#### 引文格式:

张佳帅,杨柳青,付琦麟,等. 基于芯粒架构的通信拓扑结构研究: 进展与挑战 [J]. 集成技术, 2025, 14(3): 1-23. Zhang JS, Yang LQ, Fu QL, et al. Research on communication topologies for Chiplet architecture: progress and challenges [J]. Journal of Integration Technology, 2025, 14(3): 1-23.

## 基于芯粒架构的通信拓扑结构研究:进展与挑战

张佳帅<sup>1,2</sup> 杨柳青<sup>3,4</sup> 付琦麟<sup>3,4</sup> 程慧武<sup>1,2</sup> 邵翠萍<sup>1,2,3,4</sup> 李慧云<sup>1,2,3,4\*</sup> <sup>1</sup>(中国科学院深圳先进技术研究院 深圳 518055) <sup>2</sup>(中国科学院大学 北京 100049) <sup>3</sup>(深圳理工大学算力微电子学院 深圳 518107) <sup>4</sup>(广东省算力微电子重点实验室 深圳 518107)

摘 要 基于芯粒 (Chiplet) 的多芯片集成设计为超越传统系统级芯片的单片集成提供了一种灵活且可 扩展的解决方案。然而, Chiplet 间的通信已成为制约多芯片集成系统整体性能的瓶颈之一。在此背景 下,中介层上网络 (network on interposer, NoI) 在多芯片系统中发挥着至关重要的作用,直接影响多芯 片集成系统的性能和开发成本。本文综述了基于 Chiplet 的 NoI 通信拓扑结构,深入探讨了当前 Chiplet 间通信架构的设计和实现方法,涵盖了从协议层、接口层到应用层的完整通信过程,不仅基于 互连拓扑的形状进行了分类,还对每个类别进行了详细分析和交叉比较。此外,本文还探讨了芯片间 通信技术的未来发展方向,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 可重用中介层和拓扑设计的重要性,旨在为研究人员梳理 NoI 技术现状并展望 NoI 技术的未来发展 趋势。

关键词 芯粒; 拓扑; 硅中介层; 中介层上网络; 片上网络 中图分类号 TP303 文献标志码 A doi: 10.12146/j.issn.2095-3135.20240914001 CSTR: 32239.14.j.issn.2095-3135.20240914001

收稿日期: 2024-09-14 修回日期: 2024-11-18

基金项目: 国家自然科学基金重大项目 (92473112)

作者简介:张佳帅,硕士研究生,研究方向为芯粒间通信、集成芯粒;杨柳青,硕士,助理工程师,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,学士,研究方向为芯粒集成、芯粒通信互连测试平台;程慧武,学士,研究方向为 FPGA、通用神经网络加速器、深度学习;邵翠萍,博士,副研究员,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、无人系统攻击检测与防御;李慧云(通讯作者),博士,研究员,研究方向为高性能集成电路芯片设计、系统应用、汽车电子,E-mail:hy.li@siat.ac.cn。

### Research on Communication Topologies for Chiplet Architecture: Progress and Challenges

ZHANG Jiashuai<sup>1,2</sup> YANG Liuqing<sup>3,4</sup> FU Qilin<sup>3,4</sup> CHENG Huiwu<sup>1,2</sup> SHAO Cuiping<sup>1,2,3,4</sup> LI Huiyun<sup>1,2,3,4\*</sup>

<sup>1</sup>(Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China) <sup>2</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>3</sup>(Faulty of Computility Microelectronics, Shenzhen University of Advanced Technology, Shenzhen 518107, China)

<sup>4</sup>(Guangdong Provincial Key Laboratory of Computility Microelectronics, Shenzhen 518107, China)

\*Corresponding Author: hy.li@siat.ac.cn

**Abstract** Chiplet-based multi-chip integration designs provide a flexible and scalable solution that surpasses traditional system on chip monolithic integration. However, inter-Chiplet communication has become a significant bottleneck affecting overall system performance. The network on interposer plays a pivotal role in multi-chip systems, directly influencing both performance and development costs. This paper reviews the communication topologies of Chiplet-based network on interposer 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 comprehensive overview of the current state of network on interposer technology while simultaneously forecasting its development trends in future multi-chip integrated systems, offering systematic insights to advance frontier research in semiconductor technologies.

