

基于芯粒架构的通信拓扑结构研究：进展与挑战

张佳帅^{1,2} 杨柳青^{3,4} 付琦麟^{3,4} 程慧武^{1,2} 邵翠萍^{1,2,3,4} 李慧云^{*1,2,3,4}

1 中国科学院大学 北京市 100049

2 中国科学院深圳先进技术研究院 广东省 深圳市 518055

3 深圳理工大学算力微电子学院 广东省 深圳市 518107

4 广东省算力微电子重点实验室 广东省 深圳市 518107

(js.zhang@siat.ac.cn)

摘要 基于芯粒(Chiplet)的多芯片集成设计为超越传统 System on Chip (SoC) 单片集成提供了灵活且可扩展的解决方案。然而, Chiplet 间的通信已成为影响整体性能的关键瓶颈。在这一背景下, Network on Interposer (NoI) 在多芯片系统中扮演着至关重要的作用, 直接影响着系统的性能和开发成本。本文回顾了 Chiplet 的 NoI 通信拓扑结构, 深入探讨了当前 Chiplet 间通信架构的设计与实现方法。文章涵盖了从协议层、接口层到应用层的完整通信过程, 并基于互连拓扑的形状进行分类, 对每个类别进行详细分析和交叉比较。此外, 文章还探讨了未来芯片间通信技术的发展方向, 强调了技术挑战和潜在解决方案, 并重点分析总结了基于工作负载导向的可重用中介层与拓扑设计的重要性, 旨在为研究人员提供对 NoI 技术现状梳理及未来趋势的展望。

关键字: 芯粒; 拓扑; 硅中介层; 中介层上网络; 片上网络

中图法分类号 TP303

Research on Communication Topologies for Chiplet Architecture: Progress and Challenges

ZHANG Jiashuai^{1,2} YANG Liuqing^{3,4} FU Qilin^{3,4} CHENG Huiwu^{1,2} SHAO Cuiping^{1,2,3,4} and LI Huiyun^{*1,2,3,4}

1 University of Chinese Academy of Sciences, Beijing 101408, China

2 Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

3 Faculty of Computility Microelectronics, Shenzhen University of Advanced Technology, Shenzhen 518107, China

4 Guangdong Provincial Key Laboratory of Computility Microelectronics, Shenzhen 518107, China

Abstract Chiplet-based multi-chip integration designs provide a flexible and scalable solution that surpasses traditional SoC (System on Chip) monolithic integration. However, inter-chiplet communication has become a significant bottleneck affecting overall system performance. The Network on Interposer (NoI) plays a pivotal role in multi-chip systems, directly influencing both performance and development costs. This paper reviews the communication topologies of Chiplet-based NoI structures and delves into the design and implementation methods of current inter-chiplet communication architectures. It comprehensively covers the communication process from protocol, interface, to application layers, classifying interconnect topologies based on structural configurations, and providing in-depth analyses and cross-comparisons for each category. Additionally, this paper explores the future directions of inter-chiplet communication technologies, emphasizing technical challenges and potential solutions, and highlights the importance of workload-oriented, reusable interposer layers and topology design. This review aims to provide researchers with a clear overview of the current state and future trends in NoI technology, emphasizing its key role in advancing next-generation semiconductor devices across a wide range of applications.

Keywords Chiplet, Topology, Silicon Interposer, Network on Interposer(NoI), Network on Chip(NoC)

到稿日期:

返修日期:

通信作者:

1 引言

在半导体设计领域的演进中，系统级芯片（SoC）架构长久以来被视为集成度和性能优化的典范。SoC 通过将处理器、存储、输入/输出功能以及其他必要的系统组件集成于单一的硅片上，为电子系统提供了一种高效且紧凑的解决方案（如图 1（a）所示）。这种集成方法在早期的技术发展阶段极大地推动了移动计算和智能设备的普及，并带来了最佳的芯片质量和能效。然而，随着制程技术的进步和设计复杂性的增加，SoC 架构面临着良率降低、成本上升和带宽限制等问题，难以以高效益实现高集成度和极致性能。与传统的 SoC 集成方法相比，Chiplet 技术提供了一种新的模块化范式，作为当前 Application-Specific Integrated Circuit（ASIC）和 Multi-Chip Module（MCM）集成技术之间的一种折中方案。表 1 展示了这三种集成方法的相关参数对比。

Chiplet 技术将特定功能分割成独立的可重用

表 1 不同集成技术的设计周期、成本和特性对比

Table 1 Comparison of design cycles, costs and features of different integration technologies

Integration Technology	Perspectives									
	Design Cycle	Cost/\$	Integration	Energy Efficiency	Performance	Area	Flexibility	Scalability	Reusability	Design Complexity
Monolithic										
ASIC	About 18 months	>1,000,000	+++	+	++++	+	-	-	-	++++
Chiplet	About 12 months	1,000-1,000,000	++	++	+++	+++	+++	+++	+++	++
MCM	About 12 months	500,000-5,000,000	+++	++	++++	++	++	++	++	++++

然而，Chiplet 架构引入了其自身的一系列挑战，特别是在 Chiplet 间通信的集成度、性能、能耗以及面积方面。传统的基于总线的通信结构，包括单总线、多重总线、分层总线、时间分割多路复用总线、环形总线和交叉开关总线等，虽然避免了路由节点的开销，但随着芯片数量的增加变得效率低下、能耗高、扩展性差，并且只支持边到边的通信结构，不适用于多个芯片之间的通信。因此，Chiplet 间的通信逐渐向使用路由节点的 NoI 方法转变，这种转变不仅提供了更高的数据传输效率和更低的能耗，还支持多 Chiplet 系统中更复杂的数据管理和处理功能有效提升了处理器的性能和能效比、缩短了认证周期，为 5G 网络设备、自动驾驶系统、数据中心和云计算等多个领域提供了强有力的支撑。

小型芯片单元，并通过高速互连技术将它们紧密集成在中介层上（如图 1（b）所示），且不会显著降低性能（小于 5%）[1]，被认为是推动现代半导体行业发展的关键技术之一。Chiplet 技术通过模块化设计，不仅降低了设计和制造成本，还提高了系统的可扩展性和生产效率，使得敏捷定制化解决方案成为可能[2][3]。这一技术的引入，促进了半导体产业的创新，还为解决摩尔定律放缓以及登纳德缩放定律终结所带来的挑战提供了新的途径。

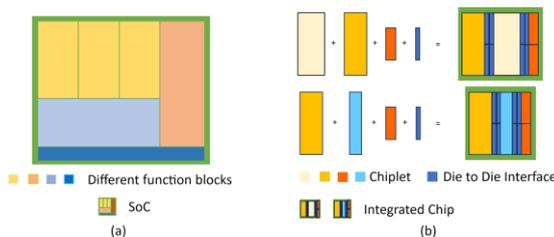


图 1 SoC 集成技术 (a) Chiplet 集成技术 (b)

Fig. 1 SoC integration technology (a) Chiplet integration technology (b)

随着芯片密度的增加和计算需求的不断提升，Chiplet 间的通信拓扑研究显得尤为重要。作为通信结构的核心组件，中介层可根据需求连接任意数量的 Chiplet，通过深入研究 Chiplet 的通信拓扑结构，研究者们可以更好地理解和解决多芯片系统中的集成、通信和可重用挑战。这对于推动半导体行业的技术进步和满足高性能计算需求具有非常重要的意义。此外，Chiplet 通信拓扑的研究不仅关乎技术创新，还涉及经济效益和环境影响，促使可重构及可重用 Chiplet 以及中介层的发展，使其成为半导体设计和制造领域中一个极为重要的研究主题。

现有研究大多关注在互联结构和 NoC 技术的发展上，部分研究对多 Chiplet 互联所面临的带宽和延迟挑战进行了讨论，但对 Chiplet 整体且全面

表 2 基于 Chiplet 的代表性通信架构比较

Table 2 Comparison of Representative Chiplet-based Communication Architectures

Paper Year	Name	Topology	Workload	Deadlock Free Method	Evaluation Tool	Results
arXiv 2024[44]	DCRA	Hierarchical Torus	Irregular Applications, Hybrid	\	Built itself evaluation framework on top of the functional simulator from dalorex	Latency reduction: up to 8x, energy efficiency: 92% of mesh, but hierarchical torus improves efficiency by 19%, cost efficiency improvement
ICCAD 2023[35]	Monad	Ring	Data-intensive	\	Utilize TENET to analyze data reuse, and accelergy to estimate area and energy	EDP reduction: 16% compared to simba, 30% compared to nn-baton
IEEE ISCA 2023[34]	NN-baton	Ring	Data-intensive	\	Simulation (gem5, nvsim), real-world deployment, topology setup (booksim)	Latency reduction: 30%, throughput improvement: 40% compared to baseline mesh
IEEE TCOMPUT AID D 2022[46]	SWAP	Irregular	Large-scale DL Workloads, Hybrid	Multi objective optimization (moo) algorithm	Simulation (booksim)	Energy reduction: 42.4%, latency reduction: 10.6%, fabrication cost lower 10.6x compared to state-of-the-art noi
IEEE DAC 2022[5]	Kite	ButterDonut	Compute-intensive	Shortest path routing combined with escap virtual channel	Simulation (desnt.heterogarnet)	Latency reduction: 7.5%, throughput improvement: 10% compared to baseline mesh, up to 17%
IEEE JSSC 2021[39]	IntAct	mesh	HPC, AI Applications, Data-intensive	\	Designed the corresponding evaluation development board	Bandwidth: 3-tbit/s/mm ² latency: 0.6-ns/mm
IEEE Communications of the ACM 2021[40]	Simba	Hierarchical Mesh	DNN Inference, Compute-intensive,	Hybrid wormhole, cut-through flow control	Attached to an x86 host through PCI-E using a xilinx FPGA	Energy efficiency improvement: 5.4x compared to V100, 2.9x compared to T4, throughput improvement: 1.8x compared to V100, 1.9x compared to T4
IEEE ICCAD 2021[47]	GIA	Irregular	General Purpose, Hybrid	Configurable router, virtual channel allocation	Simulation (booksim, dsent, ltspace, hotspot, tgf)	Performance boost: 3.15x (active), 60.92x (passive), power saving: 2.57x (active), 2.99x (passive) compared to baselines
IEEE DAC 2021[41]	NoI Design for Agile NN Processor Customization	Mesh + Fat Tree	AI Workloads (NNP), Hybrid	Dimensional-ordered	Simulation (scale-sim, booksim2.0, orion3.0)	Latency reduction: 42.7%-79.5%, area overhead reduction: 26.4% compared to baseline mesh (using a dimensional-ordered routing algorithm, the IO die is placed in the center)
IEEE DAC 2021[22]	HexaMesh	Hexagonal Mesh	Large-scale Chiplet Systems, Hybrid	\	Simulation (booksim2.0)	Latency reduction: 19%, diameter reduction 42%, throughput improvement: 34% compared to baseline 2D mesh
IEEE HPCA 2023[42]	Scalable Method for Efficient Chiplet Interconnect Design	2D High-radix Mesh	Multi-chiplet Systems, Hybrid	A minus-first-based deadlock-free adaptive routing algorithm and a safe/unsafe flow control policy	Simulation (a chiplet-specific cycle-accurate C++ simulator)	Bandwidth usage improvement: 20%, energy reduction up to 60%, latency reduction: 15% compared to baseline mesh (duato's protocol based adaptive negative-first routing on 2d-mesh)
IEEE DAC 2021[105]	Versa-DNN	Ring	Multi-DNN Workloads, Data-intensive	Separating the request and reply packets to different virtual networks	FPGA prototyping (xilinx VCU118), timeloop	Performance improvement: 2.5x, energy efficiency improvement: 30% compared to baseline
IEEE TCOMPUT AID D 2024[52]	INDM	Hybrid	DNN Accelerators, Compute-intensive	Xy-routing	Propose a latency estimation model for design-time exploration of chiplet-based DNN accelerators	Latency reduction: 26.93%-79.78%, EDP reduction: 26.00%-73.81% compared to baseline noc
IEEE TECS	SIAM	Mesh	Large DNNs, Data-	\	Simulation	Energy-efficiency

