounded by high bandwidth memory (HBM) stacks [39] which offers 512 GB/s of bandwidth. In addition, both the GPU and the HBM stacks are connected to the interposer using 45 μm pitch μbumps, and the interposer is mounted to the substrate via controlled collapse chip connection (C4) bumps [40]. As a result, this HBM and interposer design provides 60% more bandwidth than graphics double data rate 5 (GDDR5) while consuming 60% less power [38]. Memory access bandwidth is also a bottleneck in artificial intelligence (AI) processor chips, which are designed for high-performance and low-power computing [30]. The Ascend 910 AI chip from Huawei features an on-chip, ultra-high bandwidth network that links multiple Da Vinci cores [41], as shown in Figure 2(a) [38]. With on-chip L2 cache offering 4 TByte/s bandwidth and HBMs offering off-chip 1.2 TByte/s bandwidth, the high-density computing core's performance can be fully utilized. Leveraging 2.5D packaging technology, the Ascend 910 chip integrates eight dies including a compute die, some HBM stacks, an IO die (IOD), and two dummy structure dies [41]. It should be noted that the

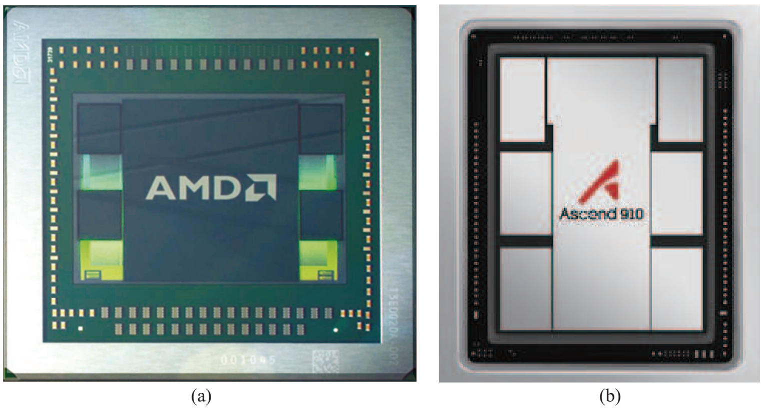


Figure 2 (Color online) Package aggregation. (a) AMD Radeon$^{\text{TM}}$ Fury [38] Copyright 2015 IEEE; (b) Huawei Ascend 910 [41] Copyright 2019 IEEE.

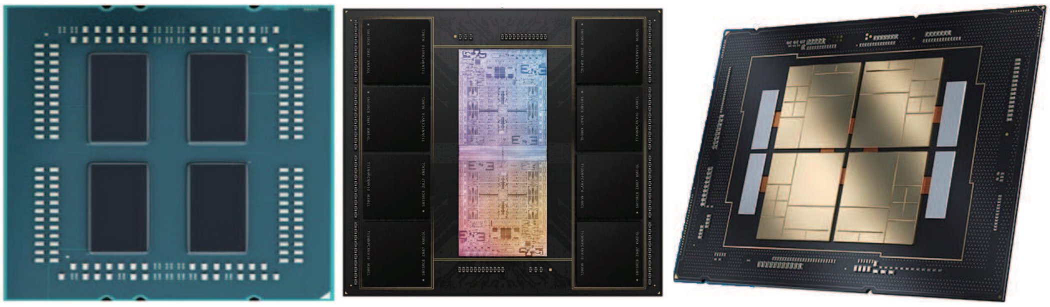
HBM is a 3D-stacked memory device in which multiple DRAM dies are stacked on top of the logic die and connected by TSV.

With the advancement of architecture technologies, the traditional boundary between compute and memory in this kind of "compute die+HBM" scheme is being broken. PIM architectures are evolving based on HBM. The PIM-HBM architecture, proposed in [42], embeds the PIM cores into the logic base of HBM. As a result, the high DRAM access computations can be performed by the in-memory logic die, which clearly reduces the data movement in the memory hierarchy. Inspired by this, emerging memory technologies such as near data processing (NDP) architectures and PIM architectures have been intensively investigated. Obviously, these new PIM architectures are no longer typical package aggregation schemes.

In addition to the on-package memory, multiple heterogeneous processors can be integrated in the same package to boost system performance. For instance, Intel's Kaby Lake-G processor integrates a discrete graphics processor produced by AMD (Radeon$^{\text{TM}}$ RX Vega M) into the same package as the microprocessor. The CPU is connected to the GPU via 8 on-package PCI express (PCIe) links, leading to a performance improvement of up to 40% and 50% board space savings compared to the typical motherboard design [43]. Additionally, data-intensive products such as network interface cards (NICs) also benefit from the high-speed interface provided by package-integration. SmartNIC of Corigine, a chiplet-based system, utilizes a dedicated data processing unit (DPU) to offload the CPU workload for performance and power efficiency. The DPU and CPU are connected by high-bandwidth on-package links [44]. Moreover, Intel introduced a package-integrated "CPU+FPGA" architecture, allowing CPU and custom-designed FPGA logic to share a single physical memory via high-bandwidth on-package links [45]. This significantly improves the performance of memory limited algorithms such as machine learning (ML) as noted in [46].

Analysis. The package aggregation schemes employ mature board-level links and packaging process. This approach has the advantage of high yields and low cost. Moreover, chiplets are designed in a similar manner to traditional monolithic designs, requiring only a high-speed interface to be added and minimal modifications to the architecture. This decreases design risk and complexity. Additionally, the chiplets are relatively functionally complete and independent. Therefore, it is possible to compose products in a single-chiplet form first and then in a multi-chiplet solution according to the research and development (R&D) timeline. This facilitates the TTM of products with varying performance levels.

The mapping approach in package aggregation schemes is relatively straightforward, that is mapping chips in the board-level system to dies in the package-level system. In some cases, multiple chips can be merged into a single die. Interconnections between the chips are realized through common industry-standard interfaces, such as PCIe. Apparently, this implementation benefits from the high-bandwidth, low-latency, and low-power interface provided by the on-package integration. More importantly, the design effort of the chiplets is minimized, and only additional high-speed interface technologies are required to be developed, which reduces design risk and complexity. Unfortunately, the resulting schemes are not always optimal. On one hand, the traditional board-level interfaces are too heavy (high power consumption, high latency) for systems in packages and, therefore, cannot meet the increasing energy efficiency requirements. On the other hand, the partition of functions is constrained by the available chips or the



(a) (b) (c)

Figure 3 (Color online) SoC disaggregation. (a) AMD EPYC$^{\text{TM}}$ processor [9] Copyright 2018 IEEE; (b) Apple M1 Ultra [50] Copyright 2022 IEEE; (c) Intel Sapphire Rapids [51] Copyright 2021 IEEE.

existing designs whose functions are not exactly optimized for the new system.

3.1.2 *SoC disaggregation into homogeneous chiplets with integrated fabric*

It has been demonstrated that the partitioning of monolithic SoCs into smaller tightly coupled dies can overcome the reticle limitation and improve die yield [9, 43, 47]. Thus, in this way, a typical case of SoC disaggregation is to divide the monolithic SoC into several small SoC chiplets, where each chiplet is functionally independent and can work as a monolithic SoC by edge-to-edge interconnection [3].

The first-generation AMD EPYC$^{\text{TM}}$ processor utilized a design composed of four identical chiplets shown in Figure 3(a) [9], codenamed “Zeppelin” [9, 48]. Each chiplet provides 8 “Zen” CPU cores, a memory controller (MC), and an integrated southbridge. Moreover, the Infinity Fabric$^{\text{TM}}$ (IF) is integrated into each chiplet to interconnect the four chiplets. Significantly, this solution allows the same Zeppelin die to be designed into separate packages and provides competitive solutions in several different market segments [48]. For instance, a 4-chip package (EPYC$^{\text{TM}}$) is configured for the server market, a 2-chip package (Ryzen$^{\text{TM}}$ Threadripper$^{\text{TM}}$) and a single-chip package (Ryzen$^{\text{TM}}$) are configured for high-end desktop and mainstream client markets, respectively [48].

In addition to the substrate (2D), a silicon interposer (2.5D) can be used for the connection of the chiplets. For instance, TSMC has proposed a dual-chiplet interposer-based system-in-package (SiP) 8-core processor using CoWoS [49] technology. In detail, each of the two identical chiplets has four Arm processors operating at 4.0 GHz. The chiplets communicate with each other through the CoWoS interposer using low-voltage-in-package-interconnect (LIPINCON), a kind of inter-chiplet parallel interface that offers 1.6 Tb/s/mm^2 bandwidth density at 0.56 pJ/bit energy efficiency [12]. It is a significant improvement over typical 2D systems, as a comparison, the energy efficiency in “Zeppelin” is 2 pJ/bit [48]. Similarly, Apple’s M1 Ultra is realized by connecting two M1 Max die with Apple’s UltraFusion packaging architecture [50], as shown in Figure 3(b) [50]. The UltraFusion uses a local silicon interconnect (LSI) that connects the chips and provides low latency 2.5 TByte/s inter-processor bandwidth. In addition, silicon bridges are also used in some designs. Intel’s Xeon processor Sapphire Rapids (SPR) [11, 51], shown in Figure 3(c) [51], consists of 4 dies, arranged in a 2×2 matrix, connected by 10 embedded multi-die interconnect bridges (EMIB) [52]. In particular, each die is built by 6×4 modular components including core/last level cache (LLC), MC, IO, or accelerator complex. The on-die coherent fabric (CF) is used to provide low latency, high bandwidth communication for the modular components, while the multi-die fabric IO (MDFIO) is introduced for high-bandwidth, low-latency, low-power (0.5 pJ/b) die-to-die interconnect [11].