Keywords Chiplet; topology; silicon interposer; network on interposer; network on chipFunding This work is supported by Major Program of National Natural Science Foundation of China (92473112)

#### 1 引 言

在半导体设计领域的演进中,系统级芯片 (system on chip, SoC)架构一直被视为集成度和 性能优化的典范。SoC 通过将处理器、存储、输 入/输出功能与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,芯粒(Chiplet)技术提供了一种新的模块化范式,是当前专用集成电路与多芯片模块集成技术之间的一种折中方



图 1 SoC 和 Chiplet 集成技术

Fig. 1 SoC and Chiplet integration technology

案。这3种集成方法的相关参数对比如表1所示。

Chiplet 技术将特定功能分割成独立的可重用 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%)<sup>[1]</sup>,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

#### 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

集成	评估视角									
技术	设计周期	花费/\$	集成度	能效	性能	面积	灵活性	可扩展性	可重用性	设计复杂度
单片集成、 专用集成电路	约 18 个月	>1 000 000	+++	+	++++	+				++++
芯粒	约 12 个月	$1000{\sim}1000000$	++	++	+++	$+\!+\!+$	+++	+++	$+\!+\!+$	++
多芯片模块	约 12 个月	$500000{\sim}5000000$	+++	++	++++	++	++	++	++	++++

注: +表示相关能力的强度, +越多, 技术越成熟。

决方案成为可能<sup>[2-3]</sup>。这一技术的引入促进了半导体产业的创新,为解决摩尔定律放缓和登纳德缩放定律终结带来的挑战提供了新途径。

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、 能耗和面积等方面。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,传统的基于总线的通信结构变得效率低、能 耗高、扩展性差,且仅支持边到边的通信结构, 不适用于多个芯片间的通信。因此, Chiplet 间的 通信逐渐向使用路由节点的中介层上网络 (network on interposer, 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。

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,这进一步推动了可重构及可重用Chiplet和 中介层的发展,使Chiplet通信拓扑成为半导体 设计和制造领域中一个极为重要的研究方向。

现有研究大多聚焦于互联结构和片上网络 (network on chip, NoC) 技术的发展,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 整体且全面的通信结构及流 程的探讨较少,且对工作负载分级下中介层可重 用性的研究尚不充分。Pande 等<sup>[4]</sup>的研究集中于 多核系统级芯片领域,而Yin等<sup>[2]</sup>的工作则侧重 于NoC技术,如模块化路由设计。文献[5-6]虽 然提供了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<sup>[7]</sup>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,该研究在复杂通信拓扑和多 样化工作负载等的性能分析方面仍存在不足。 Das 等<sup>[8]</sup>提出了可重构的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。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,旨在呈现一个全面 的NoI 研究视角。

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 间通信的最新进展,并对当前最前沿 的基于 Chiplet 的 NoI 设计进行了系统总结和对 比,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、微体系结构层面和 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 定量图表进行了技术的总结和对比。

本文结构如下:第2节概述异构 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度对中介 层的拓扑形状和可重用方案进行了详细分类和 分析。第3节介绍了评估和建模的主要模拟器 和工具集,为 NoI 和 NoC 的性能优化提供了 强有力支持。第4节则聚焦于整体通信拓扑 (NoI+NoC),系统总结并分析了能启发整体通信 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5节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。第6节对全文进行了总结。

# 2 异构 Chiplet 架构中的中介层上网络 通信设计

异构 Chiplet 架构的发展对互连技术提出了 严苛要求,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。在该架构中,微凸块 (µbumps) 和硅中介层 中的导线提供 Chiplet 间连接、机械支持和封装 外信号,或通过硅中介层中的硅通孔传输来自 C4 凸块的电源、输入输出 (I/O) 和地线 (ground) 等信号。然而, ubumps 的最小间距限制了每平 方毫米上 Chiplet 的 µbumps 数量。 Goyal 等<sup>[9]</sup> 通过将多个 IP 的单项环链接在一起, 使多个 Chiplet 间的共享端口最大程度地减少 ubumps 的 开销。此外, NoI 技术实现了 Chiplet 间的高密 度、低延迟通信,提供了直接的数据通信路径, 消除了基板上的连接,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,有效解决了多芯片模块 通信的瓶颈。此外,随着 Chiplet 数量的增加, NoI 能保持通信效率,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



#### 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 (Die-to-Die) 芯片互连设计, 如在硅中介层中高速互连的 OIF-CEI-112 G 和 224 G 等 SerDes 系列串行接口 (USR 是 SerDes 中 的一种,用于超短距离,限制了其在大规 模 Chiplet 芯片上的长距离传输),如图 3 (a) 所 示。此外,适用于硅中介层中小数据短距离频繁 移动的 AIB 和 MDIO 技术 (如 Intel 的 Embedded Multi-Die Interconnect Bridge (EMIB)和 FOVEROS), 台积电的 LIPINCON, AMD 的 Infinity Fabric 和 3D V-Cache, 以及 Marvel1的 Mochi Interconnect 等低功耗并行高端口数接口, 如图 3 (b) 所示,也被广泛应用于硅中介层中。 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 Chiplet,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<sup>[10]</sup>。近年来,标准化 D2D 接 口协议的发展,如 Bunch of Wires (BoW)<sup>[11]</sup>和 Universal Chiplet Interconnect Express (UCIe)<sup>[12]</sup> 等,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 宽和低延迟的互连方案,极大地促进了不同厂商