2021[43]			intensive		(booksim,neurosim,ramulator,vampire)	improvement: 130x, 72x compared to nvidia V100 and T4 gpus for resnet-50 on imagenet
arXiv 2024[50]	NetSmith	Irregular	General Purpose, hybrid	Table-based routing, and VC allocation	Simulation (parsec, gem5, heterogarnet, dsent), netsmith framework	Latency improvement: 8%-13.5%, throughput improvement: 18%-75% compared to legacy networks

的通信结构及流程的探讨较少，并且对于工作负载分级下中介层可重用性的研究尚不充分。Partha 等人的研究集中于多核系统级芯片（MP-SoC）领域[4]，而 Yin 等人的工作则侧重于 NoC 技术，如模块化路由设计[2]。尽管 Bharadwaj 等人[5][6]的研究提供了 NoI 架构的概述，但缺乏对 NoI 和互连网络架构的系统分析。最近，Liu 等人对基于 Chiplet 的系统架构进行了全面回顾，强调了系统级互连技术对性能的影响及其在 Chiplet 集成中的关键作用。然而，该研究在面对复杂通信拓扑和多样化工作负载时的性能分析方面仍有所不足[7]。Das 等人在文中提到了可重构的 NoI，但并未对其进行深入探讨，留下了进一步探讨及分类的空间[8]。据此，本文总结了 NoI 拓扑结构的最新进展，内容涵盖了从协议与接口层面、微体系结构层面（包括拓扑类型、工作负载、微架构）到 NoI 与 NoC 相结合架构的性能评估，意图呈现一个全面的 NoI 研究的视角。

当前的可重用 Chiplet 架构克服了单一芯片在系统级的局限性，但 NoI 和 Interposer 的设计和制造已成为 Chiplet 定制开发周期和成本的瓶颈。通过重复使用 Interposer，可以进一步降低 NoI 的开发成本，缩短产品周期。本文调查了异构 Chiplet 间通信的最新建进展，并对当前最前沿的 Chiplet 基础 NoI 设计进行系统总结和对比（如表 2 所列）。展示了多样化拓扑结构在提升系统性能和能效方面的显著进展，特别强调了中介层在不同工作负载下的可重用性的重要性。这些分析为设计人员提供了一个有价值的参考框架，有助于优化基于 Chiplet 架构的系统设计，主要贡献如下：

1. 对使用 NoI 的 Chiplet 间通信的基本原理、方法和功能进行了深入、全面的最新进展分析，展现了基于 Chiplet 的系统设计中 NoI 的重要发展趋势。

2. 展示了基于工作负载导向的可重用中介层与拓扑设计的重要性，并对现有可重用方案进行了系统的分析。

3. 从协议和接口层面、微体系结构层面以及整体通信架构层面进行了详细的系统调查，对不同形状的 NoI 拓扑结构进行了比较分析，并通过定量图表进行技术的总结和对比。

本文结构如下。第二部分概述异构 Chiplet 的 NoI 通信架构，讨论通信协议和接口层面，强调设计挑战，并从微体系结构的角对中介层的拓扑形状和可重用方案进行了详细分类和分析。第三部分介绍了评估和建模的主要模拟器与工具集，为 NoI 和 NoC 的性能优化提供了强有力的支持。第四部分则聚焦于整体通信拓扑（NoI+NoC），系统总结并分析了能够启发整体通信结构设计的死锁路由和容错机制，并介绍了新兴技术的最新进展。第五部分讨论了未来研究的重要方向、关键需求与挑战，并提出了可能的解决方案。最后对全文进行了总结。

2 异构 Chiplet 架构中的 NoI 通信设计

异构 Chiplet 架构的发展对互连技术提出了严苛的要求，其中高速且高效的数据交换机制是维持系统性能的关键。如图 2 所示，基于 Chiplet 的架构由并排镶嵌到硅中介层的 Chiplet 互连组成（无源中介层如图 4（a），有源中介层如图 4（b））。在该架构中，微凸块（ μ bumps）以及硅中介层中的导线提供 Chiplet 间连接、机械支持以及封装外信号或通过其中的 Thorough-Silicon-Via（TSV）传输的来自 C4 bumps 的电源、输入输出（I/O）、地线（Ground）等信号。然而， μ bumps 的最小间距限制了每平方毫米 Chiplet 的 μ bumps 数量，Goyal 等人通过将多个 IP 的单项环链接在一起来使多个 Chiplet 之间共享端口以最大程度的减少 μ bumps 的开销[9]。此外，NoI 技术实现了 Chiplet 间的高密度、低延迟通信，这种技术不仅提供了直接的数据通信路径并消除了基板上的连接，还支持不同制程和功能的芯片异构集成，大大增强了系统设计的灵活性和扩展性。与传统

的 NoC 相比, NoI 提供了更大的布局灵活性和更高的带宽, 有效地解决了多芯片模块通信的瓶颈。更重要的是, 随着 Chiplet 数量的增加, NoI 能够保持通信效率, 确保系统性能稳定。

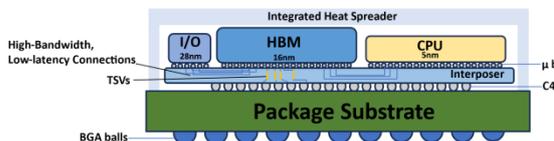


图2 基于 Chiplet 的 NoI 系统架构中的高效互连示例

Fig. 2 Example of Efficient Interconnect in Chiplet-based NoI System Architecture

2.1 协议和接口技术的演进

建立高效的 Chiplet 间通信连接是决定整个芯片系统性能的关键因素。业界已广泛采用可用于 NoI 技术的 D2D 芯片互连设计, 如在硅中介层中高速互连的 OIF-CEI-112G、224G 等 SerDes 系列串行接口 (USR 用于超短距离, 限制了在大规模 Chiplet 芯片上的长距离传输) (如图 3 (a) 所示)。适用于小数据短距离频繁移动的 AIB 和 MDIO 技术 (例如 Intel 的 Embedded Multi-Die Interconnect Bridge (EMIB)、FOVEROS), 台积电的 LIPINCON, AMD 的 Infinity Fabric、3D V-Cache 以及 Marvell 的 Mochi Interconnect 等低

功耗并行高端口数接口 (如图 3 (b) 所示)。然而, 各厂商之间多样的协议阻碍了异构集成向可重用方向的进一步发展。

开放计算项目子组开放域特定架构(ODSA)绘制了一份不同供应商的 Chiplets 也可轻松重用并集成到中介层上的蓝图, 促进了 Chiplet 接口标准化的发展[10]。近年来, 标准化 Die to Die (D2D) 接口协议的发展, 如 Bunch of Wires (BoW) [11] 和 Universal Chiplet Interconnect Express(UCIe)[12] 为 NoI 系统设计提供了重要基础。这些协议详细描述了物理层、协议层以及相关的电气和机械规格, 为 NoI 系统设计提供了标准化、高带宽、低延迟的互连方案, 极大地促进了来自不同厂商的异构计算资源的集成[13][14]。但它们在单个性能指标上并不具有显著优势, 难以对所有类型的工作负载实现最优化处理[15]。最近, Feng 等人首次提出了异构接口 (Hetero-IF) 的概念, 结合并行和串行接口, 根据需求选择合适接口, 提升异构 Chiplet 系统 NoI 的灵活性、可扩展性和性能, 在各种工作负载下提供显著的性能和能耗改进[15]。异构接口允许在更广泛应用中重复使用, 显著降低了非重复性工程 (NRE) 成本, 为高效的异构 Chiplet 系统设计开辟了新的可能性。当然, 异构接口的引入虽然灵活, 但也为只使用单一接口的芯片增加了额外的成本和设计复杂性。

表 3 典型 Chiplet 互连方案之间的比较

Table 3 Comparison between Typical Chiplet Interconnect Solutions (Data Accessed in 2024/8)

Specification	SerDes[16]	AIB[17]	BoW[11][18][19]	UCIe[12]	ACC[20]	LIPINCON[21]
Institution	Rambus/OIF	Intel	ODSA	UcIe Union	CCIA	TSMC
Maximum Data Rate (Gbps/wire)	224	2(Gen1) 6.4 (Gen2)	32	32	128	2.8
Latency (ns)	\	3.56(Gen1)	<2~4 <15(with FEC)	<2	6	\
Power (PJ/bit)	0.8	0.85(Gen1)	<0.25-1.0	0.25 - 1.25	2.5	0.56
Reach (mm)	10(USR) 50 (XSR)	<10 (Gen1)	≤10(basic) ≤50(advanced)	10-25(standard) 2 (advanced)	1~50	≤0.5
Bump Pitch(um)	\	≤55(Gen1)	≤150(basic) ≤55(advanced)	100- 130(standard) 25-55(advanced)	153	\
Transmission Mode	Serial	Parallel	Parallel	Serial	Serial	Parallel
Signal Types	Differential	Differential	Differential	Single-Ended	Differential	Differential