Analysis. In this scenario, each homogeneous chiplet is functionally complete and independent. Therefore, it is possible to assemble products in a single-chip form first and then in a multi-chip form according to the R&D schedule, as shown in the previous case of Zeppelin [48]. This facilitates the TTM of products with varying performance levels.

However, this case of splitting the SoC into homogeneous chiplets also presents some challenges. The primary challenge is determining the optimal system split method and defining the appropriate splitting interface. One major issue in system splitting is the potential performance degradation that may result from the split. For the high performance multicore processors, such as the chiplet designs mentioned above [9, 11, 12, 48, 50], a multi-level cache architecture with shared LLC is typically used [53]. And the SoC disaggregation interface is LLC, which requires high bandwidth but is relatively latency insensitive.

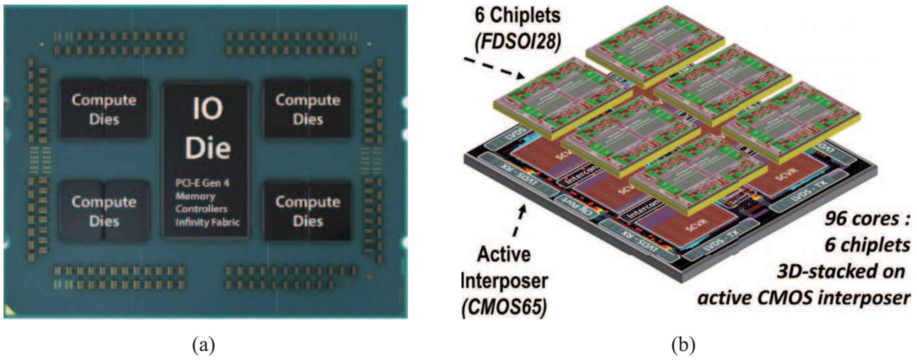


Figure 4 (Color online) SoC disaggregation. (a) AMD “Rome” processor [10] Copyright 2020 IEEE; (b) IntAct architecture [58] Copyright 2021 IEEE.

Moreover, designers are also trying their best to reduce the latency of cross-die communication, such as the above mentioned TSMC’s LIPINCON and Intel’s MDFIO, which are made possible by the dense fine pitch interconnects offered by silicon interposer [12] and silicon bridge [52], respectively. Intel also employed a metal stack with a 400 nm-pitch routing layer optimized for global interconnects, which reduces latency by 30% at the same signal density [11].

Another challenge is that the inconsistent latency of inter-chiplet accesses can lead to non-uniform memory access (NUMA), which is unfriendly to programmers as it requires complex logic for proper placement of memory [54]. Additionally, some memory requests must be serviced by inter-chiplet “remote” memory channels, which can lead to system throughput bottlenecks [3]. For instance, the latency of the inter-chiplet is 56.7% higher than that of the intra-chiplet in the AMD EPYC$^{\mathrm{TM}}$ processor [3]. To mitigate this challenge, a centralized interconnect architecture can be adopted, which will be mentioned below.

3.1.3 *SoC disaggregation with compute/fabric partitioning*

Some studies are based on 2D/2.5D packages, where the fabric die is connected to other compute or memory chiplets via an organic substrate or a passive silicon interposer. For example, the AMD Zen 2 processor [8] utilizes a fabric die, known as the IOD, as the system interconnect center. AMD “Rome” server processor with Zen 2 chiplets is shown in Figure 4(a) [10], eight core-complex dies (CCDs) are interconnected by the IOD, which also provides connectivity to external DDR memory and IOs (e.g., PCIe and universal serial bus (USB)). This provides a more balanced memory access latency, as each CCD can take a direct hop to the IOD, and the target MC is located inside the IOD. As a result, it shows a 61% reduction in access latency difference compared to the Zen 1 processor [3]. In another study [22], researchers propose the Rocket-64 architecture which employs an NoC chiplet as the interconnect hub to achieve the interconnection of eight octa-core reduced instruction set computer (RISC)-V Rocket [55] chiplets and a 4-channel MC chiplet, totaling 12 nodes. Unlike [8], it further splits the MC from the fabric die to form an MC chiplet, providing more design flexibility to use different process nodes [22]. Moreover, additional connections can be provided through the combination of fabric dies. For example, NVIDIA introduces a dedicated GPU-bridging chip called NVSwitch [56]. Each NVSwitch provides 64 ports of NVLink [57] links to accelerate multi-GPU connectivity, and a combined NVSwitch system can fully interconnect 256 GPUs.

For studies based on 3D packages, the fabric die serves as a platform for the top compute die. The below fabric die is also referred to as the base die [27] or active interposer [58]. In Lakefield processors, a compute die and a base die are stacked vertically face-to-face with a µbump array. The two dies communicate with a low-power inter-die IO called the foveros die interface (FDI). The base die contains the common basic functions, including storage, media, and PCIe [27]. Moreover, the base die can be used to achieve interconnectivity. IntAct [58] demonstrates the use of an active interposer to interconnect multi-level caches of an HPC system, as shown in Figure 4(b) [58]. For short-distance and low-latency L1-L2 cache interconnections, passive metal wire direct interconnection is utilized. For medium-distance, latency-sensitive L2-L3 cache interconnections, active links of asynchronous pipelines are employed. And for long-distance, latency relatively insensitive but high-bandwidth L3 cache and off-chip DRAM memory interconnections, interconnection is achieved through the NoC in the active interposer. In addition to

https://engine.scichina.com/doi/10.1007/s11432-023-3926-8

the fabric, the active interposer can also implement modules such as power management [22, 58–60], memory-IO controller [59, 61, 62], on-chip memory [58], and capacitors [58, 61, 62], in turn improving the utilization.

Analysis. This architecture, which separates the interconnect fabric including those low performance common components of systems from the computation cores, offers several advantages. Firstly, centralizing interconnections through the interconnect die leads to increased consistency in inter-chiplet communication latency and reduced NUMA effect, thereby improving the overall performance of multicore processors [3]. Secondly, the fabric die and the compute die can be fabricated in different process nodes, conserving the advanced process chip area and thereby reducing system cost. For example, the "Rome" product consists of a 12 nm IOD and eight 7 nm compute die [10]. As shown in [3], splitting the fabric improves the utilization of advanced process nodes, as the scalable logic (CPU and cache) increases to 86% of the total area (56% for the former). Thirdly, it simplifies the compute die's logic to separate interconnect from computation, because the routing logic is moved into the fabric die. At the same time, without an NoC router, the range of application of the compute dies becomes wider because the router design is more tightly related to the design of the network scheme. Lastly, the fabric die could simplify the package substrate design. For example, the first-generation EPYC$^{\mathrm{TM}}$ processor with four chiplets has exhausted all the routing resources of the substrate, making it difficult to add more chiplets. In comparison, in the second-generation EPYC$^{\mathrm{TM}}$, the four "inner" CCDs are connected by the central fabric die (i.e., IOD). Moreover, another four CCDs could connect to the fabric die by routing paths underneath the "inner" CCDs. Thanks to the fabric die, the number of fully connected chiplets is doubled to 8 [3].

To compare the systems that use 2D/2.5D packaging and 3D packaging, the former has more of a cost advantage, while the latter has more of a performance advantage. Compared to 2D/2.5D packaging, a 3D package offers shorter interconnection distances and a higher interconnect density since the top die and the base die are connected by μbumps or hybrid bonding. The base die enables the relay of signals over long distances through active repeaters, leading to reduced signal delay. However, the base die used in 3D systems is required to be large enough to accommodate all the top dies, and hence it still faces the limit in the aspect of reticle size and yield. To address this, the cascade expansion of the base die technique, Co-EMIB, has been proposed in [28], which is an extension combining EMIB and Foveros technologies.

3.1.4 *SoC disaggregation with multiprocessor partitioning*

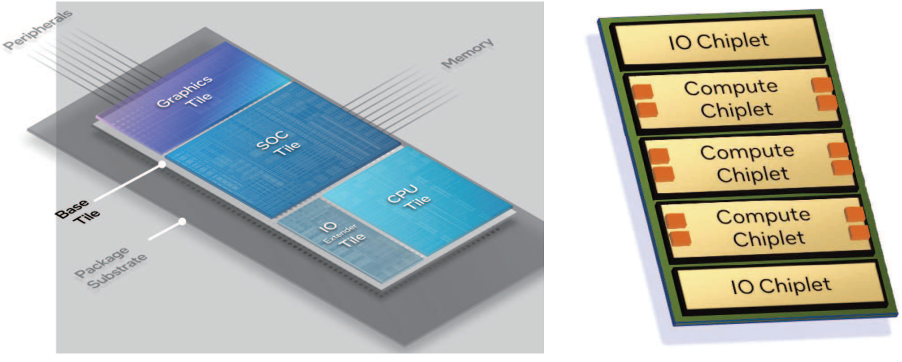
In the case of MPSoCs, they are disaggregated into multiple heterogeneous processors based on functional units. Intel's Meteor Lake processor adopts a "new flexible tiled architecture" [59]. As shown in Figure 5(a) [59], four heterogeneous die (GPU tile, SoC tile, IO tile, and CPU tile) are connected by the base tile. Moreover, each tile is "scalable", such as the core counts, process nodes, and cache amounts that can be scaled on the CPU tile.

Intel recently introduced the Xeon scalable platform in [63], which provides a range of core count products that scale with the number of compute dies present. The "Granite Rapids" processor from Intel's Xeon scalable series, as shown in Figure 5(b) [63], employs a disaggregated design with five chiplets: three compute chiplets with performance cores (P-core) and two IO chiplets with DDR and IO. The compute chiplets can scale up to 8, which delivers scalable performance.