的异构计算资源的集成<sup>[13-14]</sup>。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<sup>[15]</sup>。最近,Feng等<sup>[15]</sup>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 的灵活性和可扩展性,在各种 工作负载下提供显著的性能和能耗改进。异构 接口允许在更广泛的应用中重复使用,显著降低 了非重复性工程 (NRE) 的成本,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,但也为仅使用单一接口的 芯片增加了额外的成本和设计复杂性。

本节整理了上述接口间的差异,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 活性和兼容性方面的发展空间,如表2所示。因 此,多芯片集成系统上的接口可从多个角度进行 改进,具体如下:开发可以在应用前调整串行与 并行带宽比例的异构接口,进一步提高灵活性和 资源利用率;优化能效设计,减少功耗和热量产 生,尤其是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,多芯片集成系统的研究方向还包括如下内容:开发高效路由算法和低延迟通信设计,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,满足数据密集型应用需求;设计高度可扩展的接口,支持更大规模的芯片系统和通信节点;集成安全功能,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<sup>[22]</sup>在名

为HexaMesh的文中,提出了一种 Chiplet μbumps 的分配方法,通过统一 μbumps 的设计尺寸和分 区方式,确保 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。更进一步地,通过模块化图块 分配资源,将 Chiplet 和中介层分为大小相同的 块,每个块包含一组通过 μbumps 实现的预定 义连接,这些连接负责电源、时钟传输、外部 I/O 连接和 Chiplet 间通信,提高了中介层的可重 用性<sup>[23-24]</sup>。

表 2 典型 Chiplet 互连方案之间的比较 (数据获取时间为 2024 年 8 月)

Table 2	Comparison between	typical Chiplet interconnect solutions	(data accessed in 0	8/2024)
---------	--------------------	--	---------------------	---------

规格	机构	最大数据速率/(Gbps·wire <sup>-1</sup> )	延迟/ns	功耗/(PJ·bit <sup>-1</sup> )	传输距离/mm	凸点距离/µm	传输模式	信号类型
SerDes <sup>[16]</sup>	Rambus/OIF	224		0.8	10 (USR) 50 (XSR)		串行	差分
AIB <sup>[17]</sup>	Intel	2 (第一代) 6.4 (第二代)	3.56 (第一代)	0.85 (第一代)	<10 (第一代)	≼55 (第一代)	并行	差分
BoW <sup>[11, 18-19]</sup>	ODSA	32	<2~4 <15 (with FEC)	<0.25~1.0	≤10 (标准) ≤ 50 (加强)	≼150 (标准) ≼55 (加强)	并行	差分
UCIe <sup>[12]</sup>	UCIe Union	32	<2	0.25~1.25	10~25 (标准) 2 (加强)	100~130 (标准) 25~55 (加强)	串行	单端
ACC <sup>[20]</sup>	CCIA	128	6	2.5	1~50	153	串行	差分
LIPINCON <sup>[21]</sup>	TSMC	2.8		0.56	≼0.5		并行	差分

#### 2.2 中介层上网络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, 极大地推进了芯片单元的模块化系统集成,提升 了系统性能,并降低了通信延迟。全连接拓扑虽 然在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。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的需求也相应 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,以及不规则拓扑和混合 拓扑等,以适应不同的应用需求和优化数据处理 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。上述中介层 拓扑设计虽然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<sup>[25]</sup>,而在不同负载下, 其扩展性、可重用性和能效等仍略有不足。因 此,为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和形状 的 NoI 微体系结构,以推动高性能、低成本和可 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,更旨在推动通信策略创新,以 适应现代计算基础设施的不断演进和发展需求。

硅中介层实际上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<sup>[26]</sup>, 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。图 5 中的结构通过适 配 M1、M2、M3 的门极控制信号 (V<sub>C1</sub>、V<sub>C2</sub>、 V<sub>C3</sub>),可以在不同工作模式下切换电路,满足不 同的性能需求 (如低插损、高隔离、高带宽等)。 当开关导通时,晶体管 R<sub>ON</sub> 较小,允许信号顺畅 通过; 而关断时, 晶体管主要表现为 Corr, 能够 阻隔直流和低频信号,但可能耦合高频信号), 不仅能实现更高效的数据路由和电源管理,还能 优化性能,简化 Chiplet 的复杂度,并能显著提 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,但可以提供更灵活和可扩展的解决方 案。若硅中介层的设计可重用于不同规格的同种负 载的 Chiplet 架构,则将有助于降低整体成本、