表_3 整理了上面提到的接口间的差异, 揭示了开发通用接口的挑战, 同时展现了当前接口技术在灵活性及兼容性方面的发展空间。为此, 我

们可以从多个角度进行改进: 开发可以在应用前调整串行与并行带宽比例的异构接口以进一步提高灵活性和资源利用率; 优化能效设计以减少功

耗和热量产生，尤其在能量和面积受限的场景中；制定统一的接口协议标准，促进不同厂商芯片的兼容性和互操作性，并将异构类型接口也纳入统一标准。同时，研究方向还包括开发高效路由算法和低延迟通信设计以提高网络性能和实时应用表现；提升接口的高带宽传输能力以满足数据密集型应用需求；设计高度可扩展的接口以支持更大规模的芯片系统和通信节点；集成安全功能以确保数据传输的安全性和完整性。此外，在高互连带宽的多芯片系统中，D2D 接口占据了大量面积，因此进一步优化接口面积成为亟待解决的问题。Iff 等人在名为 HexaMesh 的文中，提出了一种 Chiplet μ bumps 的分配方法[22]，通过统一 μ bumps 的设计尺寸和分区方式，确保了 D2D 连接与中介层的匹配。这种方法通过生产一些专门用于边界的 Chiplet，可以有效减少 D2D 接口的设计面积，无需四周都设计为 D2D 接口。更进一步的，通过模块化图块分配资源，将 Chiplets 和中介层分为大小相同的块，每个块包含一组通过 μ bumps 实现的预定义连接，这些连接负责电源、时钟传输、外部 IO 连接和 Chiplet 间通信，提高了中介层的可重用性[23][24]。

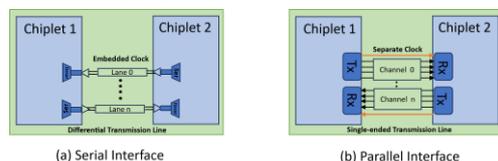


图 3 芯粒间通信的串行 SerDes 系列接口 (a) 与并行接口 (b)

Fig.3 Serial SerDes (a) and Parallel(b) Interfaces in Chiplet-to-Chiplet Communication

2.2 NoI 设计中的架构选择与优化

2.2.1 硅中介层的设计选择与应用场景

NoI 技术通过定制网络有效连接各 Chiplet，极大推进了芯片单元的模块化系统集成，提升系统性能并降低通信延迟。虽然全连接拓扑理论上提供了最高的带宽和最低的延迟，但考虑到全连接拓扑的成本（大量的预配置冗余电路）和实现难度，在实际应用中通常不会作为大规模 NoI 的首选。特别是人工智能快速发展后，基于 Chiplet 架构的芯片迭代速度加快，工作场景变得更加复杂，对模块化和定制化设计需求也相应增加。因此，常

采用多样的通信拓扑，如规则的环形、网状、环面、蝶形以及不规则拓扑、混合拓扑等，以适应不同的应用需求，优化数据处理效率。这些拓扑结构与精细的路由算法相结合，优化了 Chiplet 间的数据传输路径。尽管上述中介层拓扑设计提高了系统的可扩展性，但也导致了成本的增加。此外，平分交叉链路容易成为 Chiplet 硅中介层中的瓶颈[25]，而在不同负载下，其性能、可扩展性、可重用性以及能效方面仍略有不足。因此，为了避免中介层设计费用吞噬掉 Chiplet 重用带来的好处（如节省设计成本和迭代时间），学术界和工业界已开始深入探索不同工作负载及形状的 NoI 微体系结构，推动高性能、低成本且可重用 NoI 架构的发展。这种对芯片通信拓扑结构的持续优化和研究，不仅旨在克服传统 SoC 和基于总线架构的限制，更是推动通信策略创新，以适应现代计算基础设施的不断演进和发展需求。

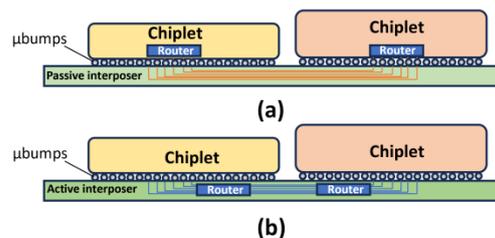


图 4 用于 Chiplet 间通信的无源 (a) 和有源中介层 (b)
Fig. 4 Passive(a) and Active(b) Interposer for Inter-Chiplet Communication

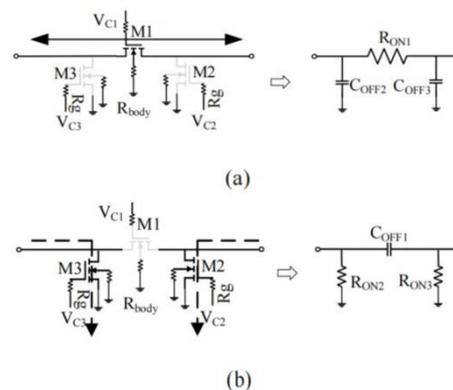


图 5 一种可重用中介层中“路由器”开关单元示意图(a)导通状态(b)关断状态的简化小信号等效电路[26]

Fig. 5 Simplified small-signal equivalent circuit of the switch unit in a reusable interposer: (a) on state (b) off state

硅中介层实际上就是一个可以集成其他较小 Chiplet 的大芯片。在现代集成电路技术中，用于 NoI 的硅中介层分为无源和有源两种类型[27]，分别应对不同的设计需求和成本效益。无源中介层

仅通过金属导线（如图 4（a）所示）提供连接，适用于成本敏感但面积和复杂度较高的应用；而有源中介层则集成了 CMOS 晶体管等有源元件（如图 4（b）所示，部分可重用中介层采用如图 5 所示的结构进行调整），不仅能实现更高效的数据路由和电源管理，还能够优化性能、简化 Chiplet 的复杂度，并显著提升数据传输速度。有源中介层尤其适用于需要高数据带宽和低延迟的大型系统级封装，虽然制造成本高，但可以提供更加灵活和可扩展的解决方案。如果硅中介层的设计可重用于不同规格的同种负载的 Chiplet 架构，将有助于降低整体成本，并缩短开发周期，使其在高性能应用中的应用前景更加广阔[28]。

研究人员已经探索了众多通用和特定领域应用的可重用硅中介层架构设计。早期阶段，例如 FENG 等人[29]提出的 FSMC 复用方案在中介层上构建标准化插座并将 Chiplet 插入其中。此方法严格限于拥有相同占用空间的小芯片，并且仅允许固定放置。Stow 等人[30]提出的包含可用于连接 Chiplet 的 Silicon Interposer Service Layer(SISL)，为 Chiplet 提供了灵活且重用的互连架构，但其采用的固定拓扑策略会在处理各种工作负载时遇到性能问题，并造成面积资源浪费。而 Nexsus 是一种新颖的利用中介层特性，支持模块化“即插即用”的 Chiplet 集成架构，旨在降低系统级封装（SiP）的制造成本，通过利用迷你链 IP 连接拓扑上的直接通信来解决 SiP 封装技术的局限性，但 Nexsus 主要专注无源中介层[31]。

早期阶段所提出的可重用中介层架构主要采用固定网络拓扑，通常不可重构并且芯粒位置固定，在处理多样化工作负载时可能导致性能问题和资源浪费。此后，可作为中介层可重用发展过渡的 SiPterposer[23]是另一种完全无源结构，可以在 Chiplet 组装时配置为任意自定义互连拓扑，具有明显的经济优势，但这种设计也面临未充分利

用的电线资源导致的性能问题。近两年，基于图块、可配置的可重用中介层成为新的发展趋势，像是 Jiao 等人[32]提出的基于网状结构且采用现场可编程的硅中介层（FPIA）的多芯片集成方案。该设计通过可编程互连结构和自动化物理集成工具提升系统的灵活性和硬件效率，采用基于瓦片的互连布局，包括交汇点和平行轨道，以实现微凸点连接的灵活性。将在下文提到的通用中介层架构（GIA）也属于此类。当前的无源中介层解决方案仍然缺乏灵活高效的远距离通信，不易于集成可扩展性较差的模拟功能，如电源管理和系统输入输出信号（IO），并且 Chiplet 集成在无源中介层上仍然存在一定的局限性。当有源中介层多次重用，可以抵消掉有源中介层成本贵的缺陷，故本文主要关注有源中介层可重用方面的内容。

2.2.2 多样化的 NoI 拓扑结构分析

虽然可重用中介层架构取得了一定的成果，但中介层上 NoI 架构的通用性和性能互为制约。通用拓扑结构设计无法提供最高的通信效率，对于特定工作负载（例如计算密集型、数据密集型、混合型）的中介层可重用性和可重构性仍需深入研究。异构 Chiplet 的 NoI 架构面临独特挑战，多样化工作负载对通信网络提出差异化需求，开发出适用于所有工作负载且不增加成本的可重用中介层困难重重。为了促进可重用性及可重构性的进一步发展，使 Chiplet 芯片集成成本降低并便于小型和利基市场采用，本节主要关注不同拓扑形状及其可应用工作负载的可重用中介层的构建，分析每种架构的使用场景及整体工作负载的性能瓶颈和潜在优化方向（表 4 对本节提到的 NoI 拓扑结构进行了直观的对比）。

表 4 NoI 拓扑结构直观对比

Table 4: Intuitive Comparison of NoI Topologies

Topology	Characteristics	Advantages	Disadvantages	Cost&Complexity
Ring	Closed loop, each node connects to two adjacent nodes.	Simple, easy to implement, reduces coherence complexity.	Single point of failure, scalability issues with latency and bandwidth as node count increases.	Low complexity, low cost.
Mesh	Nodes connect through a grid pattern, suitable for medium-scale systems.	Simple, robust fault tolerance, flexible routing.	Increased latency and energy consumption with network scale.	Medium complexity, moderate cost.
Torus	Combines features of ring and mesh, with additional connections for higher efficiency.	Higher communication efficiency and bandwidth on larger scales.	Complexity in connections increases implementation difficulty and cost.	High complexity, higher cost.

Butterfly	Multi-level switching nodes forming a structure similar to a butterfly.	High bandwidth and low latency, smaller network diameter.	High implementation complexity, may require sophisticated routing and management.	Very high complexity, very high cost.
Irregular	Flexible node connections tailored to specific needs, optimizing performance.	Customizable for specific high-performance tasks, adaptable.	High design and debugging costs, may lead to increased development effort.	High complexity, variable cost based on design specifics.
Hybrid	Combines multiple topological advantages, optimizing data paths and bandwidth.	Optimizes latency and bandwidth, flexible and scalable.	High design complexity, often comes with higher costs and power consumption.	High complexity, typically higher cost.