Analysis. Meteor Lake is considered the next step in the "disaggregation journey" [59], which implements the compute, graphics, media, and IO partitioning. This approach provides a flexible way to build a differentiated product family to address different applications and needs. Splitting the SoC and reusing the chiplets in multiple products is considered a complex challenge, yet an important design trend [21, 64, 65].

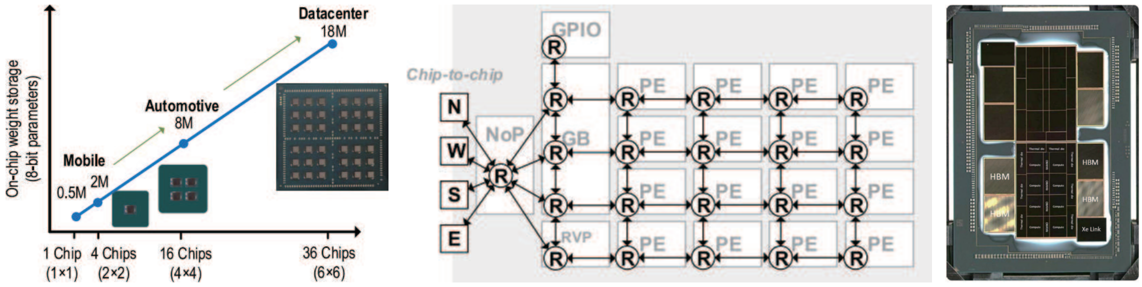
3.1.5 *Tile-based distributed architecture*

NVIDIA introduces a chiplet-based deep neural network (DNN) accelerator in publications [66, 67], as shown in Figures 6(a) and (b). This accelerator is composed of 36 chips connected in a mesh network on a multi-chip-module (MCM). The communication architecture is hierarchical and comprises an NoC and a network-on-package (NoP). Each chip contains an NoP router, a global buffer, an RISC-V processor and 16 processing elements (PEs) connected via a mesh NoC. Moreover, the NoP router transfers packets between the NoC and neighboring chips via on-package links. This tile-based architecture enables flexible scaling, enabling efficient inference on a wide range of DNNs, from mobile to data center domains [67].



(a) (b)

Figure 5 (Color online) Multiprocessor partitioning. (a) Meteor Lake [59] Copyright 2022 IEEE; (b) Granite Rapids [63] Copyright 2023 IEEE.



(a) (b) (c)

Figure 6 (Color online) Tile-based architecture. (a) NVIDIA DNN accelerator and (b) their scalable NoP and NoC architecture [67] Copyright 2020 IEEE; (c) Intel's Ponte Vecchio [62] Copyright 2022 IEEE.

Another example is Tianjic AI chip proposed in [68], it presents a multi-grained scalable architecture similar to [66]. It can be configured with fine-grained integration, with 1×4 chips, for embedded systems, and coarse-grained integration, with a 5×5 chips board design for a cloud platform.

Another example is Intel's PVC, as shown in Figure 6(c) [61,62], which is a processor comprising 47 functional tiles. PVC is composed of sixteen compute tiles, and eight memory tiles optimized for random access bandwidth-optimized SRAM (RAMBO) 3D stacked on the Foveros base dies. The compute tiles and RAMBO tiles are connected by the base tile using the die-to-die FDI link, enabling a scalable compute and memory architecture [62].

Analysis. This architecture provides a flexible and scalable solution for parallel processing, which is a promising approach for building large-scale systems [69]. Moreover, each chip in the system can operate as a standalone subsystem for smaller workloads; meanwhile, by scaling up the number of parallel computing units, the system's computing power can be improved. Besides, in order to reduce data exchange frequency and simplify interconnect design, each tile uses a data buffer to store inter-tile data exchanges.

The main challenge of this architecture is the potential off-chip bandwidth wall, in which bandwidth limitation becomes a bottleneck that limits multicore scaling [70]. Since the memory bandwidth is shared between the compute tiles, as the number of compute tiles increases, the performance will be limited by the memory bandwidth [70]. Apparently, it is a common concern for many-core systems [69, 71]. To alleviate this problem, one way is to integrate more memory into the package, such as by adopting HBM technology [40,72]. Another way is to minimize external memory access and place all data on-chip, such as by using the PIM architecture mentioned above [73–75]. Additionally, the increasing hop count for those packets traversing between the chiplets [76] also limits system scaling. As a result, highly-parallel memory architectures are emerging to alleviate the traffic congestion. It involves distributing the memory access nodes near the compute nodes, so they can be accessed in high parallel with high throughput. Several published designs, such as the distributed MC placement adopted by [71], the distributed tensor direct memory access (DMA) engines introduced in [77], and the distributed RAMBO cache introduced in [62], have demonstrated significant improvements in system performance.

Another issue to consider is redundancy, which brings cost and power overhead. In a tiled chiplet-

Table 1 Comparison of chiplet-based system architecture

		Interconnect efficiency	Scalability	Flexibility	Design effort	Special issues
Package aggregation		Low	Limited scalable	Limited	Low, as minimal design modification	Limited performance
SoC disaggregation	Homogeneous chiplets with integrated fabric	Medium	Limited scalable	Flexible	Low	Limited scalable, NUMA effect
	Compute/fabric partitioning	Medium	Scalable due to fabric die, but is limited by available fabric die size	Compute and fabric flexible using different process nodes	Medium, design extra fabric die	Extra fabric die, yield issue
	Multiprocessor partitioning	High	Scalable performance	Flexible to build differentiated product	High, design multiple heterogeneous die	More complex in SoC splitting and chiplet reusing
	Distributed tile-based architecture	Medium	High scalable structure for parallel processer	Potential flexible to heterogeneous die	Low, increase with heterogeneous chiplet types	Off-chip bandwidth limitation

based system, typically, each chiplet integrates a router, physical channels, and 4 to 6 physical interfaces. And these circuit overheads are for everyone. When the system consists of only a few chiplets, the chiplets located at the edges have inactive interfaces and account for a high proportion, which results in redundancy. In addition, the physical interfaces usually do not scale well with logic shrinking [6], which can diminish the advantages of advanced process nodes for the entire chip.

3.1.6 *Summary of chiplet-based system architecture*

The comparison between various chiplet-based system architectures is shown in Table 1. It highlights the advantages and drawbacks of each architecture.

3.1.7 *Special discussion: the memory subsystem*

Thanks to advanced packaging technologies, chiplet-based designs could achieve larger scale chip than ever before. Furthermore, the growing demand for memory volume and bandwidth for computation in large scale systems has led to a design revolution in memory design. In addition to the above-mentioned HBM technique, cache chiplets have appeared in some products. One example is the RAMBO cache, which is a kind of separated cache dies that are distributed among the individual compute dies as additional L3 cache nodes [62]. Another example is the 3D V-cache technique, which provides a flexible way to scale the cache size of compute chiplet. AMD attaches an additional 3D V-Cache die to the base die via copper-to-copper hybrid bonding, thus tripling the L3 capacity in each CCX [78, 79]. Both of the above techniques expand L3 cache size through advanced packaging with minimal latency overhead.

The utilization of DRAM also benefits from advanced packaging technologies, of which 3D stacking technology has received a great deal of attention. Specifically, a straightforward way is to stack multiple DRAM dies on top of each other. This approach is adopted by HBM. However, a more aggressive yet attractive approach is to stack a DRAM die directly on another logic die. It is called "true" 3D DRAM [80] or "monolithic" 3D DRAM [81]. For example, Park et al. [82] reports that the hybrid bonding process can be applied to DRAM, where a fabricated DRAM wafer could be 3D stacked on another bare silicon wafer by wafer-to-wafer hybrid bonding. Significantly, the monolithic 3D DRAM approach can take full advantage of the die-to-die bandwidth provided by 3D stacking, thus minimizing the performance gap between processor and off-chip DRAM, which is known as the memory wall problem. In addition, this architecture uses a highly parallel way of organizing memory and has also facilitated a revolution in memory architecture design [80].

3.1.8 *Special discussion: platform on package and reusable substrate*

Lowering the overall portfolio cost with a TTM advantage is a compelling driver for the deployment of chiplets [83]. In addition, a scalable and tailorable chiplet-based system platform, which is named as "platform on a package" in [83], can facilitate the development of bespoke solutions based on reusable chiplets agilely [84]. Consequently, some entities have announced the reusable platform schemes.

The fabric of a system, in the form of a standalone fabric die in 2D/2.5D systems or an active interposer with interconnect fabric in 3D systems, usually is the kernel of the platform because the fabric die is the central part implementing interoperability between the chiplets. A well-designed fabric die can be applied to connect different sets of chiplets. Therefore, most of the platform on a package is fabric-centered. More importantly, the design and manufacture of fabric dies also introduce NRE cost and time overhead. If the fabric die can be reused, it can help mitigate costs and shorten the development cycle [85].

The standalone fabric die can be applied to the OCME (one center multiple extensions) reuse scheme proposed in [7], which shows a cost advantage in heterogeneous integration. In addition, fabric dies in

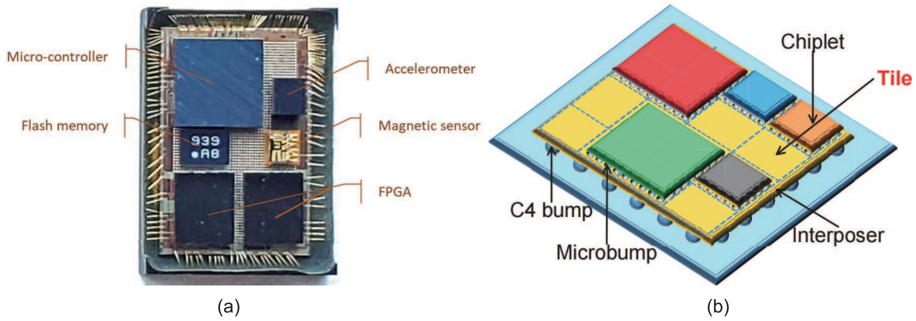


Figure 7 (Color online) (a) Chiplet-based SoC by zGlue [86] Copyright 2021 IEEE; (b) chiplet-based system of GIA [85].

the form of active interposers can be applied to the FSMC (sockets multiple collocations) reuse scheme, although it is based on the 2D/2.5D package. The FSMC reuse scheme shows that chiplets with the same footprint can be integrated into a package with several standardized sockets. By combining with different chips, the NRE cost of the interposer can be amortized to negligible [7].