图 5 一种可重用中介层中 "路由器"开关单元的简化小 信号等效电路<sup>[27]</sup>

# Fig. 5 Simplified small-signal equivalent circuit of the "router" switch unit in a reusable interposer<sup>[27]</sup>

缩短开发周期,使其在高性能应用中的前景更 广阔<sup>[28]</sup>。

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,Feng 等<sup>[29]</sup>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,且仅允许固定 放置。Stow等<sup>[30]</sup>提出的可用于连接Chiplet的硅 中介层服务层为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 Neksus 是一种新颖的、利用中介层特性 的、支持模块化"即插即用"的Chiplet集成架 构,旨在降低系统级封装的制造成本,通过利用 迷你链 IP 连接拓扑上的直接通信解决系统级 封装技术的局限性,但 Neksus 主要专注无源中 介层<sup>[23]</sup>。

早期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,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,作为中介层可重用发展过渡的 SiPterposer<sup>[31]</sup>则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 和可配置的可重用中介层成为新的发展趋势。例 如, Jiao 等<sup>[31]</sup>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,实现微凸点 连接的灵活性。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), 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时,可以抵消有 源中介层成本高的缺陷, 故本文主要关注有源中 介层可重用方面的内容。

2.2.2 多样化的中介层上网络拓扑结构分析

虽然可重用中介层架构取得了一定的成果, 但中介层上 NoI 架构的通用性与性能相互制约。 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 (如计算密集型、数据密集型和 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 发出通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。为促进可重用性和可重构性的 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 (本文直观地对比了 NoI 拓扑结构,如表 3 所示)。

(1) 环形拓扑

环形 (ring) 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<sup>[32]</sup>,适用于小规模的芯片 系统;缺点是单点故障会应影响整个网络,随着 节点数的增加,延迟和带宽受限制。应用于神经 网络的 NN-Baton 架构采用了环状拓扑,考虑了 多种应用于深度神经网络 (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和 Chiplet 粒 度指导的不足<sup>[33]</sup>。无源中介层上的 Monad 架构虽 使用的是环形拓扑,但其能量延迟积 (energy-delay product, EDP) 优于 Simba 和 NN-Baton,依旧为 其应用在计算密集型负载上奠定了基础<sup>[34]</sup>。此 外, POPSTAR<sup>[35-36]</sup>基于光电连接的 Chiplet 间环

表 3 Nol 拓扑结构的直观对比

Table 3	Intuitive	comparison	of NoI	topologies
---------	-----------	------------	--------	------------

拓扑结构	特征	优势	劣势	成本与复杂度
环形	闭合环路,每个节点仅连接到两个 相邻节点	简单,易于实现,降低延迟 复杂性	单点故障,可扩展性随节点数增加 而带宽和延迟增加	低复杂度,低成本
网状	节点通过网格模式连接,适用于中 等规模系统	简单,鲁棒性高,容错,灵 活路由	随网络规模增加, 延迟和能耗增加	中等复杂度,中等成本
环面	结合环形和网格特征,增加额外连 接以提高效率	通信效率和带宽在大规模系 统中更高	连接复杂性增加,实施难度和成本 增加	高复杂度,较高成本
蝶形	多层交换节点,形成类似蝴蝶的 结构	高带宽,低网络直径	高实施复杂性,需要复杂的路由和 管理	非常高复杂度,非常高 成本
不规则	灵活的节点连接,针对特定需求 定制	可为特定高性能任务定制, 可适应	高设计和调试成本,开发工作增加	高复杂度,成本因设计 而异
混合	组合多种拓扑优势,优化数据路径 和带宽	优化延迟和带宽,灵活且 可扩展	设计复杂性高,成本和功耗较高	高复杂度,通常成本较高

形结构和无缓冲多环 (multi-ring) 结构<sup>[37]</sup>都属于通用拓扑。



#### Fig. 6 Typical NoI network topology

(2) 网状拓扑

随着互连 Chiplet 数量的增加,更高基数的 可扩展网状 (mesh) 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,能提供 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 6 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 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<sup>[32]</sup>。Simba 架构通过引入切片减少 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 入研究<sup>[38]</sup>。Wang 等<sup>[39]</sup>结合胖树拓扑,设计了一 种可重新划分的网状 Nol, 可灵活地重新配置为 权重或者特征子网,取代了单一的 NoI 网络,其 额外增加的开销可忽略不计,可在不同 AI 芯片 上重复使用。而为满足大规模 Chiplet 的需求,

克服传统的扁平拓扑结构并不灵活的缺点,Feng 等<sup>[40]</sup>提出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。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,EDP 逊于 INDM,但其在处理数据密集型负载上表现优 异<sup>[41]</sup>。最近,HexaMesh 架构以其紧凑封装和高 扇出特性,通过优化 Chiplet 在网状拓扑上的排 列,扩展到了数百个 Chiplet,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<sup>[22]</sup>。