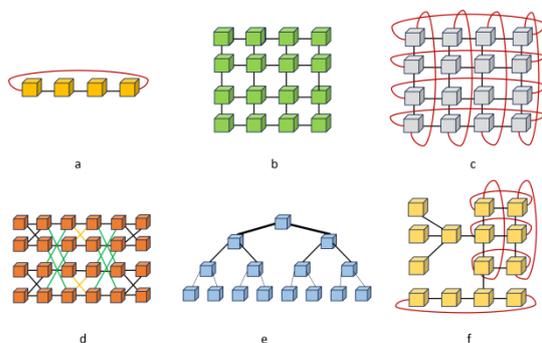


图 6 典型 NoI 网络拓扑结构

Fig. 6 Typical NoI network topology

A) 环形 (Ring) 拓扑

环形拓扑是一种闭环结构，其每个节点都连接到两个相邻节点，形成一个环，如图 6 (a) 所示。该拓扑结构简单且易于实现，简化了缓存一致性的实现问题[33]，适用于小规模芯片系统。缺点是单点故障会应影响整个网络，随着节点数量的增加，延迟和带宽会受到限制。应用于神经网络中的 NN-Baton 架构采用了环状拓扑，考虑了多种应用于 Deep Neural Network (DNN) 的 NoI 探索，并弥补了 Simba 架构没有为工作负载映射提供完整的分析框架以及 Chiplet 粒度的指导的不足[34]。在无源中介层上的 Monad 架构虽使用的是环形拓扑，但其 EDP 优于 Simba 以及 NN-Baton，依旧为其应用在计算密集型负载上奠定了基础[35]。此外，POPSTAR[36][37]基于光电连接的 Chiplet 间环形结构，以及无缓冲多环 (Multi-Ring) 结构[38]都属于通用拓扑。

B) 网状 (Mesh) 拓扑

随着互连 Chiplet 数量的增加，更高基数的可扩展网状拓扑通常成为中等规模芯片集成系统的首选方案。在网状拓扑中，每个节点通过网格状连接与相邻的节点通信，如图 6 (b) 所示。这种结构的优点是简单且容易实现，并且能够提供较高的容错性和灵活的路由选择。适用于需要均衡通信负载的应用，如一般的计算和数据传输任务。缺点是随着网络规模的增大，通信的延迟和能耗会增加。IntAct 是最早在有源中介层上使用网状

NoI 的架构之一，通过在一个有源中介层上集成六个 Chiplet，实现了分布式互连和集成电源管理，在性能和功耗方面都取得了显著提升，能够集成多达 896 个核心，展现了具有低延迟分布式互连的 Chiplet 系统的可扩展性，便于实现中介层的重用[39]。Simba 架构通过引入切片优化来减少 Chiplet 间的通信开销，在硅中介层上构建了平铺优化的网状 NoI，并对非均匀工作负载进行了深入的研究[40]。Wang 等人结合胖树拓扑，设计了一种灵活的可重新划分的网状 NoI 设计，可以灵活的重新配置为权重或者特征子网，取代了单一的 NoI 网络，其额外增加的开销可以忽略不计，可以在不同 AI 芯片上重复使用[41]。而为了满足大规模 Chiplet 的需求，克服传统的扁平拓扑结构并不灵活的缺点，Feng 等人提出了一种更高效和可扩展的基于 2D 网状 NoC 的灵活建立高基数互连网络的方法[42]。此外，基于网状拓扑的 SIAM 架构致力于针对基于 Chiplet 的 DNN 加速器进行拓扑探索，Energy-Delay-Product (EDP) 逊于 INDM，但其在处理数据密集型负载上表现优异[43]。最近，紧凑封装高扇出的 HexaMesh 架构则通过优化网状拓扑上 Chiplet 的排列顺序扩展到了数百个 Chiplets，摆脱了网格性能有限的缺点，同时保持了距离短和频率高的优点[22]。

C) 环面 (Tours) 拓扑

环面拓扑则是一种结合环形和网状拓扑特性的混合拓扑结构，通过在多个环之间建立额外的连接来提高通信效率。每个节点不仅连接到相邻节点，还连接到远端节点，从而形成多个相互交错的环，如图 6 (c) 所示。这种结构能够在更大规模的系统中提供更高的通信效率和带宽。环面拓扑适用于对高吞吐量和低延迟要求较高的应用，如高性能计算和数据中心网络。环面拓扑额外的连接使得数据在网络中有更多的路径选择，减少了通信瓶颈和延迟。缺点是其复杂的连接结构也增加了实现的难度和成本。DCRA 使用了可软件配置的 2D 环面网络来进行 Chiplets 之间的连接，有效应对了不规则应用[44]。

D) 蝶形 (Butterfly) 拓扑

蝶形拓扑是一种高效的 NoI 结构，适用于需要大规模并行计算的应用中。在蝶形拓扑中，节点通过多个层级的交换节点进行连接，形成类似蝴蝶的结构，如图 6 (d) 所示。该结构提供高带宽和低延迟的数据传输路径，并具有比上述拓扑更小的网络直径，适用于高吞吐量和低延迟的并行计算任务[45]。蝶形拓扑的实现复杂度较高，尤其在大规模系统中，可能需要更复杂的路由算法和管理机制。一种名为“双蝴蝶”的拓扑被提出，可用于 64 核芯片和四个周围 DRAM 堆栈的基准系统[45]。该拓扑略增加了总链路和长链路的使用，充分利用了硅中介层的额外路由资源，减少了核心到内存的路由跳数，从而提升了性能。为应对碎片化 NoC 的挑战，Kannan 等人提出了 ButterDonut “错位”拓扑，将路由器放置在相邻 Chiplet 之间共享，避免了中介层流量对共享链路的竞争问题，从而减少了延迟[25]。在此基础上，研究人员还开发了 Kite 系列拓扑。Kite Medium 和 Kite Large 蝶形拓扑架构通过高效利用长链路，在处理 HPC 负载时减少了平均跳数，实现了更低的延迟和更高的吞吐量[5]。此外，Kite Small 和 Kite Medium 拓扑在数据中心和云计算环境中，通过自适应调整拓扑结构，能够支持多种工作负载，同时提供更高的吞吐量和更低的延迟[5]。

E) 不规则 (Irregularly) 拓扑

在网状或环形等常规拓扑可能不适用的情况下，不规则拓扑凭借其灵活性被证明能够提供更低功耗和更高性能。不规则拓扑节点间的连接根据具体需求定制，能够根据应用场景灵活调整，以优化性能和资源利用率，如图 6 (e) 所示的胖树拓扑。不规则拓扑适用于异构计算环境和特殊应用场景，特别是需要高度定制化和优化的高性能计算任务。然而，不规则拓扑的设计和实现复杂度较高，可能导致开发和调试成本上升。例如，SWAP 通过高效的多目标优化 (MOO) 机制生成单跳的 NoI 架构，其不规则拓扑特性使其能够并行运行多个深度学习 (DL) 工作负载，提高链路利用率[46]。Floret 基于空间填充曲线，提出了一种新型不规则网状 NoI 架构，克服了传统 NoI 架构的多跳拓扑限制，能够将相邻的神经网络层分配到相邻的 Chiplet 中，提升卷积神经网络推理任务的性能和成本效益[6]。此外，GIA 支持不同尺寸 Chiplet 的灵活放置，并提供可配置的高性能互连

结构[47]。GIA 采用模块化的基于瓦片的中介层结构，允许 Chiplet 放置在中介层的任意部分，支持多种网络拓扑和路由策略，以增强中介层的可重用性和适应性。实验表明，GIA 相比其他架构显著提升了性能并节省了功耗。此外，支持 Chiplet 间多方式互连的无线接口技术也能够实现多种不规则网络拓扑[48][49]。拓扑生成框架 NetSmith 则通过 MILP 公式生成相对最优的拓扑，为可重用中介层提供了新的思路[50]。

F) 混合 (Hybrid) 拓扑

混合拓扑通过结合多种拓扑结构的优点，优化数据传输路径，减少延迟，提高通信带宽，从而显著提升性能，并提供灵活、可扩展的设计方案，优化资源利用，如图 6 (f) 所示。虽然混合拓扑因其灵活性和高性能受到广泛关注，但其设计和实现复杂度较高，可能需要更复杂的路由算法和管理机制。此外，这种灵活性通常伴随着更高的成本和功耗。例如，Adapt-NoC 通过自适应链路和路由器架构，改变不同子网的拓扑以适应不同工作负载，从而提高系统性能和资源利用率[51]。INDM 通过优化跨芯片互连网络和数据流映射，显著减少能耗和延迟，提高神经网络推理效率。其架构采用层次化互连网络，包括多环片上网络和基于集群的片间网络。多环片上网络利用环形拓扑，实现高效数据传输和低延迟；基于集群的片间网络则采用不规则拓扑，通过灵活的 IO die 连接集群内的计算芯片和 DRAM，相较于其他架构（如 SIMBA 和 NN-Baton），显著降低能耗和延迟[52]。此外，Feng 等人提出了基于新的软件定义接口分组方法，将基于 2D 网状 NoC 的 Chiplet 连接到高基数网络的规范[42]。

2.2.3 不同应用场景中的 NoI 拓扑选型

在不同的应用场景中，选择 NoI 拓扑结构时需要综合考虑系统规模、性能需求、成本以及实现的复杂性等多方面因素。基于对多样化 NoI 拓扑结构特性的分析，以下总结分析了一些典型应用场景及其对应的拓扑选择标准。

计算密集型应用

计算密集型芯片通常涉及大量的计算任务，需要高带宽和低延迟的 NoI 架构以支持频繁的数据传输和快速的 Chiplet 间通信。这类工作负载主要涉及科学计算、并行处理、模拟仿真以及神经网络

络的推理等，强调低功耗和高效率。树形或蝶形拓扑结构较为适合。这些拓扑能够提供高带宽和低延迟的通信路径，满足频繁的数据传输需求。

数据密集型应用

数据密集型芯片通常涉及大量的数据处理和内存访问，特别是在训练深度神经网络、数据分析和视频流处理时，需要频繁访问内存和存储设备。这需要高带宽的内存访问和高效的数据传输，以支持大规模数据集的处理和传输。网状或环形拓扑能够提供多路径通信和高冗余性，适应大规模数据集的处理。

混合型应用

混合型负载芯片需要处理多种类型的工作负载，包括计算密集型和数据密集型任务，以及不同应用之间的通信需求。设计时需要确保集成芯片能够提供足够的灵活性和性能，以满足广泛应用场景的需求。环面、不规则或混合拓扑结构能够提供高灵活性和可定制性，通过动态重构优化资源利用，提供更高的吞吐量和更低的延迟。

在设计异构 Chiplet 的 NoI 通信拓扑时，应根据具体应用场景选择最合适的中介层拓扑，综合各结构的优缺点，以实现最佳的性能和成本效益。通过深入理解 NoI 拓扑的分类以及应用，研究人员可以为未来芯片间通信技术和可重用中介层的发展提供支持，推动下一代半导体设备的进步。表 2 总结了 NoI 通信架构的相关方法，并对表中提到的方法进行了比较。

总结分析表明，整体工作负载的性能瓶颈主要集中在通信延迟、带宽限制、高功耗和低能效，以及固定拓扑结构的可扩展性差。此外，适应不同负载的可重用性和生成合适的最小 Chiplet 与中介层的集合仍面临重要挑战。为应对这些瓶颈，

研究人员可以采用动态自适应拓扑结构和智能路由算法，以优化数据传输路径并减少延迟。同时，应用数据压缩技术和能效优化策略，有助于降低带宽需求和功耗。模块化设计和可重构架构的应用，不仅提高了系统的可扩展性和灵活性，还增强了其适应性和可重用性。通过结合多种拓扑结构的优点，优化数据传输路径并提升通信带宽，结合异构计算集成和系统级优化，可以实现最佳的性能和成本效益，最终推动下一代半导体设备的发展。