Moreover, if the substrate or interposer could be reconfigured rapidly, this would accelerate the development of SoC. A chiplet-based SoC architecture is proposed in [86], in which chiplets with different functions such as micro-controllers (MCUs), sensors, and FPGAs are connected by an active silicon interposer, as shown in Figure 7(a) [86]. The active silicon interposer, which is called "zGlue Smart Fabric" in [87], provides programable connectivity that allows rapid reconfiguration of the chiplet interconnection, thus it can be reused in multiple products. A more fine-grained approach called general interposer architecture (GIA) is proposed in [85], in which the chiplets and interposers are divided into multiple tiles with the same size, as shown in Figure 7(b) [85]. Each tile contains a group of predefined connections via µbumps. With this modular design, a chiplet can plug into the free tiles of the interposer [85]. To improve the efficiency of the configuration of the interconnect substrates generation process, the paper further introduced an integration flow with the proposed design automation framework. However, the additional reconfigurable logic leads to power and area overhead, and a yield loss for the defect-prone active interposer [85].

As presented in the above platforms, a standard interface is necessary for a reusable platform, and an industry standard interconnect called universal chiplet interconnect express (UCIe) has been proposed by the UCIe consortium [83]. This approach offers high-performance, low-latency, power-efficient, and cost-effective connectivity between heterogeneous chiplets, which enables designers to package chiplets from different sources, including different fabs, using a wide range of packaging technologies.

3.1.9 *Special discussion: architecture design automation*

As there are so many factors to consider during architecture design, special automation design tools, including tools for design space exploration and architectural simulation, are required. Kim et al. [22] presented a design flow for building and simulating heterogeneous 2.5D designs, which covers and fully automates the entire design phases of architecture, circuit, and package. Their other work [88] proposes a holistic and in-context chiplet-package co-design flow for interposer-based 2.5D systems. The flow includes co-analysis of the power delivery network (PDN) in chiplets and interposers, which helps to optimize the IR-drop with a 27.17% reduction. Moreover, in [89], the researchers discuss the significant coupling between chiplet and package routing wires in experimental designs, which can impact system performance. By properly planning, parasitic extracting, analyzing, and iteratively optimizing the design, the performance gap between the 2D chip and 2.5D system was reduced by 62.5%. In their other work [90], they adopt a similar approach to the RDL-based 2.5D system. To evaluate the performance of chiplet-based DNN architectures and enable efficient design space exploration, a chiplet-based in-memory computing (IMC) architecture simulator called "SIAM" is proposed in [74].

3.2 Interconnection

The chiplet-based interconnections have received significant attention, and the following topics have been extensively investigated.

Topology. The physical layout and connections between nodes and channels are determined by the topology of the fabric, also known as the on-chip interconnect network [91]. Moreover, the selection of the fabric topology in chiplet-based design has a significant impact on overall performance. In particular, the fabric design is slightly different in 2D/2.5D systems (as a standalone fabric die) and in 3D systems (as an active interposer with fabric).

Routing. According to research [23], chiplets need to be designed in a modular manner without considering the holistic system knowledge, so that they can be reused in different systems. However, such a modular design approach presents a significant challenge to the networks, as assembling individually designed chiplets together may lead to deadlocks, even if the NoC of each chiplet is designed to be deadlock-free.

3.2.1 *Topology of the fabric*

In chiplet-based systems with homogeneous chiplets, the chiplets are simply interconnected edge-to-edge. Therefore, the interconnect topology of such systems appears to be relatively straightforward, and we will not analyze it too much. On the other hand, in chiplet-based systems with separated interconnect fabric, all data exchanges between chiplets are realized by the fabric die. The topology of the fabric die determines the overall performance of the system. It should be noted that we focus on the topology of the fabric (interconnect networks) rather than the layout. For example, the EPYC$^{\text{TM}}$ processor with 8 CCDs and an IOD presents a star topology in layout, while the central fabric IOD is in a ring topology [3]. The crossbar provides non-blocking connectivity between any pair of nodes with high-bandwidth and fairly low delay, which is the most classic topology for small-scale systems, though it scales poorly in terms of area and power [91]. In contrast, NoC represents a scalable solution due to its ability to supply scalable bandwidth at low area and power overheads [91]. The ring topology is simple in design, and its naturally ordering properties simplify the implementation of cache coherence [92]. For example, a hybrid ring topology is implemented in the fabric die of the AMD Zen 2 processor, which reduces the memory access latency difference between the nearest and farthest memory channels [3]. However, as the number of interconnected chiplets increases, higher-radix scalable topologies like mesh are preferred. For example, Kim et al. [22, 93] utilized a 4×3 mesh topology in the NoC chiplet to interconnect eight processor chiplets and four channel MCs, totaling 12 nodes.

In tile-based MCM systems, the placement of the chiplet is uniform, and topologies like mesh are therefore suitable [94]. For instance, NVIDIA's scalable DNN accelerator [66] integrates 36 chiplets on a package, connected via a mesh topology. Each die contains 16 PEs connected via a mesh NoC. Similarly, Krishnan et al. [74] has proposed an IMC architecture for deep learning accelerators, which also utilizes NoP interconnections in a mesh topology.

Topologies optimization. Silicon interposers enable the integration of multiple chiplets, providing higher bandwidth and lower power compared to the traditional MCM approach. There are a series of studies centered on silicon interposers [91, 95–99]. To be specific, if the performance of network topologies is considered first, high-radix, high-performance networks are often suggested. However, this kind of network requires long wires and repeaters in physical implementation, which necessitates expensive active interposer technology [95]. Moreover, distributing the networks to the interposer and forming the "active" interposer enables better exploitation of the already "paid-for" interposer [96]. On the other hand, a bottom-up, cost-centric perspective prompts using passive interposers, which limits the performance and length of the link performance because repeaters are unavailable, leading to low-radix, low-performance networks [95].

Nevertheless, active interposers usually face the yield issue because they are expected to have a large area. Accordingly, most studies on active interposer network topologies focus on "minimally active" interposers, with only a small percentage of the active area utilized to minimize yield losses. Firstly, a topology called Double Butterfly is proposed in [96] for the active interposer for a baseline system with a 64-core die and four DRAM stacks around it. The Double Butterfly topology slightly increases the total links and utilities long links, as shown in Figure 8(a) [96]. This topology better exploits the additional routing resources of the silicon interposer, which helps to reduce the network diameter (the longest distance in the network, serves as a proxy of the maximum latency in the topology [91]). The performance improvement comes from the reduced hop count in core-to-memory routing. To address the challenges with respect to the fragmented NoC, researchers proposed a "misaligned" topology called ButterDonut in [47], for the same baseline system as [96]. The key observation of the proposed "misaligned" topology

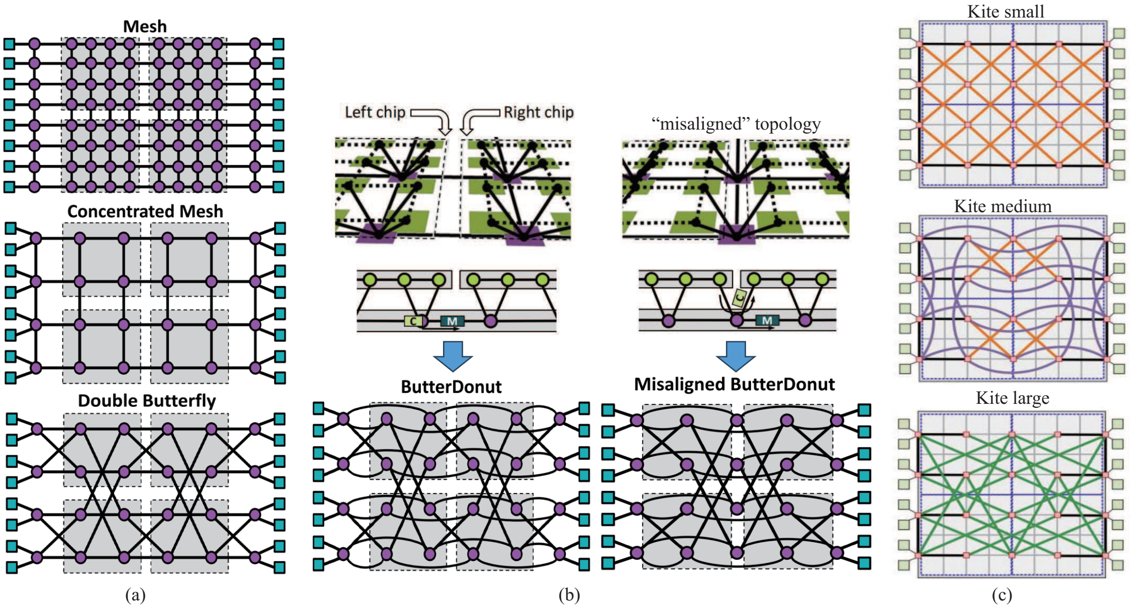


Figure 8 (Color online) Topologies of fabric in silicon interposer. (a) Mesh and Double Butterfly [96] Copyright 2014 ACM; (b) compare of aligned and misaligned topology [47] Copyright 2015 ACM; (c) Kite topology family [97] Copyright 2020 IEEE.