#### (3) 环面拓扑

环面 (tours) 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 额外的连接提高通信效率。每个节点不仅连接相 邻节点,还连接远端节点,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 扑适用于对吞吐量和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 使用可软件配置的 2D 环面网络进行 Chiplet 间的连接,有效应对了 不规则应用<sup>[42]</sup>。

#### (4) 蝶形拓扑

蝶形 (butterfly) 拓扑是一种高效的 Nol 结构,适用于需要大规模并行计算的应用。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,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<sup>[43]</sup>。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 4 个周围

DRAM 堆栈的基准系统<sup>[43]</sup>。该拓扑略微增加了总 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<sup>[25]</sup>提出了 ButterDonut"错位"拓扑,将路由器 放置在相邻 Chiplet 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。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<sup>[5]</sup>。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,能支持多种工作负载, 同时提供更高的吞吐量和更低的延迟<sup>[5]</sup>。

(5) 不规则拓扑

在网状或环形等常规拓扑可能不适用的情况 下,不规则 (irregularly) 拓扑凭借其灵活性,能 提供更高的性能。不规则拓扑节点间的连接根据 具体需求定制,能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。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 Nol 架 构,其不规则拓扑特性使其能并行运行多个深度 学习 (DL) 工作负载,提高链路利用率<sup>[44]</sup>。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<sup>[6]</sup>。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<sup>[45]</sup>。 GIA 采用模块化的基于瓦片的中介层结构,允许 Chiplet 放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,与其他架构相比,GIA 的性能显 著提升,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实现多种不规则 网络拓扑<sup>[46-47]</sup>。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 层提供了新思路<sup>[48]</sup>。

(6) 混合拓扑

混合 (hybrid) 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。混 合拓扑因其灵活性和高性能而受到广泛关注,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 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,适 应不同工作负载,从而提高系统性能和资源利用 率<sup>[49]</sup>。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,与其他 架构 (如 SIMBA 和 NN-Baton 等)相比,能耗和 延迟显著降低<sup>[50]</sup>。此外, Feng 等<sup>[40]</sup>提出了一种新 的软件定义接口分组方法,该方法将基于 2D 网 状 NoC 的 Chiplet 连接到高基数网络。

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 需综合考虑系统规模、性能需求、成本和实现的 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,本文总结分析了一些典型应用场 景及其对应的拓扑选择标准。

(1) 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、模拟仿真和神 经网络的推理等,强调低功耗和高效率。树形或 蝶形拓扑结构较适合,能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。

(2) 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 环形拓扑能提供多路径通信和高冗余性,因此能 适应大规模数据集的处理。

(3) 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,实现最佳的性能和成本效 益。通过深入理解 NoI 拓扑的分类和应用,研究 人员可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。 本文总结了 NoI 通信架构的相关方法,并进行了 比较,如表4所示。

本节分析表明,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、低能效 和固定拓扑结构的可扩展性差等方面。此外,在 适应不同负载的同时,如何优化 Chiplet 与中介 层的组合,以生成最小且高效的可重用结构,仍 然是一个重要挑战。为应对这些瓶颈,研究人员 可采用动态自适应拓扑结构和智能路由算法,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 系统与中介层上网络架构的性 能评估与优化

第2节对可重用中介层的发展、应用需求、 多样化拓扑和不同工作负载对 NoI 的需求进行了 分析,在此基础上,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, 能更深入地理解 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 通信与中介层上网络架构的仿真工具

评估 NoI 性能需要先进的软件模拟器处理信息,并对 NoI 的各种功能作出评估,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。

Gem5 等 NoC 模拟器可以根据实验要求灵活 地对 Chiplet 通信进行模拟,包括 NoI 和 NoC。 由于 NoI 网络通常受传输延迟、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。