3 Chiplet 系统与 NoI 架构的性能评估与优化

在上一节对可重用中介层的发展、应用需求、多样化拓扑，以及不同工作负载对 NoI 需求的分析基础上，本节将进一步讨论如何配置并仿真整个 Chiplet 系统架构，以评估其在实际工作负载下的性能表现。通过这些仿真和优化，能够更深入理解 Chiplet 间通信和 NoI 架构的设计空间，并探索各种配置和组合策略对系统性能的影响，从而为特定应用找到最优解。

3.1 Chiplet 通信与 NoI 架构的仿真工具

评估 NoI 性能需要先进的软件模拟器来处理信息并对其各种功能做出评估，以快速探索和优化设计空间，实现特定工作负载下的最佳性能。虽然目前尚无专门用于 NoI 性能评估的工具，但通过修改和扩展现有的 NoC 模拟器，研究人员可以在专用 NoI 评估工具开发完成之前，对 NoI 的性能进行有效测试。

表 5 当前常用 Chiplet 模拟器的比较

Table 4 Comparison of Currently Used Chiplet Simulators

Simulator Name	Supported Architectures (Types)	Chiplet Support	GPU Support	Power Simulation	Detailed Timing Simulation	System-level Simulation	Programming Language	Additional Notes
Gem5[53][54]	ARM, x86, MIPS, etc.	✓ (Via gem5_chiplet extension)	✓	✓	✓	✓	C++, Python	-
Sniper[57]	Primarily x86	✓ (Via chiplet_sniper)	-	-	✓	-	C++	-
GPGPU-Sim (shared memory)[58]	CUDA GPU	✓ (Via shared memory model)	✓	-	✓	-	C++	-
GPGPU-Sim (message passing)[59]	CUDA GPU	✓ (Via message passing model)	✓	-	✓	-	C++	-
SIAM[43]	Deep neural networks (DNNs)	✓ (Chiplet-based)	-	✓	✓	✓	Python, C++	-

		imc architecture)						
HeteroGarnet [56]	Multi-clock domain, frequency-crossing	✓	-	✓	✓	✓	C++, Python	Supports diverse interconnect systems
Gem5-Chiplet extension [55]	ARM, x86, MIPS, etc.	✓	✓	✓	✓	✓	C++, Python	Extension for gem5, enhanced for chiplet simulation
Chiplet-Sim[60]	Systolic array-based accelerators	✓	-	-	✓	-	Python	-
Muchisim[50]	Multi-node, multi-chiplet tile-based	✓	-	✓	✓	✓	C++, Python	-

像 Gem5 这样的 NoC 模拟器可以根据实验要求灵活的对 Chiplet 通信进行模拟, 包括 NoI 和 NoC。考虑到 NoI 网络通常会受到传输延迟、带宽、吞吐量和功耗等多种因素的影响, 因此需要在模拟中综合考虑这些指标。

表 5 中比较了当前广泛使用的 Chiplet 技术仿真工具平台。Gem5[53][54]及精确建模的 Gem5-Chiplet Extension[55]支持 ARM、x86、MIPS 架构, 适用于研究 Chiplet 系统和多种配置。值得注意的是, HeteroGarnet[56]作为 Gem5 的一个重要扩展, 支持多时钟域和频率交叉, 能够模拟多样化的互连系统, 这使得它在异构 Chiplet 设计中尤为有用。Sniper 通过 Chiplet_Sniper 集成, 支持 x86 架构[57]。GPGPU-Sim 通过共享内存[58]和消息传递[59]模型支持 CUDA GPU。Muchisim 专注于多 Chiplet 设计的性能和能耗分析[50]。SIAM[43]和基于 SCALE-Sim 的 Chiplet-Sim[60], 专注于阵列加速器。Muchisim 模拟多节点多 Chiplet 设计, 提供性能和成本分析。这些仿真器利用 C++、Python 或两者结合, 进行详细的系统级仿真和性能评估。此外, 对 NoI 网络延迟的模拟常用修改后的 BookSim2.0[61]进行, 并使用 DSENT[62]或者 Orion3.0[63]来获得 NoI 的面积开销, 其中 DSENT 还可以获得相关的功率开销。这些工具能够全面评估 NoI 和 NoC 系统在不同配置和工作负载下的性能和能效, 为优化设计提供有力支持。

3.2 Chiplet 组合选择与 NoI 架构的优化策略与评估

Chiplet 技术的成功在很大程度上依赖于确定最优 Chiplet 组合, 并优化计算和处理功能的映射以及物理布局, 以创建高度可定制的集成芯片, 这需要在系统延迟、功耗、性能和成本等多方面目标间寻找平衡。通过从可重用的 Chiplet 集合中选择最优组合, 可以满足特定的设计目标和约束, 并针对不同工作负载, 使用最小的 Chiplet 集为应

用程序提供近乎定制的系统性能[64]。多种优化方法已被提出, 如基于贝叶斯优化的 HyperMapper 框架, 通过探索大型设计空间来优化延迟、功耗、面积和成本等多种目标[65], 其可以快速迭代多个 Chiplet 组合, 评估每个组合的潜在性能, 并帮助设计人员找到最优解。为了进一步提升设计的多目标优化能力, Chopin 提出了一种使用算法将 Chiplet 组合成具有成本效益的定制芯片的方法, 实现了空间探索和性能提升的功能[66]。同时, Feng 等人提出的模型将 NRE 成本表示为系统总体、各个 Chiplet 及架构中包含模块的 NRE 成本之和, 展示了多芯片架构如何受益于 Chiplet 产量提高、重用以及异构性[67]。Pal 等人开发了基于整数线性规划的 Chiplet 选择框架和成本模型, 以优化设计和制造成本。这些方法都致力于在复杂的设计空间中寻找最优解。为了在重用中介层的过程中同样有效地针对各种工作负载, 当前亟需将可重用中介层集合及选择标准添加到算法中, 以便能够得到更加全面的最优组合, 进一步降低成本。Graening 等人对影响 Chiplet 架构成本的关键因素进行了量化和建模[68], 为进一步优化提供了基础。前面得到的最优解集中选择最符合设计需求的方案, 进行细化优化。

在 NoI 设计中, 总线宽度、路由器端口配置、缓冲区大小和链路长度等属性直接影响系统的延迟、功率、能耗和总面积。不同 NoI 架构主要在路由器端口数量和链路长度上有所区别, 这影响了架构的灵活性和能效。通过改进仿真平台, 我们可以更有效地验证可重用中介层的性能, 从而避免厂商自行设计中中介层。最终, NoI 架构设计需要在性能、能耗、面积和制造成本之间进行权衡, 以适应不同工作负载应用的需求。这种多目标优化方法对于推动 Chiplet 技术的进一步发展至关重要。

4 NoC 与 NoI 结合架构中的关键技术分析及应用

在上一节中, 本文介绍了对基于 Chiplet 的 NoI 互连网络架构的性能评估。在基于有源中介层的系统中, Chiplets 和中介层可以各自拥有独立的互连网络。NoI 和 NoC 分别解决了不同层面的通信挑战。NoI 提供高密度的 Chiplet 间连接, 通过中介层布线实现不同芯片单元间的高效数据交换, 将当前的 NoC 设计扩展到双层架构[69]。而 NoC 则专注于单个芯片内部的高效通信, 优化功能模块间的数据流。这两种技术的结合构成了一种尖端架构(如图 7 所示), 不仅实现芯片内部高效通信, 还能在 Chiplets 间进行高速、低延迟的数据传输。此 NoC+NoI 架构广泛应用于高性能计算系统, 尤其在先进微处理器和多芯片模块设计中发挥关键作用, 其核心包含互连拓扑、死锁解决和容错机制三个关键技术。本节将深入探讨 NoC 架构如何解决上述核心问题, 从中获取 NoC+NoI 关键技术的启示, 为研究者在设计可重用的中介层时提供参考。

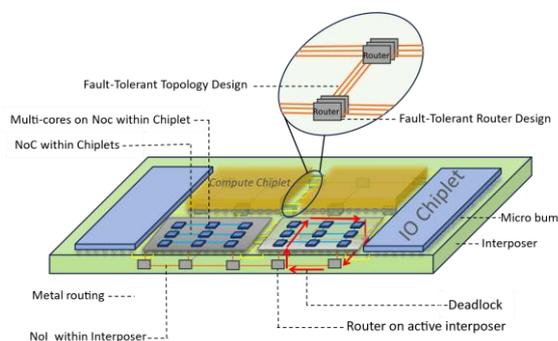


图 7 NoI+NoC 结构示意

Fig. 7 Schematic diagram of NoI+NoC topology

A) 死锁解决与路由优化

在构建可重用中介层的 NoI+NoC 结构中, 多个 Chiplets 的互连网络并保持高性能无死锁路由是一个复杂的挑战。传统的布线和基于全局视图的死锁自由方法难以解决由于集成引起的死锁问题[70][71][72], 因为即使单个 Chiplet 和中介层没有死锁, 但集成到同一封装中可能导致最终死锁[73](如图 7 红色箭头所示)。为此, 需要新的方法来解决这一挑战[2][74]。常见的 NoC 解决方案可

以分为避免、检测与恢复以及周期性恢复方法, 这些方法可为 NoI+NoC 结构的设计提供借鉴。

死锁避免策略可以分为两个不同的分支: 一是基于虚拟通道(VC)和转向模型(turn model)的方法, 这些方法通过施加限制适用于处理少量请求, XY 和奇偶路由算法则是转向模型的代表[70][71][75]; 二是基于流量控制的技术[42][76][77][78], 包括基于消息、包、流量控制单元(Flits)和物理传输单元(Phits)的流量控制。近年来涌现了多种创新方案, 例如, Wang 等提出的无缓冲多环 NoC 高带宽设计, 适用于服务器 CPU 和 AI 处理器, 提供了良好的扩展性并实现了死锁避免[79]。Xiang 等采用带偏转路由的无缓冲区方法减少了硬件开销, 降低了功耗, 并通过分层环扩展了无缓冲区 NoC[80]。Farrokhbakht 等的 Pitstop 方法, 通过解决协议和网络级死锁而无需使用虚拟网络、复杂硬件或错误路由, 以低成本方式避免/解决死锁[81]。Ejaz 等提出的 HighwayNoC 设计, 包含一个 2D 网状网络, 综合了虚拟通道和基于信用的流量控制, 具备广泛适应性[82]。此后, Srivastava 等提出的基于强化学习的 Q-routing 算法, 使数据包选择不太拥挤的路径, 缓解网络拥塞[83]。Orenes-Vera 等提出的 DCRA 架构在运行时可重配置 2D 环面 NoC, 进一步提升了资源利用率[44]。