is to place interposer routers "in between" neighboring chiplets instead of "aligned beneath" them, as illustrated in Figure 8(b) [47]. With this change, the routers are shared by neighboring chiplets, so chip-to-chip traffic and on-interposer traffic can flow simultaneously via different router ports without having to compete for the shared links, thereby reducing queuing delays. It is a hybrid topology of both the Double Butterfly and the Folded Torus, without increasing the router complexity or adding additional links (which cause interposer yield loss). Following this lead, researchers proposed the Kite topology family in [97], shown in Figure 8(c) [97], which utilizes longer links more efficiently. This results in a reduced network diameter and higher throughput compared to [47]. It further improved the maximum throughput by 17% versus Double Butterfly and ButterDonut. Furthermore, researchers introduced the ClusCross topology in [98], which treats each chiplet as a cluster and adds cross-cluster long (longer than [97]) links to increase cross-chip bandwidth. It shows a 10% latency reduction compared to the ButterDonut. Contrary to the prior assumptions that the percentage of the active area in an interposer should be minimized, the study [99] demonstrates how the extra "prepaid" silicon area of the interposers can be leveraged for fault tolerance to improve yield and cost-effectiveness. Based on the same baseline as [47, 96–98], the yield loss of a typical 32-core, four-chiplet interposer could be reduced by 0.44× by adding fault-tolerant routers and redundant interconnects.

3.2.2 *Customized topology and reconfigurable topology*

For designs with a wide variety of heterogeneous processors, such as MPSoCs, regular topologies such as mesh or ring as described above may not be appropriate. In these cases, custom topologies or irregular topologies are often prove to be more power-efficient and perform better than standard topologies [91]. The SiPterposer in [100] is a fully passive interconnect structure that allows for custom topology configuration during chiplet assembly. However, its performance is affected by underutilized wire resources. To optimize topologies and routing for specific workloads, application-specific interconnection networks are often used. In a recent study [85], researchers proposed a configurable inter-chiplet network with a mesh topology that utilizes configurable routers and NIs to support customization of topology and routing. More similar studies can be seen in [101, 102].

3.2.3 *Deadlock-free routing*

In chiplet-based systems, each chiplet is independently designed and validated. Although each chiplet is designed to be deadlock-free, new deadlock may still be introduced in the network when different chiplets are assembled together, leading to interconnection failures in chiplet-based systems [23]. However, traditional deadlock-free schemes require a global view of the entire system and prevent independent configuration and optimization of the chiplets, thus violating modularity and reusability [23]. Therefore, routing in modular chiplet design is a key issue in the interconnection network design of chiplets. From the perspective of routing decision implementation, it is generally divided into two routing strategies: distributed and centralized. The following analysis will be conducted respectively.

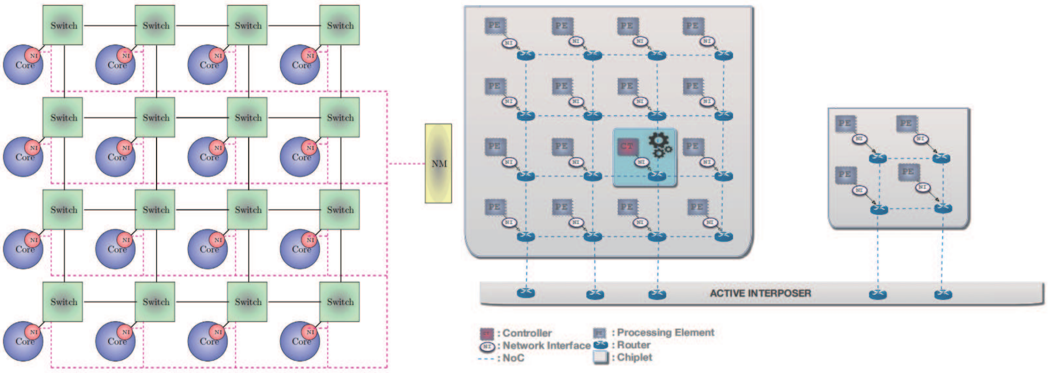
Distributed routing. Distributed routing refers to the independent routing calculation of each node in the network, often using simple and resource-saving routing algorithms, and is commonly found in resource-restricted scenarios such as on-chip networks [91]. Currently, the most widely used routing algorithm in on-chip networks is the simple dimension order routing (DOR) [103]. Moreover, dimension-order routing is a deterministic routing algorithm, which means that all data packets from node A to node B follow the same path. By imposing a direction-of-turn restriction on DOR routing, loops can be avoided in the network. However, as mentioned above, for modularly designed chiplet systems, the use of traditional DOR algorithms may still struggle with deadlock problems.

Yin et al. [23] proposed a modular routing restriction (MTR) that abstracts all external nodes of a chiplet as a single nod, and avoids the formation of deadlock loops by imposing turn restrictions on the boundary routers (BRs). The key point is that through these abstract nodes, the network topology of other chiplets can be ignored, thus obtaining a deadlock-free chiplet network while maintaining modularity. However, the generation of turn restriction tables for BRs could be a time-consuming iterative search process. To speed up this process, a presort turn restriction algorithm (Presort-TRA) is proposed in [104]. Since the objective of the search process is to minimize ϕ (defined as AverageDistance/AverageReachability), the proposed algorithm first presorts the candidates by their ϕ values and then performs the search. The iteration search time is reduced by 50% by the proposed algorithm [104]. Taheri et al. [24] proposed a deadlock-free routing algorithm based on virtual networks (VNs) called DeFT. Unlike [23], the turn restrictions in DeFT are not imposed on specific routers, but on VNs. By imposing turn restrictions on two independent VNs without cyclic dependencies, VN0 and VN1, a deadlock-free network is formed. The deadlock-free routing algorithm based on VNs provides path diversity from source to destination node, and its advantage is that fault tolerance can be achieved through path diversity, which is particularly important for vertically interconnected links that formed during integration and have lower yield. There is a relatively large area and power overhead associated with this VN-based routing since it requires twice as many virtual channel (VC) buffers as [23]. Due to the existence of turn restrictions, the above routing algorithms may limit the choice of paths, potentially cause traffic competition, and ultimately affect performance.

Another way to achieve deadlock-free is based on flow control, which prevents the formation of buffer occupation-request cycles in the network, thus achieving deadlock-free [105]. Majumder et al. [106] proposed a solution called remote control (RC) based on the fact that inter-chiplet networks are guaranteed to be deadlock-free at design time, thus deadlocks are only involved in the inter-chiplet BR, during the high congestion of outbound and inbound packets. RC uses an additional flow control network to store outbound packets in the reserved RC_buffer, thus avoiding potential cyclic dependencies. Compared to turn-restriction-based solutions, flow-control-based solutions can guarantee network path diversity and achieve higher performance in throughput. However, the flow control-based solution will increase latency due to the injection of control, and the additional flow control communication network and the RC_buffer that needs to cache the entire outgoing packet will lead to increased area and power consumption.

Furthermore, there is another approach that allows for deadlocks to occur and periodically recovers from them, called upward packet popup (UPP), proposed in [107]. Based on the same assumptions as [23, 24, 106] that all chiplets are designed to be deadlock-free, the researchers in [107] observe that new occurrence of deadlock will always involve BRs, particularly an upward packet. Therefore, the deadlock in BRs can be detected and broken periodically. According to this observation, the locked packet can be properly routed to its destination using the reserved circuit-switched channels in chiplets, thereby achieving deadlock recovery. Moreover, this detect-and-break strategy avoids the turn restriction of router and control injection [107]. Therefore, this approach can achieve better performance in throughput and latency than the turn restriction-based approach when the traffic load and the probability of deadlocks are low. However, accurately and promptly finding the router where the deadlock occurs is challenging, especially when there are a large number of vulnerable routers. In addition, the deadlock recovery phase will inevitably result in increased latency.

Centralized routing. Centralized routing can obtain the working status of each node in the network, making it easier to implement adaptive routing and thus deadlock-free. For instance, a typical NoC architecture that employs centralized routing is the software-defined NoC (SDNoC) [108–110]. Specifically, the



(a)

(b)

Figure 9 (Color online) (a) SDNoC within monolithic chip [108] Copyright 2017 Elsevier; (b) SDNoC architecture within chiplet [111] Copyright 2019 IEEE.

Table 2 Summary of modular deadlock-free routing

Methods	Deadlock-free routing schematic	Performance		Implementation		Advantage	Drawback
		Path diversity	Latency	Additional buffer	Control network		
MTR [23]	Turn restriction on BRs	Limited	Low	NO	NO	Simple implementation (minimum design change and overhead)	Generating turn restriction tables can be a complex task
DeFT [24]	Turn restriction on VNs	Limited	Lower than [23] due to more VCs	YES, large	NO	Fault tolerance (based on additional VN)	Large implementation overhead
RC [106]	Flow control	Full	Lower than [23] due to path diversity	YES, small	YES	High performance in throughput and latency, adaptive routing	Additional control networks, more complex
UPP [107]	Detect and recover	Full	Lower than [23] due to path diversity	YES, small	NO	No need control network	Vulnerable to false positives
SDNoC [108–111]	Depend on software	Full	High depend on software	NO	YES	Flexible and adaptive routing	May be large overhead (can be simpilified)

control network and the data network are physically separated. And the data network is controlled by a centralized network manager (NM) via the control network [108]. Specifically, the software obtains routing information through the control plane and performs path calculation, thus enabling flexible adaptive routing. Unfortunately, the complex control network introduces communication overhead. Therefore, to reduce communication overhead in chiplet-based systems, as shown in Figure 9(b) [111], a lightweight SDNoC communication protocol, called MicroLET, is proposed in [111]. In particular, the protocol consists of a new simplified message stack specifically developed for micro-scale networks, and three main phases, including handshaking, network monitoring, and routing. Additionally, relying on the centralized management mechanism and the turn model routing algorithm, the system ensures deadlock-freedom.