本文比较了当前广泛使用的 Chiplet 技术仿 真工具平台,如表 5 所示 (表中√表示当前列项

#### 表 4 基于 Chiplet 的代表性通信架构比较

#### Table 4 Comparison of representative Chiplet-based communication architectures

文献	名称	拓扑结构	工作负载	死锁自由方法	评估工具	结果
Orenes-Vera 等 <sup>[42]</sup>	DCRA	分层环面	不规则应用、 混合		在 dalorex 功能模拟器的 基础上构建自身的评估 框架	延迟减少最高8倍,能源效率为 网格的92%,但分层环形拓扑将 效率提升19%,成本效率得到改善
Hao 等 <sup>[34]</sup>	Monad	环形	数据密集型		利用 TENET 分析数据重 用情况,利用 accelergy 估 算面积和能量	EDP(能耗延迟乘积) 与 Simba 相比减少 16%, 与 NN-baton 相比减少 30%
Tan 等 <sup>[33]</sup>	NN-Baton	环形	数据密集型		仿真 (Gem5, nvsim), 实际 部署和拓扑设置 (booksim)	与基线网状拓扑相比,延迟减 少 30%, 吞吐量提升 40%
Sharma 禁 <sup>[44]</sup>	SWAP	不规则形	大规模深度学 习,混合型	多目标优化算法	仿真 (booksim)	与基线网络相比,能耗减少 42.4%,延迟减少10.6%, 制造成本降低10.6倍
Bharadwaj 等 <sup>[6]</sup>	Kite	蝶形	计算密集型	最短路径路由与虚拟 通道相结合	仿真 (desnt, heterogarnet)	与基线网络相比,延迟减少7.5%, 吞吐量提高10%;与网状网络相 比,吞吐量提升更是达到17%
Vivet 等 <sup>[32]</sup>	IntAct	网状	HPC, AI 应用, 数据密集型		设计了相应的评估开发板	带宽 3 tbit/(s·mm <sup>2</sup> ), 延迟 0.6 ns/mm
Shao 等 <sup>[38]</sup>	Simba	分层网络	计算密集型	混合虫洞、直通式流量 控制	使用 xilinx FPGA 通过 PCI-E 连接到 x86 主机	能源效率比 V100 提高 5.4 倍,比 T4 提高 2.9 倍;吞吐量比 V100 提高 1.8 倍,比 T4 提高 1.9 倍
Li 等 <sup>[45]</sup>	GIA	不规则形	通用,混合型	可配置路由器、虚拟 通道分配	仿真 (booksim, dsent, ltspice, hotspot, tgff)	与基线网络相比,性能提升为有 源中介层 3.15 倍,无源中介层 60.92 倍;功耗节省为有源中介 层 2.57 倍,无源中介层 2.99 倍
Wang 等 <sup>[39]</sup>	NoI design for agile NN processor customization	网状+胖 树拓扑	AI 工作负 载 (NNP), 混合型	按维度排序	仿真 (scale-sim, booksim2.0, orion3.0)	延迟减少 42.7%~79.5%, 面积开 销减少 26.4% (使用维度有序路由 算法, IO 芯片位于中心)
Iff 等 <sup>[22]</sup>	HexaMesh	六角网格	大型芯粒系统, 混合型		仿真 (booksim2.0)	与基准 2D 网格相比,延迟减少 19%,直径减少 42%,吞吐量 提升 34%
Feng 等 <sup>[40]</sup>	scalable method for efficient Chiplet interconnect design	2D 高基 数网格	多芯粒系统, 混合型	基于 Minus-First 机制 的无死锁自适应路由 算法及安全/非安全 流控策略	仿真 (a Chiplet-specific cycle-accurate C++ simulator)	与基于 Duato 协议的自适应负面优 先路由 2D 网格相比,带宽使用率 提高 20%,能耗最高减少 60%, 延迟减少 15%
Yang 等 <sup>[51]</sup>	Versa-DNN	环形	多 DNN 工作负载,数据密集型	将请求包和回复包分离 到不同的虚拟网络中	FPGA 原型验证 (Xilinx VCU118), Timeloop	与基线网络相比,性能提升 2.5 倍,能源效率提升 30%
Zhang 等 <sup>[50]</sup>	INDM	混合型	DNN 加速器, 计算密集型	XY 路由算法	提出一种用于设计时探索 的 Chiplet 架构 DNN 加 速器延迟估算模型	与基准网络芯片相比,延迟减少 26.93%~79.78%, EDP 减少 26.00%~73.81%
Krishnan 等 <sup>[41]</sup>	SIAM	网状	大型 DNN,数据 密集型		仿真 (booksim, neurosim, ramulator, vampire)	对于 ImageNet 上的 ResNet-50, 该基于 Chiplet 的内存计算架构在 能效方面相比于 NVIDIA V100 和 T4 GPU 分别提升了 130 倍和 72 倍
Green 等 <sup>[48]</sup>	NetSmith	不规则形	通用,混合型	基于表的路由和虚通 道分配	仿真 (parsec, Gem5, heterogarnet, dsent), netsmith framework	与传统规则网络相比,延迟改善 8%~13.5%,吞吐量提升 18%~75%

目在模拟器中存在)。Gem5<sup>[51-53]</sup>及精确建模的 Gem5-Chiplet extension<sup>[54]</sup>支持ARM、x86和MIPS 架构,适用于研究 Chiplet 系统和多种配置。此

外,HeteroGarnet<sup>[55]</sup>作为Gem5的一个重要扩展,支持多时钟域和频率交叉,能模拟多样化的互连系统,这使得它在异构Chiplet设计中尤其