死锁检测和恢复方法通过监控网络并及时修正来处理死锁。例如, SPIN 通过探测网络来检测可能的死锁, 并同步死锁链内的路由器释放冻结的数据包[71]。静态气泡利用气泡流控制思想, 在任何环形/圆环拓扑中保留一个空闲缓冲区以避免死锁[84][85]。恢复机制使用额外的缓冲区或同步路由器以取得前进。Wu 等人提出了一种通过向上数据包弹出的死锁恢复框架 UPP, 以克服 Chiplet 设计模块化、性能和灵活性的挑战, 进一步提升了性能[74]。

周期性恢复策略如 SWAP[86]和 DRAIN[87]则不需要显示的死锁检测, 适用于一些特定的场景。通过定期处理(交换或排空)潜在的死锁来达成目的。

B) 容错机制与可靠性增强

墨菲芯片良率模型揭示了基于封装的 Chiplet 集成技术的优势[88]。通过使用已知良好芯片

(KDG) 及已知良好中介层进行集成, 可以显著缓解芯片的良率问题[89]。在芯片的总面积相同的情况下, Chiplet 分解的越细, 良率就越高。然而, 这种集成方法也面临挑战: 可重用 Chiplet 及中介层在互连过程中可能会出现物理故障, 从而影响系统的整体性能。为应对这一挑战, Chiplet 及中介层的接口和协议需要专门设计以支持容错功能。

容错措施主要可以通过类似 NoC 的容错拓扑设计与容错路由两方面的优化来实现, 以应对永久性故障带来的系统性能损失。(如图 7 放大处所示)。容错拓扑设计可通过冗余网络和高连接性网络来提升系统容错性能, 前者通过多路径通信确保可靠性, 例如 [90][91][92][93]文中所提到备份路径。后者通过增加节点基数提供路径多样性和减少次要节点的硬件开销, 例如 Chang 等[90]以及 Ren 等[94]提出的路由器冗余法。而容错路由通过动态路径选择和基于负载均衡的路由算法提升系统容错性能, 前者根据实时网络状态选择最佳路径绕过故障, 如 DeFT 以及利用基于虚拟网络的方法来保证无死锁, 同时增强垂直链接 (VL) 选择以容忍 VL 和水平链接 (HL) 故障的 ReD[95]。后者在故障时选择较空闲路径避免过载。

在此基础上, DFT (Design for Test) 技术在 NoI+NoC 架构中也起到了重要作用, 满足了传统的测试方法已难以满足的高效、全面的测试需求。DFT 方法包括 BIST (内建自测试)、JTAG (联合测试行动组) 接口和扫描链设计, 尤其适合于复杂的 NoI+NoC 结构。通过在 Chiplet 和中介层中集成 DFT 结构, 系统能够对关键模块进行深度在线测试和监控, 从而实现故障的有效检测与隔离, 进一步确保系统的整体可靠性。

容错路由还可以视为解决 Chiplet 间死锁的延伸问题, 例如, 通过自适应路由和增加垂直链路的选择冗余度, 缓解网络拥塞和故障问题, 如前文提到的 Q-Routing 算法。对于无线片上网络 (WiNoC), 低延迟区域故障感知 (RFA) 算法通过结合故障和拥塞信息, 选择最优数据包路由路径, 降低故障链路遇到的概率, 并通过拥塞感知机制平衡网络负载[96]。网状和环面网络架构的普遍应用为容错设计 (如 X-Y/Y-X 尺寸有序布线) 提供了良好支持, 且实现难度较低。此外, Stow 等人还展示了如何利用中介层的额外“预付费”硅面积来实现容错, 同时提高产量和成本效益[28]。对于学术界和工业界来说, NoI+NoC 架构的容错

仍然是一个开放的领域, 需要更多研究人员的投入。

C) 实际应用与性能优化

除了在第二节中介绍不同工作负载 NoI 时提及的应用场景, Chiplet 技术已在多个商业架构中得到广泛应用。英特尔的 EMIB 技术用于单个封装内不同芯片的互连, 与 NoC 架构结合处理芯片内通信。赛灵思的 Versal ACAP 通过 NoI 架构连接不同计算内核和存储器, 并集成了 NoC 来高效处理数据, 促进了芯片组之间的低延迟、高带宽通信。AMD 的 Ryzen 处理器利用 Infinity Fabric 高速互连技术连接多个 Chiplets, 其中 NoC 管理本地流量, NoI 处理芯片组间通信[97]。不仅如此, 像是在汽车电子和自动驾驶中, NoI 提供不同传感器、处理器和控制单元之间的高效通信, 而 NoC 优化每个模块内部的数据流动, 确保系统的整体性能和稳定性。这些实例展示了 NoI 及 NoC 架构在现代计算系统中管理复杂通信的重要性, 尤其在 AI (GPT-4、Google 的 PALM)、数据中心和服务器中提升数据吞吐量和降低延迟, 这对处理海量数据集和要求快速处理能力的系统性能至关重要。NoI+NoC 架构通过优化数据传输路径和提升带宽利用率, 在满足多种复杂应用需求的同时, 实现了系统架构的可重用性。这种协同作用使设计者能够灵活应对不断变化的技术挑战 and 市场需求, 优化整个系统的设计和性能, 成为现代半导体设计向更高集成度和更大功能性迈进的关键。

随着网络拓扑结构、路由算法、路由器架构及容错方法的不断改进, 数据包延迟得以显著降低。通过优化拓扑结构、使用多个子网络以及改进资源分配, 网络的吞吐量得到了进一步提升。这些改进显著增强了可重用中介层的 NoI 设计以及 NoI+NoC 架构通信设计的性能。

不仅在可重用性方面, Chiplet 架构的可重构性也引起了研究人员的广泛关注。Chiplet 架构的可重构性源于其模块化设计和灵活的互连机制, 使芯片能够根据实际需求动态激活或关闭特定的 Chiplet 模块和数据路径, 从而优化性能、功耗和热管理。这种灵活性使得 Chiplet 架构在高性能计算、数据中心和人工智能等领域具有广阔的应用前景。随着技术的进步, Chiplet 的可重构性将进一步增强, 为更复杂和多样化的应用提供支持。

5 未来研究方向

尽管技术不断进步，目前的可重用 Chiplet 架构和中介层设计依然面临多种挑战，未来研究方向也十分多样。

中介层灵活性

研究方向涉及中介层在不同尺寸和数量 Chiplet 情况下的灵活放置能力，同时需要在通用性和优化性之间取得有效平衡，因为每个 Chiplet 的需求各不相同，包括带宽、延迟和面积，这不仅涉及功能接口的兼容性，还包括供电和热管理。其中，供电必须可靠，不受芯片布局影响，温度管理需要适应不同位置的芯片，根据实时温度变化调整芯片运行状态。未来趋势应该是向实现高布线利用率的灵活细粒度互连、可扩展、支持远距离传输的方向发展。

通信模式和网络拓扑

各种特定应用的通信模式需要灵活配置 Chiplet 间的互连拓扑和路由。这涉及到网络架构的动态调整，以满足不同应用的需求，确保高效的数据传输和处理能力。考虑到未来 AI 和数据中心的高并发需求，如何确保这种灵活配置不会引发瓶颈也是亟须解决的问题。

容错机制

建立高效的容错机制，确保系统在发生故障时仍能正常运行。需要对制造成本与性能损失之间进行平衡，以实现更具可持续性和成本效益的设计。例如，基于 A*算法的动态路由选择算法，可以动态调整最优路径，最小化数据包传输延迟，适合实时系统，也可以用于故障规避。尽管如此，建立真正高效的容错机制仍然是一个复杂的任务，需要持续的研究和创新。

安全性和验证

随着系统规模的增加，需要在保持高能效、实现网络可扩展性和并行化的同时，解决平面拓扑不能充分利用互连资源的问题，并确保在应用增多时数据传输的安全性，同时考虑到 Chiplet 系统的安全性带来的重大挑战，尤其是在结合不同供应商的芯片时，系统更容易受到硬件安全威胁，这要求开发高效的安全验证工具，以抵御日益复杂的安全威胁，如 LI 等人提出了一种准确检测故障注入攻击引起的软错误位置的方法[98]以及一种识别设计时的漏洞并增强单粒子翻转可靠性的方

法[99]等。

标准化测试

市场需要标准来描述可重用中介层的测试、性能和延迟等内容。由于不能在不损坏 Chiplet 的情况下将其从中介层中剥离，因此不仅需要在组装前对 Chiplet 进行测试，还需要对可重用中介层进行测试，以识别出错误的 Chiplet 和中介层。这包括制定统一的性能和可靠性评估标准，开发 Chiplet 和中介层的联合测试策略，优化现有技术（如 TSV 检测[100][101]）以适应中介层结构，并探索非侵入式测试方法。

硅光子中介层

硅光子中介层作为未来的重要方向，通过光子技术实现超高速数据传输，降低延迟和能耗，提高系统性能和带宽。这一技术有望在高性能计算和数据中心应用中发挥重要作用[102][103]。

基于 Chiplet 设计的仿真和 EDA 工具优化

随着基于 Chiplet 的系统规模不断扩大，目前的开源仿真器和模拟器因缺少精确的 Chiplet 间互连模型和大规模并行仿真能力，无法精确模拟 Chiplet 间的路由层，不足以支持大规模多 Chiplet 系统的仿真需求。这突显了开发更高效、全面的仿真和模拟工具的迫切性，这些工具应能准确地模拟不同 Chiplet 架构在实际应用中的表现，并支持基于 Chiplet 的复杂系统设计。

同时，当前的物理设计和 EDA/CAD 工具亟需优化，以更好地应对复杂的异构集成挑战，提高设计效率，并减少设计与制造之间的差距。进一步发展设计自动化框架将加快设计空间的探索和芯片产品的上市过程，满足快速迭代和高效设计的需求。

系统性能评估架构

基于 Chiplet 的系统需要一个完整的并尽可能通用的系统性能评估架构，以实现快速得出可重用中介层所适用的工作负载。这有助于优化设计，提高系统整体性能。例如，RapidChiplet[104]等预测工具链正在研发和优化中。

通过这些方向的深入研究和开发，不仅能够提升芯片间通信性能和系统集成度，还能推动中介层可重用性的实现，促进异构集成技术的发展，减少 NRE 成本，并加速芯片产品的上市时间。这些努力将为未来计算技术的发展开辟新的道路，并为芯片市场带来显著增长。

6 结论

Chiplet 异构集成技术相较于传统 SoC 设计更具灵活性、可重用性和高效性。作为基于 Chiplet 架构的关键通信基础设施, NoI 对于构建高性能、高能效的系统至关重要。本文深入探讨了基于 Chiplet 架构的通信拓扑结构, 特别是可重用中介层架构, 并提出了一种基于不同工作负载的 NoI 架

构分类方法。文章总结了当前最先进的 NoI 架构, 并详细探讨 NoC 架构对于 NoI+NoC 的网络架构及其关键技术的启发, 明确了未来的关键研究方向。这些研究不仅帮助读者理解这一领域的最新进展和面临的挑战, 还为进一步优化 NoI 与 NoC 结合的架构提供了重要参考。这些优化有望显著提升多个领域的应用性能和效率, 推动半导体设计和集成技术的未来发展。