The comparison of different deadlock-free routings is listed in Table 2 [23, 24, 106–111]. Centralized routing adds complexity to system design because it requires an additional control network for routing and traffic control. Nevertheless, centralized routing has a number of advantages due to its adaptability. For example, adaptive routing can dynamically select the path from the source node to the destination node based on the network's traffic conditions, which is not supported by deterministic routing. In addition, adaptive routing can take advantage of the path diversity from the source node to the destination node to address the issue of network congestion and failures. More importantly, due to its adaptability, centralized routing is more practical than distributed routing in custom topologies or irregular topologies.

3.2.4 *Routing efficiency*

In addition to the deadlock issue of routing, there are also concerns about routing efficiency and robustness. Traditional 2D on-chip networks feature relatively uniform on-chip links and hence similar link delays. Conversely, 3D chiplet systems experience varying link delays between vertical paths (inter-chiplets) and horizontal paths (intra-chiplets). In particular, the horizontal link delay is relatively lower than the vertical link delay in 3D NoCs. Based on this observation, the Chiplet-First routing algorithm for 3D NoC was proposed in [112]. This algorithm extends the DOR routing algorithm from the XY direction to the XYZ direction, and prioritizes routing within the chiplets before transmitting across chiplets, which improves the throughput by 15% on average. Meanwhile, the study [24] focuses on the scenario where vertical links may fail, and fault tolerance is achieved through the diversity of routing paths in the routing algorithm to increase the routing robustness.

3.2.5 *Special discussion: chiplet design automation*

Designing a network of chiplets involves handling several factors, including cost, performance, thermal constraints, and more. This task is difficult to design manually, which is why an automatic network optimization is required. To address this challenge, researchers have proposed a cross-layer co-optimization approach for network design and chiplet placement, as mentioned in publication [95]. The approach takes into account logical, physical, and circuit layers, achieving an energy-efficient network while maximizing performance and minimizing cost, and adhering to thermal constraints. In their other research [113], they take μbump overhead, inter-chiplet circuit designs, and richer topologies into consideration, and improve the cost-performance by 88%.

However, this cross-layer co-optimization has some limitations, the most important being that the designer needs to know the details of the local network of the chiplets, which violates the modularity of chiplets to some degree. On the contrary, when adhering to a modular design approach, each individual chiplet should be designed and verified without any knowledge of the entire system, as mentioned in [23].

4 Discussion

In this study, we have reviewed the three interrelated aspects of package scheme, system architecture, and interconnection design. It is undeniable that cost reduction remains one of the most important drivers for the development of most commercial products. Emerging advanced packaging technologies such as copper-copper hybrid bonding show notable advantages in high-performance interconnects and will be more widely adopted as manufacturing processes mature and costs decrease. This, in turn, will stimulate innovation in system architecture and interconnection design.

In the aspect of system architecture design, the package aggregation approach benefits from the advantages of minimal design effort, and low technical risk, while SoC disaggregation represents a future trend that involves implementing various disaggregation and recombination schemes to obtain a better PPC. SoC disaggregation into homogeneous chiplets with integrated fabric is relatively straightforward partitioning scheme that can further increase the system integration. However, it faces limitations in terms of scalability and inconsistent latency. The separation of interconnect fabric from the compute offers better scalability and flexibility, as well as improved consistency of inter-chiplet latency. Additionally, it helps to preserve advanced process chip area, reduce system costs, and simplify compute and substrate design. In the case of MPSoCs, disaggregating them into multiple heterogeneous tiles (chiplets) based on functional units, not only the types of compute, memory, and fabric die, is a flexible and scalable approach and is a promising approach for building high energy-efficiency systems and achieving product serialization. However, the scale of the system based on fabric or base die is still limited by the size of the fabric or base die. An approach to resolve this limitation is to use a hybrid packaging technique, which connects the base die through 2.5D packaging, such as a silicon bridge. Additionally, the tile-based chiplet architecture also provides a flexible and scalable solution for large-scale parallel processing, but it faces the off-chip bandwidth wall that limits scaling. To address this challenge, state-of-the-art chiplet-based systems employ distributed storage architectures by bringing memory closer to the compute chiplets both horizontally and vertically, and unlocking significant potential bandwidth between memory and compute.

In the aspect of interconnection design, the separation of fabric die facilitates the implementation of multiple topologies. In addition, several studies based on active silicon interposers demonstrate that the performance of interconnects could be improved by making better use of the routing resources on the interposer. When comparing deadlock-free routing schemes, turn restriction-based schemes have the advantage in simple and minimum overhead. On the other hand, flow control and deadlock recovery-based schemes show better performance but tend to be more complex in design. Additionally, centralized routing schemes allow flexibility in achieving deadlock-free routing, especially for irregular topologies. Although centralized routing may introduce higher overhead, this can be minimized by implementing lightweight control networks. Therefore, centralized routing is an attractive approach in chiplet-based design.

According to this survey, chiplet-based systems are currently dominated by homogeneous multicore processors, which are mainly high-performance processors. Splitting a chip into smaller and finer-grained chiplets would significantly improve the yield and it also works for MPSoCs. However, it is not common to see chiplet-based designs in MPSoCs, which may be due to several obstacles such as the packaging and design costs, as well as a lack of off-the-shelf chiplet ecosystem. Nevertheless, these obstacles are

https://engine.scichina.com/doi/10.1007/s11432-023-3926-8

now changing. The differentiation and customization of heterogeneous chip products have become a new driving factor and fine-grained chiplet splitting will prove to be cost-effective. Moreover, with the widespread adoption of chiplet interface standards, more modular chiplet designs will emerge. This will lead to a commercial off-the-shelf chiplets based design ecosystem, which will result in a boom in chiplet-based designs.

In addition, platformization is a significant trend in the evolution of chiplet-based ecosystems. As proposed, the fabric die, which is the core part that provides interconnectivity between chiplets, demonstrates the ability to develop bespoke solutions based on reusable chiplets. Moreover, if the fabric die can be reused, it can help mitigate costs and shorten development cycles, and it promotes a variety of fabric die with reconfigurable topologies, also called active interposer. However, employing a fairly fine-grained reconfigurable active interposer may not be the optimal design choice, as it requires a significant pre-configured redundancy circuit, leading to power and area overheads. Finally, there are many works that use automation tools to analyze or optimize the topology or partition scheme, as well as perform co-optimization across layers. Since these factors impact the PPAC of a chiplet-based system, EDA technologies that perform architecture design space exploration are necessary for chiplet-based design.

5 Conclusion

Chiplet-based design represents a key technology for the post-Moore's law era, offering a cost-effective, high-integrated, and agile design approach. Given the difficulties that such systems must overcome, i.e., choosing the partition schematic, addressing the cross-die communication bottleneck, and ensuring deadlock-free chiplet networks, chiplet-based system design is by no means an easy task. Consequently, a plethora of research papers addressing the challenge have been presented, and a state-of-the-art review can be of great help in summarizing all of the work done to this point.

In this paper, we try to present a survey of the state-of-the-art studies in a classification method and from the perspective of chiplet-based system designers. We have cast the chiplet design challenge as architecture design and interconnect design and subcategorized them separately by design task. Consequently, we have reviewed and analyzed the literature while explicitly elucidating the design principles for the different subcategories. Moreover, we have included a discussion section, where we presented our viewpoint of the chiplet-based design, given the reviewed papers as well as our own experience, and pointed out the current weaknesses and possible future trends.

The primary conclusion is that in the last decade, there has been an enormous research effort in chiplet-based design. However, we are still a long way from a chiplet-based system that better meets all the benefits, such as cost effectiveness, high performance, low power, and agility. It is clear that significant progress has been made in the area of HPC, mainly due to the "disintegration" of the monolithic die by interposers to address the cost and yield issues. Moreover, the research and analysis show that the co-optimization of package, architecture, and interconnect is an important trend and has demonstrated significant improvements in system performance. We have argued that such a responsibility must be shared collaboratively by chip designers, package designers, and EDA vendors, among others. Altogether, we are optimistic about future achievements in this field.

Acknowledgements This work was supported by Shenzhen Science and Technology Program (Grant No. JCYJ20200109143003-935).