模拟器名称	支持的架构 (类型)	支持芯粒	支持 GPU	功耗模拟	详细计时模拟	系统级模拟	编程语言	额外说明
Gem5 <sup>[51-53]</sup>	ARM,x86,MIPS 等	√ (ViaGem5_Chiplet extension)		$\checkmark$			C++, Python	
Gem5-Chiplet extension <sup>[54]</sup>	ARM,x86,MIPS 等	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	C++, Python	为芯片模拟增强 的 Gem5 扩展
HeteroGarnet <sup>[55]</sup>	多时钟域、跨频率	$\checkmark$		$\checkmark$	$\checkmark$	$\checkmark$	C++, Python	支持多样互连 系统
Sniper <sup>[56]</sup>	主要是 x86	√(Via Chiplet_Sniper)			$\checkmark$		C++	
GPGPU-Sim (shared memory) <sup>[57]</sup>	CUDA GPU	$\sqrt{\text{(Via shared memory} model)}$	$\checkmark$		$\checkmark$		C++	
GPGPU-Sim (message passing) <sup>[58]</sup>	CUDA GPU	$\sqrt{\text{(Via message passing model)}}$	$\checkmark$		$\checkmark$		C++	
SIAM <sup>[41]</sup>	DNN	$\sqrt{\text{(Chiplet-based imc})}$ architecture)		$\checkmark$	$\checkmark$	$\checkmark$	Python, C++	
Chiplet-Sim <sup>[59]</sup>	脉动阵列加速器	$\checkmark$			$\checkmark$		Python	
Muchisim <sup>[48]</sup>	多节点、多芯粒块	$\checkmark$		$\checkmark$	$\checkmark$	$\checkmark$	C++, Python	

表 5	芯粒集成模拟器的技术对比分析

 Table 5
 Comparative analysis of Chiplet simulation platforms

有用。Sniper 通过 Chiplet Sniper 集成,支持 x86 架构<sup>[56]</sup>。GPGPU-Sim 通过共享内存<sup>[57]</sup>和消息传 递<sup>[58]</sup>模型支持 CUDA GPU。Muchisim 专注于多 Chiplet 设计的性能与能耗分析<sup>[48]</sup>。SIAM<sup>[41]</sup>和基 于 SCALE-Sim 的 Chiplet-Sim<sup>[59]</sup>专注于阵列加速 器。Muchisim 可以通过模拟多节点多 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 Nol 延迟的模拟常用 修改后的 BookSim2.0<sup>[60]</sup>进行,并使用 DSENT<sup>[61]</sup> 或者 ORION3.0<sup>[62]</sup>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 能全面评估 Nol 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 组合选择与中介层上网络架构的优 化策略与评估

Chiplet 技术的成功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 和物理布局,以创建高度可定制的集成芯片,这 需要在系统延迟、功耗和成本等多个性能指标间

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 集合为 应用程序提供近乎定制的系统性能<sup>[63]</sup>。多种探索 Chiplet 设计空间的优化方法已被提出,如基于贝 叶斯优化的 HyperMapper 框架,通过探索大型设 计空间优化延迟、功耗、面积和成本等多种指 标<sup>[64]</sup>,可快速迭代多个 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。为进一步提升设计的多目标优化能力, Chopin<sup>[65]</sup>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。同时, Feng 等<sup>[66]</sup>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、重用和 异构性。Pal 等<sup>[63]</sup>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,以便能得到更全面 的最优组合,进一步降低成本。Graening等<sup>[67]</sup>对 影响 Chiplet 架构成本的关键因素进行了量化和 建模,为进一步优化提供了基础。研究人员可以 从前面得到的最优解集中选择最符合设计需求的 方案,对 Chiplet 的设计空间进行细化优化。

在 Nol 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 Nol 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,可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介层。最终,Nol 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 标优化方法对推动 Chiplet 技术的进一步发展至 关重要。

### 4 片上网络与中介层上网络结合架构中 的关键技术分析及应用

本文第3节介绍了对基于 Chiplet 的 NoI 互 连架构的性能评估。在基于有源中介层的系统 中, Chiplet 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<sup>[68]</sup>;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),不仅能实现芯片内部高效通 信,还能在 Chiplets 间进行高速和低延迟的数据 传输。此外,NoC+NoI 架构还广泛应用于高性 能计算系统,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3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。



Fig. 7 Schematic diagram of NoI+NoC topology

#### (1) 死锁解决与路由优化

在构建可重用中介层的 NoI+NoC 结构时, 多个 Chiplets 的互连网络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集成引起的死锁问 题<sup>[69-71]</sup>,因为即使单个 Chiplet 和中介层没有死 锁,集成到同一封装中也可能导致最终死锁<sup>[72]</sup>, 如图 7 中红色箭头所示。因此需要新的方法解决 这一挑战<sup>[2,73]</sup>。常见的 NoC 解决方案可分为避 免、检测与恢复和周期性恢复等方法,可为 NoI+NoC 结构的设计提供参考。