参考文献

- [1] JANGAM S C, PAL S, BAJWA A, et al. Latency, bandwidth and power benefits of the superchips integration scheme[C]//2017 IEEE 67th Electronic Components and Technology Conference (ECTC). IEEE, 2017: 86-94.
- [2] YIN J, LIN Z, KAYIRAN O, et al. Modular routing design for chiplet-based systems[C]//2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2018: 726-738.
- [3] LOH G H, NAFFZIGER S, LEPAK K. Understanding chiplets today to anticipate future integration opportunities and limits[C]//2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2021: 142-145.
- [4] PANDE P P, GRECU C, JONES M, et al. Performance evaluation and design trade-offs for network-on-chip interconnect architectures[J]. IEEE Transactions on Computers, 2005, 54(8): 1025-1040.
- [5] SHARMA H, PFROMM L, TOPALOGLU R O, et al. Florets for Chiplets: Data Flow-aware High-Performance and Energy-efficient Network-on-Chip Interposer for CNN Inference Tasks[J]. ACM Transactions on Embedded Computing Systems, 2023, 22(5s): 1-21.
- [6] BHARADWAJ S, YIN J, BECKMANN B, et al. Kite: A Family of Heterogeneous Interposer Topologies Enabled via Accurate Interconnect Modeling. [C]//In 2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 1-6, 2020.
- [7] LIU Y, LI X, YIN S. Review of chiplet-based design: system architecture and interconnection[J]. Science China Information Sciences, 2024, 67(10): 200401.
- [8] Das A, Palesi M, Kim J, et al. Chip and Package-Scale Interconnects for General-Purpose, Domain-Specific and Quantum Computing Systems-Overview, Challenges and Opportunities[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2024.
- [9] GOYAL V, WANG X, BERTACCO V, et al. Nexus: An interconnect for heterogeneous system-in-package architectures[C]//2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2020: 12-21.
- [10] OCP.2019.ODSAWiki.<https://www.opencompute.org/wiki/Server/ODSA>
- [11] FARJADRAD R, KUEMERLE M, VINNAKOTA B. A bunch-of-wires (BoW) interface for interchiplet communication[J]. IEEE Micro, 2019, 40(1): 15-24.
- [12] uciexpress.org/files/ugd/0c1418_c5970a68ab214ffc97fab16d11581449.pdf (accessed on 6/12/2024)
- [13] ARDALAN S, FARJADRAD R, KUEMERLE M, et al. An open inter-chiplet communication link: Bunch of wires (BoW)[J]. IEEE Micro, 2020, 41(1): 54-60.
- [14] Debendra Das Sharma, UCIE White Paper, 2022, [online]Available:<https://www.uciexpress.org/general-8>.
- [15] FENG Y, XIANG D, MA K. Heterogeneous Die-to-Die Interfaces: Enabling More Flexible Chiplet Interconnection Systems[C]//Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture.2023:930-943.
- [16] oiforum.com/wp-content/uploads/OIF-FD-CEI-24G-01.0.pdf (accessed on 6/12/2024)
- [17] [AIB-specification/AIB Specification 2.0.pdf at master · chipsalliance/AIB-specification \(github.com\)](https://aib-specification.github.io/AIB-specification/AIB-specification-2.0.pdf) (accessed on 6/12/2024)
- [18] opencompute.org/documents/bow-specification-v2-0d-1-pdf (accessed on 6/12/2024)
- [19] ARDALAN S, CIRIT H, FARJAD R, et al. Bunch of wires: An open die-to-die interface[C]//2020 IEEE Symposium on High-Performance Interconnects (HOTI). IEEE, 2020: 9-16.
- [20] <http://www.iiisct.com/smart/upload/CMS1/202303/ACCI.0.pdf> (accessed on 6/12/2024)

- [21] LIN M S, HUANG T C, TSAI C C, et al. A 7-nm 4-GHz Arm[®] core-based CoWoS[®] 1 chiplet design for high-performance computing[J]. *IEEE Journal of Solid-State Circuits*, 2020, 55(4): 956-966.
- [22] IFF P, BESTA M, CAVALCANTE M, et al. HexaMesh: Scaling to Hundreds of Chiplets with an Optimized Chiplet Arrangement[C]//2023 60th ACM/IEEE Design Automation Conference (DAC). IEEE, 2023: 1-6.
- [23] EHRETT P, AUSTIN T, BERTACCO V. SiP interposer: A fault-tolerant substrate for flexible system-in-package design[C]//2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019: 510-515.
- [24] NASRULLAH J, LUO Z, TAYLOR G. Designing Software Configurable Chips and SIPs using Chiplets and zGlue[C]//International Symposium on Microelectronics. International Microelectronics Assembly and Packaging Society, 2019, 2019(1): 000027-000032.
- [25] KANNAN A, JERGER N E, LOH G H. Enabling interposer-based disintegration of multi-core processors[C]//Proceedings of the 48th international symposium on Microarchitecture. 2015: 546-558.
- [26] ZHENG Y, SHAN G, LIU Z, YANG Z. A reusable interposer for multi-chiplet microsystems reconfiguration[C]//2023 IEEE MTT-S International Wireless Symposium (IWS). Qingdao, China, 2023: 1-3.
- [27] STOW D, AKGUN I, XIE Y. Investigation of cost-optimal network-on-chip for passive and active interposer systems[C]//2019 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP). IEEE, 2019: 1-8.
- [28] STOW D, XIE Y, SIDDIQUA T, et al. Cost-effective design of scalable high-performance systems using active and passive interposers[C]//2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, 2017: 728-735.
- [29] FENG Y, MA K. Chiplet Actuary: A Quantitative Cost Model and Multi-Chiplet Architecture Exploration[C]//Proceedings of the 2022 IEEE/ACM International Conference on Computer-Aided Design. 2022: 1-9.
- [30] STOW D, AKGUN I, BARNES R, et al. Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5 D/3D integration[C]//2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, 2016: 1-6.
- [31] EHRETT P, AUSTIN T, BERTACCO V. SiP interposer: A fault-tolerant substrate for flexible system-in-package design[C]//2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019: 510-515.
- [32] Jiao B, Xu L, Yu X, et al. FPIA: Communication-Aware Multi-Chiplet Integration With Field-Programmable Interconnect Fabric on Reusable Silicon Interposer[J]//*IEEE Transactions on Circuits and Systems I: Regular Papers*. 2024;71(9):4156-4168.
- [33] VIVET P, GUTHMULLER E, THONNART Y, et al. IntAct: A 96-core processor with six chiplets 3D-stacked on an active interposer with distributed interconnects and integrated power management[J]. *IEEE Journal of Solid-State Circuits*, 2020, 56(1): 79-97.
- [34] Z TAN Z, CAI H, DONG R, et al. Nn-baton: Dnn workload orchestration and chiplet granularity exploration for multichip accelerators[C]//2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021: 1013-1026.
- [35] HAO X, DING Z, YIN J, et al. Monad: Towards Cost-Effective Specialization for Chiplet-Based Spatial Accelerators[C]//2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD). IEEE, 2023: 1-9.
- [36] THONNART Y, BERNABÉ S, CHARBONNIER J, et al. POPSTAR: A robust modular optical NoC architecture for chiplet-based 3D integrated systems[C]//2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020: 1456-1461.
- [37] NARAYAN A, THONNART Y, VIVET P, et al. System-level evaluation of chip-scale silicon photonic networks for emerging data-intensive applications[C]//2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020: 1444-1449.
- [38] WANG T, FENG F, XIANG S, et al. Application defined on-chip networks for heterogeneous chiplets: An implementation perspective[C]//2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2022: 1198-1210.
- [39] VIVET P, GUTHMULLER E, THONNART Y, et al. IntAct: A 96-core processor with six chiplets 3D-stacked on an active interposer with distributed interconnects and integrated power management[J]. *IEEE Journal of Solid-State Circuits*, 2020, 56(1): 79-97.
- [40] SHAO Y S, CLEMONS J, VENKATESAN R, et al. Simba: Scaling deep-learning inference with multi-chip-module-based architecture[C]//Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture. 2019: 14-27.
- [41] WANG M, WANG Y, LIU C, et al. Network-on-interposer design for agile neural-network processor chip customization[C]//2021 58th ACM/IEEE Design Automation Conference (DAC). IEEE, 2021: 49-54.

- [42] FENG Y, XIANG D, MA K. A Scalable Methodology for Designing Efficient Interconnection Network of Chiplets[C]//2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2023: 1059-1071.
- [43] KRISHNAN G, MANDAL S K, PANNALA M, et al. SIAM: Chiplet-based scalable in-memory acceleration with mesh for deep neural networks[J]. ACM Transactions on Embedded Computing Systems (TECS), 2021, 20(5s): 1-24.
- [44] ORENES-VERA M, TURECI E, MARTONO SI M, et al. DCRA: A distributed chiplet-based reconfigurable architecture for irregular applications[J]. arXiv preprint arXiv:2311.15443, 2023.
- [45] JERGER N E, KANNAN A, LI Z, et al. NoC architectures for silicon interposer systems: Why pay for more wires when you can get them (from your interposer) for free?[C]//2014 47th Annual IEEE/ACM International Symposium on Microarchitecture. IEEE, 2014: 458-470.
- [46] SHARMA H, MANDAL S K, DOPPA J R, et al. SWAP: A server-scale communication-aware chiplet-based manycore PIM accelerator[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(11): 4145-4156.
- [47] LI F, WANG Y, CHENG Y, et al. Gia: A reusable general interposer architecture for agile chiplet integration[C]//Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design. 2022: 1-9.
- [48] KADOMOTO J, IRIE H, SAKAI S. Design of shape-changeable chiplet-based computers using an inductively coupled wireless bus interface[C]//2020 IEEE 38th International Conference on Computer Design (ICCD). IEEE, 2020: 589-596.
- [49] KADOMOTO J, MITSUNO S, IRIE H, et al. An inductively coupled wireless bus for chiplet-based systems[C]//2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2020: 9-10.
- [50] GREEN C, THOTTETHODI M. NetSmith: A Design Optimization Framework for Machine-Discovered Network Topologies[J]. arXiv preprint arXiv:2404.02357, 2024.
- [51] KIM D H, ATHIKULWONGSE K, HEALY M B, et al. Design and analysis of 3D-MAPS (3D massively parallel processor with stacked memory)[J]. IEEE Transactions on Computers, 2013, 64(1): 112-125.
- [52] ZHANG J, FAN X, YE Y, et al. INDM: Chiplet-Based Interconnect Network and Dataflow Mapping for DNN Accelerators[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023.
- [53] <https://www.gem5.org/documentation/> (accessed on 07/2024)
- [54] BINKERT N, BECKMANN B, BLACK G, et al. The gem5 simulator[J]. ACM SIGARCH computer architecture news, 2011, 39(2): 1-7.
- [55] https://github.com/GT-CHIPS/gem5_chips (accessed on 07/2024)
- [56] https://www.gem5.org/documentation/general_docs/ruby/heterogarnet/ (accessed on 07/2024)
- [57] https://github.com/FCAS-ZJU/Chiplet_sniper (accessed on 07/2024)
- [58] <https://github.com/FCAS-SCUT/Chiplet-Gem5-sharedMemory> (accessed on 07/2024)
- [59] <https://github.com/FCAS-SCUT/Chiplet-GPGPU-Sim-massgepassing> (accessed on 07/2024)
- [60] <https://github.com/l11bbbyyy/Chiplet-sim> (accessed on 07/2024)
- [61] JIANG N, BECKER D U, MICHELOGIANN AKIS G, et al. A detailed and flexible cycle-accurate network-on-chip simulator[C]//2013 IEEE international symposium on performance analysis of systems and software (ISPASS). IEEE, 2013: 86-96.
- [62] SUN C, CHEN C H O, KURIAN G, et al. DSENT-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling[C]//2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip. IEEE, 2012: 201-210.
- [63] KAHNG A B, LIN B, NATH S. ORION3.0: A comprehensive NoC router estimation tool[J]. IEEE Embedded Systems Letters, 2015, 7(2): 41-45.
- [64] PAL S, PETRISKO D, KUMAR R, et al. Design space exploration for chiplet-assembly-based processors[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2020, 28(4): 1062-1073.
- [65] NARDI L, SOUZA A, KOEPLINGER D, et al. Hypermapper: a practical design space exploration framework[C]//2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2019: 425-426.
- [66] EHRETT P, AUSTIN T, BERTACCO V. Chipin: Composing cost-effective custom chips with algorithmic chiplets[C]//2021 IEEE 39th International Conference on Computer Design (ICCD). IEEE, 2021: 395-399.
- [67] FENG Y, MA K. Chiplet actuary: A quantitative cost model and multi-chiplet architecture exploration[C]//Proceedings of the 59th ACM/IEEE Design Automation Conference. 2022: 121-126.