References

1 Hennessy J L, Patterson D A. A new golden age for computer architecture. Commun ACM, 2019, 62: 48–60

2 Hao Y, Xiang S Y, Han G Q, et al. Recent progress of integrated circuits and optoelectronic chips. Sci China Inf Sci, 2021, 64: 201401

3 Naffziger S, Beck N, Burd T, et al. Pioneering chiplet technology and design for the AMD EPYC$^{\text{TM}}$ and Ryzen$^{\text{TM}}$ processor families: industrial product. In: Proceedings of the ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), 2021. 57–70

4 EPS I. Chiplet definition-IEEE electronics packaging society. https://eps.ieee.org/technology/definitions.html

5 Nagisetty R. The Path to a Chiplet Ecosystem. Technical Report, ODSA Workshop, 2019

6 Loh G H, Naffziger S, Lepak K. Understanding chiplets today to anticipate future integration opportunities and limits. In: Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021. 142–145

7 Feng Y X, Ma K S. Chiplet actuary: a quantitative cost model and multi-chiplet architecture exploration. In: Proceedings of the 59th ACM/IEEE Design Automation Conference, 2022. 121–126

8 Suggs D, Subramony M, Bouvier D. The AMD "Zen 2" processor. IEEE Micro, 2020, 40: 45–52

9 Beck N, White S, Paraschou M, et al. "Zeppelin": an SoC for multichip architectures. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2018. 40–42

10 Naffziger S, Lepak K, Paraschou M, et al. 2.2 AMD chiplet architecture for high-performance server and desktop products. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2020. 44–45

11 Nassif N, Munch A O, Molnar C L, et al. Sapphire rapids: the next-generation Intel Xeon scalable processor. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2022. 44–46

12 Lin M S, Huang T C, Tsai C C, et al. A 7-nm 4-GHz arm^{1}-core-based CoWoS^{1} chiplet design for high-performance computing. IEEE J Solid-State Circ, 2020, 55: 956–966

13 Lau J H. Chiplet Heterogeneous Integration. Singapore: Springer Singapore, 2021. 413–439

14 Li T, Hou J, Yan J L, et al. Chiplet heterogeneous integration technology-status and challenges. Electronics, 2020, 9: 670

15 Jiang J, Wang Q, He G, et al. Research and prospect on chiplet technology. Microelectron Comput, 2022, 39: 1–6

16 Ma X H, Wang Y, Wang Y J, et al. Survey on chiplets: interface, interconnect and integration methodology. CCF Trans HPC, 2022, 4: 43–52

17 Shan G B, Zheng Y W, Xing C Y, et al. Architecture of computing system based on chiplet. Micromachines, 2022, 13: 205

18 Lau J H. Recent advances and trends in advanced packaging. IEEE Trans Compon Packag Manufact Technol, 2022, 12: 228–252

19 EPS I. Heterogeneous integration roadmap-IEEE electronics packaging society. https://eps.ieee.org/hir

20 Lau J H. State-of-the-art of advanced packaging. In: Proceedings of the Chiplet Design and Heterogeneous Integration Packaging, 2023. 1–99

21 Stow D, Akgun I, Barnes R, et al. Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration. In: Proceedings of the 35th International Conference on Computer-Aided Design, 2016. 1–6

22 Kim J, Murali G, Park H, et al. Architecture, chip, and package co-design flow for 2.5D IC design enabling heterogeneous IP reuse. In: Proceedings of the 56th Annual Design Automation Conference 2019, 2019. 1–6

23 Yin J M, Lin Z F, Kayiran O, et al. Modular routing design for chiplet-based systems. In: Proceedings of the ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), 2018. 726–738

24 Taheri E, Pasricha S, Nikdast M. DeFT: a deadlock-free and fault-tolerant routing algorithm for 2.5D chiplet networks. In: Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2022. 1047–1052

25 Chen W C, Hu C, Ting K C, et al. Wafer level integration of an advanced logic-memory system through 2nd generation CoWoS®technology. In: Proceedings of the Symposium on VLSI Technology, 2017. 54–55

26 Wuu J, Agarwal R, Ciraula M, et al. 3D V-cache: the implementation of a hybrid-bonded 64 MB stacked cache for a 7 nm x86-64 CPU. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2022, 65: 428–429

27 Gomes W, Khushu S, Ingerly D B, et al. 8.1 Lakefield and mobility compute: a 3D stacked 10 nm and 22 FFL hybrid processor system in 12×12 mm^2, 1 mm package-on-package. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2020. 144–146

28 Ingerly D B, Enamul K, Gomes W, et al. Foveros: 3D integration and the use of face-to-face chip stacking for logic devices. In: Proceedings of the IEEE International Electron Devices Meeting (IEDM), 2019

29 Drucker K, Jani D, Agarwal I, et al. The open domain-specific architecture. In: Proceedings of the IEEE Symposium on High-Performance Interconnects (HOTI), 2020. 25–32

30 Qian X H. Graph processing and machine learning architectures with emerging memory technologies: a survey. Sci China Inf Sci, 2021, 64: 160401

31 Zou X Q, Xu S, Chen X M, et al. Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology. Sci China Inf Sci, 2021, 64: 160404

32 Zhao Y L, Yang J L, Li B, et al. NAND-SPIN-based processing-in-MRAM architecture for convolutional neural network acceleration. Sci China Inf Sci, 2023, 66: 142401

33 An J J, Wang L F, Ye W, et al. Design memristor-based computing-in-memory for AI accelerators considering the interplay between devices, circuits, and system. Sci China Inf Sci, 2023, 66: 182404

34 Lee H J, Mahajan R, Sheikh F, et al. Multi-die integration using advanced packaging technologies. In: Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2020. 1–7

35 Kurd N, Chowdhury M, Burton E, et al. 5.9 Haswell: a family of IA 22nm processors. In: Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014. 112–113

36 Hamzaoglu F, Arslan U, Bisnik N, et al. 13.1 A 1 Gb 2 GHz embedded DRAM in 22 nm tri-gate CMOS technology. In: Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014. 230–231

37 Hammarlund P, Martinez A J, Bajwa A A, et al. Haswell: the 4th-generation Intel core processor. IEEE Micro, 2014, 34: 6–20

38 Macri J. AMD's next generation GPU and high bandwidth memory architecture: fury. In: Proceedings of the IEEE Hot Chips 27 Symposium (HCS), 2015. 1–26

39 Lee D U, Kim K W, Kim K W, et al. A 1.2 V 8 Gb 8-Channel 128 GB/s high-bandwidth memory (HBM) stacked DRAM with effective I/O test circuits. IEEE J Solid-State Circ, 2015, 50: 191–203

40 Lee C C, Hung C, Cheung C, et al. An overview of the development of a GPU with integrated HBM on silicon interposer. In: Proceedings of the IEEE 66th Electronic Components and Technology Conference (ECTC), 2016. 1439–1444

41 Liao H, Tu J J, Xia J, et al. DaVinci: a scalable architecture for neural network computing. In: Proceedings of the IEEE Hot Chips 31 Symposium (HCS), 2019. 1–44

42 Kim S, Kim S, Cho K, et al. Processing-in-memory in high bandwidth memory (PIM-HBM) architecture with energy-efficient and low latency channels for high bandwidth system. In: Proceedings of the IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), 2019. 1–3

43 Viswanath R, Chandrasekhar A, Srinivasan S, et al. Heterogeneous SoC integration with EMIB. In: Proceedings of the IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), 2018. 1–3

44 J.Finnegan, Viljoen N. Fourth generation architecture for smartNICs. In: Proceedings of the SmartNICs Summit, 2022

45 Sheffield D. IvyTown Xeon+FPGA: the HARP program. In: Proceedings of the International Symposium on Computer Architecture (ISCA), 2016

46 Hwang R, Kim T, Kwon Y, et al. Centaur: a chiplet-based, hybrid sparse-dense accelerator for personalized recommendations. In: Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), 2020. 968–981

47 Kannan A, Jerger N E, Loh G H. Enabling interposer-based disintegration of multi-core processors. In: Proceedings of the 48th International Symposium on Microarchitecture, 2015. 546–558

48 Burd T, Beck N, White S, et al. "Zeppelin": an SoC for multichip architectures. IEEE J Solid-State Circ, 2019, 54: 133–143

49 Huang P K, Lu C Y, Wei W H, et al. Wafer level system integration of the fifth generation CoWoS®-s with high performance Si interposer at 2500 mm^2. In: Proceedings of the IEEE 71st Electronic Components and Technology Conference (ECTC), 2021. 101–104
50 Mattioli M. Meet the fam1ly. IEEE Micro, 2022, 42: 78–84
51 Biswas A. Sapphire rapids. In: Proceedings of the IEEE Hot Chips 33 Symposium (HCS), Palo Alto, 2021. 1–22
52 Mahajan R, Sankman R, Patel N, et al. Embedded multi-die interconnect bridge (EMIB) – a high density, high bandwidth packaging interconnect. In: Proceedings of the IEEE 66th Electronic Components and Technology Conference (ECTC), 2016. 557–565
53 Mekkat V, Holey A, Yew P C, et al. Managing shared last-level cache in a heterogeneous multicore processor. In: Proceedings of the 22nd International Conference on Parallel Architectures and Compilation Techniques, 2013. 225–234
54 Lameter C. NUMA (non-uniform memory access): an overview. Queue, 2013, 11: 40–51
55 Asanović K, Avizienis R, Bachrach J, et al. The Rocket chip generator. Technical Report UCB/EECS-2016-17, 2016
56 Ishii A, Foley D, Anderson E, et al. Nvswitch and Dgx-2 Nvlink-switching chip and scale-up compute server. In: Proceedings of the Hot Chips, 2018
57 Ishii A, Wells R. The nvlink-network switch: Nvidia's switch chip for high communication-bandwidth superpods. In: Proceedings of the IEEE Hot Chips 34 Symposium (HCS), 2022. 1–23
58 Vivet P, Guthmuller E, Thonnart Y, et al. IntAct: a 96-Core processor with 6 chiplets 3D-Stacked on an active interposer with distributed interconnects and integrated power management. IEEE J Solid-State Circ, 2021, 56: 79–97
59 Gomes W, Morgan S, Phelps B, et al. Meteor lake and Arrow lake Intel next-gen 3D client architecture platform with Foveros. In: Proceedings of the IEEE Hot Chips 34 Symposium (HCS), 2022. 1–40
60 Ilderem V, Pellerano S, Tschanz J, et al. Innovations for intelligent edge. In: Proceedings of the 48th European Solid State Circuits Conference (ESSCIRC), 2022. 41–44
61 Blythe D. $\mathrm{X^e}_{\mathrm{hpc}}$ ponte vecchio. In: Proceedings of the IEEE Hot Chips 33 Symposium (HCS), Palo Alto, 2021. 1–34
62 Gomes W, Koker A, Stover P, et al. Ponte vecchio: a multi-tile 3D stacked processor for exascale computing. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2022. 42–44
63 Gianos C. Architecting for flexibility and value with next gen Intel® Xeon® processors. In: Proceedings of the IEEE Hot Chips 35 Symposium (HCS), 2023. 1–15
64 Munoz R. Furthering Moore's law integration benefits in the chiplet era. IEEE Design Test, 2024, 41: 81–90
65 Ehrett P, Austin T, Bertacco V. Chopin: composing cost-effective custom chips with algorithmic chiplets. In: Proceedings of the 39th International Conference on Computer Design (ICCD), 2021. 395–399
66 Zimmer B, Venkatesan R, Shao Y S, et al. A 0.11 pJ/Op, 0.32–128 TOPS, scalable multi-chip-module-based deep neural network accelerator with ground-reference signaling in 16 nm. In: Proceedings of the Symposium on VLSI Circuits, 2019. 300–301
67 Zimmer B, Venkatesan R, Shao Y S, et al. A 0.32–128 TOPS, scalable multi-chip-module-based deep neural network inference accelerator with ground-referenced signaling in 16 nm. IEEE J Solid-State Circ, 2020, 55: 920–932
68 Pei J, Deng L, Ma C, et al. Multi-grained system integration for hybrid-paradigm brain-inspired computing. Sci China Inf Sci, 2023, 66: 142403
69 Shao Y S, Clemons J, Venkatesan R, et al. SIMBA: scaling deep-learning inference with multi-chip-module-based architecture. In: Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019. 14–27
70 Ahsan B. Off-chip bandwidth for multicore processors: managing the next big wall. Dissertation for Ph.D. Degree. New York: The City University of New York, 2010
71 Jang H, Kim J, Gratz P, et al. Bandwidth-efficient on-chip interconnect designs for GPGPUs. In: Proceedings of the 52nd Annual Design Automation Conference, 2015
72 Park M J, Cho H S, Yun T S, et al. A 192 Gb 12-high 896 GB/s HBM3 DRAM with a TSV auto-calibration scheme and machine-learning-based layout optimization. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2022. 444–446
73 Imani M, Gupta S, Kim Y, et al. Floatpim: in-memory acceleration of deep neural network training with high precision. In: Proceedings of the 46th International Symposium on Computer Architecture, 2019. 802–815
74 Krishnan G, Mandal S K, Pannala M, et al. SIAM: chiplet-based scalable in-memory acceleration with mesh for deep neural networks. ACM Trans Embed Comput Syst, 2021, 20: 1–24
75 Dou C M, Chen W H, Xue C X, et al. Nonvolatile circuits-devices interaction for memory, logic and artificial intelligence. In: Proceedings of the IEEE Symposium on VLSI Technology, 2018. 171–172
76 Zheng H, Wang K, Louri A. A versatile and flexible chiplet-based system design for heterogeneous manycore architectures. In: Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC), 2020. 1–6
77 Jouppi N P, HyunYoon D, Ashcraft M, et al. Ten lessons from three generations shaped Google's TPUv4i: industrial product. In: Proceedings of the ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), 2021. 1–14
78 Burd T, Li W, Pistole J, et al. Zen3: the AMD 2nd-generation 7 nm x86-64 microprocessor core. In: Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), 2022. 1–3
79 Evers M, Barnes L, Clark M. The AMD next-generation "Zen 3" core. IEEE Micro, 2022, 42: 7–12
80 Loh G H. 3D-stacked memory architectures for multi-core processors. In: Proceedings of the International Symposium on Computer Architecture, 2008. 453–464
81 Huang C H, Thakkar I G. Improving the latency-area tradeoffs for DRAM design with coarse-grained monolithic 3D (M3D) integration. In: Proceedings of the IEEE 38th International Conference on Computer Design (ICCD), 2020. 417–420
82 Park J, Lee B, Lee H, et al. Wafer to wafer hybrid bonding for DRAM applications. In: Proceedings of the IEEE 72nd Electronic Components and Technology Conference (ECTC), 2022. 126–129
83 Das Sharma D, Pasdast G, Qian Z, et al. Universal chiplet interconnect express (UCIe): an open industry standard for innovations with chiplets at package level. IEEE Trans Compon Packag Manufact Technol, 2022, 12: 1423–1431
84 Sharma D D. System on a package innovations with universal chiplet interconnect express (UCIe) interconnect. IEEE Micro, 2023, 43: 76–85
85 Li F P, Wang Y, Cheng Y Q, et al. GIA: a reusable general interposer architecture for agile chiplet integration. In: Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design, 2022. 1–9
86 ChanMok M P, Chan C H, ShuiChow W C, et al. Chiplet-based system-on-chip for edge artificial intelligence. In: Proceedings of the 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), 2021. 1–3
87 Nasrullah J, Luo Z Q, Taylor G. Designing software configurable chips and SIPs using chiplets and zGlue. Int Symp

MicroElectron, 2019, 2019: 000027

88 Kim J, Chekuri V C K, Rahman N M, et al. Chiplet/interposer co-design for power delivery network optimization in heterogeneous 2.5-D ICs. IEEE Trans Compon Packag Manufact Technol, 2021, 11: 2148–2157

89 Kabir M A, Petranovic D, Peng Y R. Coupling extraction and optimization for heterogeneous 2.5D chiplet-package co-design. In: Proceedings of the 39th International Conference on Computer-Aided Design, 2020. 1–8

90 Kabir M A, Peng Y. Chiplet-package co-design for 2.5D systems using standard ASIC CAD tools. In: Proceedings of the 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020. 351–356

91 Jerger N E, Krishna T, Peh L S. On-Chip Networks. Synthesis Lectures on Computer Architecture. Cham: Springer International Publishing, 2017

92 Marty M R, Hill M D, Coherence ordering for ring-based chip multiprocessors. In: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06), 2006. 309–320

93 Kim J, Murali G, Park H, et al. Architecture, chip, and package codesign flow for interposer-based 2.5-D chiplet integration enabling heterogeneous IP reuse. IEEE Trans VLSI Syst, 2020, 28: 2424–2437

94 Salihundam P, Jain S, Jacob T, et al. A 2 Tb/s 6×4 mesh network for a single-chip cloud computer with DVFS in 45 nm CMOS. IEEE J Solid-State Circ, 2011, 46: 757–766

95 Coskun A, Eris F, Joshi A, et al. A cross-layer methodology for design and optimization of networks in 2.5D systems. In: Proceedings of the International Conference on Computer-Aided Design, 2018. 1–8

96 Jerger N E, Kannan A, Li Z M, et al. NoC architectures for silicon interposer systems: why pay for more wires when you can get them (from your interposer) for free? In: Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, 2014. 458–470

97 Bharadwaj S, Yin J M, Beckmann B, et al. Kite: a family of heterogeneous interposer topologies enabled via accurate interconnect modeling. In: Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC), 2020. 1–6

98 Shabani H, Guo X C. Cluscross: a new topology for silicon interposer-based network-on-chip. In: Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip, 2019. 1–8

99 Stow D, Xie Y, Siddiqua T, et al. Cost-effective design of scalable high-performance systems using active and passive interposers. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2017. 728–735

100 Ehrett P, Austin T, Bertacco V. Sipterposer: a fault-tolerant substrate for flexible system-in-package design. In: Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2019. 510–515

101 Tan C, Karunaratne M, Mitra T, et al. Stitch: fusible heterogeneous accelerators enmeshed with many-core architecture for wearables. In: Proceedings of the ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), 2018. 575–587

102 Goyal V, Wang X W, Bertacco V, et al. Neksus: an interconnect for heterogeneous system-in-package architectures. In: Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2020. 12–21

103 Glass C J, Ni L M. The turn model for adaptive routing. In: Proceedings of the 19th Annual International Symposium on Computer Architecture, 1992. 278–287

104 Chen C X, Yin J M, Peng Y R, et al. Design challenges of intrachiplet and interchiplet interconnection. IEEE Des Test, 2022, 39: 99–109

105 Duato J, Yalamanchili S, Ni L. Interconnection Networks. New York: Morgan Kaufmann, 2003

106 Majumder P, Kim S, Huang J, et al. Remote control: a simple deadlock avoidance scheme for modular systems-on-chip. IEEE Trans Comput, 2021, 70: 1928–1941

107 Wu Y B, Wang L, Wang X H, et al. Upward packet popup for deadlock freedom in modular chiplet-based systems. In: Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2022. 986–1000

108 Berestizshevsky K, Even G, Fais Y, et al. SDNoC: software defined network on a chip. Microprocess MicroSyst, 2017, 50: 138–153

109 Liu C, Wang W, Wang Z Y. A configurable, programmable and software-defined network on chip. In: Proceedings of the IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), 2014. 813–816

110 Ruaro M, Medina H M, Amory A M, et al. Software-defined networking architecture for NoC-based many-cores. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), 2018. 1–5

111 Ellinidou S, Sharma G, Kontogiannis S, et al. MicroLET: a new SDNoC-based communication protocol for chiplet-based systems. In: Proceedings of the 22nd Euromicro Conference on Digital System Design (DSD), 2019. 61–68

112 Pano V, Kuttappa R, Taskin B. 3D NoCs with active interposer for multi-die systems. In: Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip, 2019. 1–8

113 Coskun A, Eris F, Joshi A, et al. Cross-layer co-optimization of network design and chiplet placement in 2.5-D systems. IEEE Trans Comput-Aided Des Integr Circ Syst, 2020, 39: 5183–5196