- [68] GRAENING A, PAL S, GUPTA P. Chiplets: How small is too small?[C]//Proceedings of the 60th ACM/IEEE Design Automation Conference. 2023: 1-6.
- [69] ZHENG H, WANG K, LOURI A. A versatile and flexible chiplet-based system design for heterogeneous manycore architectures[C]//2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, 2020: 1-6.
- [70] DALLY, SEITZ. Deadlock-free message routing in multiprocessor interconnection networks[J]. IEEE Transactions on computers, 1987, 100(5): 547-553.
- [71] RAMRAKHYANI A, GRATZ P V, KRISHNA A T. Synchronized progress in interconnection networks (SPIN): A new theory for deadlock freedom[C]//2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2018: 699-711.
- [72] DUATO J. A new theory of deadlock-free adaptive routing in wormhole networks[J]. IEEE transactions on parallel and distributed systems, 1993, 4(12): 1320-1331.
- [73] TAHERI E, PASRICHA S, NIKDAST M. DefT: A deadlock-free and fault-tolerant routing algorithm for 2.5 D chiplet networks[C]//2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2022: 1047-1052.
- [74] WU Y, WANG L, WANG X, et al. Upward packet popup for deadlock freedom in modular chiplet-based systems[C]//2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2022: 986-1000.
- [75] EBRAHIMI M, DANESHTALAB M. EbDa: A new theory on design and verification of deadlock-free interconnection networks[C]//Proceedings of the 44th Annual International Symposium on Computer Architecture. 2017: 703-715.
- [76] AL FARUQUE M A, EBI T, HENKEL J. Configurable links for runtime adaptive on-chip communication[C]//2009 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2009: 256-261.
- [77] MA S, WANG Z, LIU Z, et al. Leaving one slot empty: Flit bubble flow control for torus cache-coherent NoCs[J]. IEEE Transactions on Computers, 2013, 64(3): 763-777.
- [78] MAJUMDER P, KIM S, HUANG J, et al. Remote control: A simple deadlock avoidance scheme for modular systems-on-chip[J]. IEEE Transactions on Computers, 2020, 70(11): 1928-1941.
- [79] WANG T, FENG F, XIANG S, et al. Application defined on-chip networks for heterogeneous chiplets: An implementation perspective[C]//2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2022: 1198-1210.
- [80] XIANG X, SIGDEL P, TZENG N F. Bufferless network-on-chips with bridged multiple subnetworks for deflection reduction and energy savings[J]. IEEE Transactions on Computers, 2019, 69(4): 577-590.
- [81] FARROKHBAKHT H, KAO H, HASAN K, et al. Pitstop: Enabling a virtual network free network-on-chip[C]//2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2021: 682-695.
- [82] EJAZ A, PAPAEFSTATHIOU V, SOURDIS I. HighwayNoC: Approaching ideal NoC performance with dual data rate routers[J]. IEEE Transactions on Networking, 2020, 29(1): 318-331.
- [83] SRIVASTAVA S, SHAIKH M A, SHIVANEETHA G, et al. Intelligent congestion control for NoC architecture in Gem5 simulator[C]//2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc). IEEE, 2022: 353-360.
- [84] CHEN L, PINKSTON T M. Worm-bubble flow control[C]//2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2013: 366-377.
- [85] RAMRAKHYANI A, KRISHNA T. Static bubble: A framework for deadlock-free irregular on-chip topologies[C]//2017 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2017: 253-264.
- [86] PARASAR M, JERGER N E, GRATZ P V, et al. Swap: Synchronized weaving of adjacent packets for network deadlock resolution[C]//Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture. 2019: 873-885.
- [87] PARASAR M, FARROKHBAKHT H, JERGER N E, et al. Drain: Deadlock removal for arbitrary irregular networks[C]//2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020: 447-460.
- [88] MURPHY B T. Cost-size optima of monolithic integrated circuits[J]. Proceedings of the IEEE, 1964, 52(12): 1537-1545.
- [89] XU Q, JIANG L, LI H, EKLOW B. Yield Enhancement for 3D-Stacked ICs: Recent Advances and Challenges[C]//IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC), Sydney, Australia, 2012:invited paper.
- [90] CHANG Y C, GONG C S A, CHIU C T. Fault-tolerant mesh-based NoC with router-level redundancy[J]. Journal of Signal Processing Systems, 2020, 92: 345-355.
- [91] LEHTONEN T, WOLPERT D, LILJEBERG P, et al. Self-adaptive system for addressing perm

- anent errors in on-chip interconnects[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 18(4): 527-540.
- [92] CHEN C, FU Y, COTOFANA S. Towards maximum utilization of remained bandwidth in defected NoC links[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 36(2): 285-298.
- [93] KIA H S, ABABEI C. Improving fault tolerance of network-on-chip links via minimal redundancy and reconfiguration[C]//2011 International Conference on Reconfigurable Computing and FPGAs. IEEE, 2011: 363-368.
- [94] REN Y, LIU L, YIN S, et al. A fault tolerant NoC architecture using quad-spare mesh topology and dynamic reconfiguration[J]. Journal of Systems Architecture, 2013, 59(7): 482-491.
- [95] TAHERI E, PASRICHA S, NIKDAST M. ReD: A Reliable and Deadlock-Free Routing for 2.5 D Chiplet-Based Interposer Networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024.
- [96] OUYANG Y, WANG Q, RU M, et al. A novel low-latency regional fault-aware fault-tolerant routing algorithm for wireless NoC[J]. IEEE Access, 2020, 8: 22650-22663.
- [97] NAFFZIGER S, BECK N, BURD T, et al. Pioneering chiplet technology and design for the amd epyc™ and ryzen™ processor families: Industrial product[C]//2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021: 57-70.
- [98] LI H, SHAO C, WANG Z. Detecting Fault Injection Attacks Based on Compressed Sensing and Integer Linear Programming[J]. IEEE Transactions on Dependable and Secure Computing.
- [99] LI H, HU X, SHAO C, ZHOU J, XU G. S EU Reliability Evaluation of 3D ICs[J]. IET Electronics Letters, 2015, 51(4): 362-364.
- [100] ZHANG X, LI H, JIANG L, XU Q. A Low-Cost TSV Test and Diagnosis Scheme Based on Binary Search Method[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2015, 23(11): 2639-2647.
- [101] JIANG L, HUANG X, XIE H, XU Q, LI C, LI X, LI H. A Novel TSV Probing Technique with Adhesive Test Interposer[C]//The 33rd IEEE International Conference on Computer Design (ICCD), New York City, USA, 2015:597-604.
- [102] SAINT-PATRICE D, MALHOITRE S, ASSOUS M, et al. Process Integration of Photonic Interposer for Chiplet-based 3D Systems[C]//2023 IEEE 73rd Electronic Components and Technology Conference (ECTC). IEEE, 2023: 5-12.
- [103] NARAYAN A, THONNART Y, VIVET P, et al. System-level evaluation of chip-scale silicon photonic networks for emerging data-intensive applications[C]//2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020: 1444-1449.
- [104] IFF P, BRUGGMANN B, BESTA M, et al. RapidChiplet: A Toolchain for Rapid Design Space Exploration of Chiplet Architectures[J]. arXiv preprint arXiv:2311.06081, 2023.
- [105] YANG J, ZHENG H, LOURI A. Versa-DNN: A Versatile Architecture Enabling High-Performance and Energy-Efficient Multi-DNN Acceleration[J]. IEEE Transactions on Parallel and Distributed Systems, 2023.

张佳帅 出生于2000年, 硕士研究生在读, 主要研究方向为芯粒间通信、集成芯粒。

杨柳青 硕士学历, 中国科学院深圳先进技术研究院助理工程师, 主要研究方向为车联网、自动驾驶高精地图与高精定位、芯粒集成。

付琦麟 出生于2001年, 大学学历, 研究方向为芯粒集成, 芯粒通信互连测试平台

程慧武 出生于2001年, 硕士研究生在读, 主要研究方向:FPGA, 通用神经网络加速器, 深度学习。

邵翠萍 中国科学院大学博士, 中国科学院深圳先进技术研究院副研究员, 主要研究方向为车规级芯片的功能安全设计及测试技术, 国密芯片设计及车联网安全通信和身份认证研究、无人系统攻击检测与防御。

李慧云, 英国剑桥大学博士, 中国科学院深圳先进技术研究院研究员, 主要研究方向为高性能集成电路芯片设计、系统应用、汽车电子。



JASHAI ZHANG Born in 2000, he is a Master's degree student, and his main research interests are inter-chiplet communication and integrated chiplet.



Li Huiyun, received an M.Eng. degree in electronic engineering from Nanyang Technological University, Singapore, in 2001, and a Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 2006. She is currently a professor at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. Her research interest is high-performance IC chip design, system application, automotive electronics.