死锁避免策略可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,通过施加限制,适用于处理少量请求,其中 XY 和奇偶路由算法是转向模型的代表<sup>[69-70,74]</sup>; 二是基于流量控制的技术<sup>[41,75-77]</sup>,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<sup>[37]</sup>提出的无缓冲多环 NoC 高 带宽设计,适用于服务器级别 CPU 和 AI 处理 器,提供了良好的扩展性,并实现了死锁避免; Xiang 等<sup>[78]</sup>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; Farrokhbakht 等<sup>[79]</sup>的 Pitstop 方法 通过解决协议和网络级死锁而无须使用虚拟网 络、复杂硬件或错误路由,以低成本方式避免/解 决死锁; Ejaz 等<sup>[80]</sup>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;此后,Srivastava 等<sup>[81]</sup>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,缓解了网络拥塞; Orenes-Vera 等<sup>[42]</sup>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。

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<sup>[70]</sup>。静态气泡利用气泡流控制思想, 通过在任何环形/圆环拓扑中保留一个空闲缓冲区 避免死锁<sup>[82-83]</sup>。恢复机制使用额外的缓冲区或同 步路由器取得前进。Wu等<sup>[73]</sup>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 设计模块化和灵活性等与性能相关的挑 战,进一步提升了性能。

周期性恢复策略 (如 SWAP<sup>[84]</sup>和 DRAIN<sup>[85]</sup> 等)则无须死锁检测,适用于一些特定的场景。 周期性恢复策略通过定期处理 (交换或排空) 潜在 的死锁达到解决死锁的目的。

(2) 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<sup>[86]</sup>。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<sup>[87]</sup>。在芯片的总面积相同 的情况下, Chiplet 分解的越细,良率越高。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。 容错措施主要通过类似 NoC 的容错拓扑设 计和容错路由两方面的优化实现,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提 升系统容错性能,前者通过多路径通信确保可靠 性,如文献[88-91]中提到的备份路径;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,如 Chang 等<sup>[88]</sup>和 Ren 等<sup>[92]</sup>提出的路由 器冗余法。而容错路由通过选择动态路径和基于 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,以绕过故障,如 DeFT 利用基于虚拟网络的方法保证无死锁,同 时增强垂直链接选择,以容忍垂直链接和水平链 接 (HL) 故障的 ReD<sup>[93]</sup>。后者在故障时选择较空 闲路径,以避免过载。

在此基础上,可测试性设计 (design for test, DFT) 技术在 NoI+NoC 架构中也起到了重要作 用,满足了传统测试方法难以满足的高效、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,尤其适合复杂的 NoI+NoC 结构。通过在 Chiplet 和中介层中集成 DFT 结构,系统能对关 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<sup>[94]</sup>。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<sup>[28]</sup>还展示了如何利用中介层的额外"预付 费"硅面积实现容错,同时提高产量和成本效益。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。

(3) 实际应用与性能优化

除了在第2节中介绍不同工作负载 NoI 时提 及的应用场景外, 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,基于 NoC 架构处理芯 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,并通过集成 NoC 高效处理数据,促进了芯片组之间的低延 迟和高带宽通信。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<sup>[95]</sup>。此外,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 和 NoC 架构在现代计算系统中管理复杂通信 的重要性,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 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 同作用使设计者能灵活应对不断变化的技术挑战 和市场需求,优化整个系统的设计和性能,是现 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 和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、使用多个子网络和改 进资源分配,网络的吞吐量得到进一步提升。这 些改进显著增强了可重用中介层的 NoI 设计和 NoI+NoC 架构通信设计的性能。 除了可重用性外,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 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

虽然技术不断进步,但目前的可重用 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。

(1) 中介层灵活性

研究方向涉及中介层灵活放置不同尺寸和数 量 Chiplet 情况下的能力,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 需求不同,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。

(2) 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。由于未来的 AI 和数据中 心具有高并发需求,因此如何确保上述灵活配置 不会引发瓶颈也是亟须解决的问题。

(3) 容错机制

只有建立高效的容错机制,才可确保系统在

发生故障时仍能正常运行。因此需平衡制造成本 与性能损失,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,也可用于故障规避。尽管如此, 建立真正高效的容错机制仍是一个复杂的任务, 需要持续的研究和创新。

(4) 安全性和验证

随着系统规模的增加,不仅需要保持可重用 中介层的高能效,实现网络可扩展性和并行化, 还需解决平面拓扑不能充分利用互连资源的问 题,确保应用增多时的数据传输安全性,考虑 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。例如,Li等<sup>[96-97]</sup>提出一 种准确检测故障注入攻击引起的软错误位置的方 法和一种识别设计时的漏洞并增强单粒子翻转可 靠性的方法。

(5)标准化测试

市场需要统一的标准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,因此需要 在组装前对 Chiplet 和可重用中介层进行测试, 以识别错误的 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 (如硅通孔 检测<sup>[98-99]</sup>),以适应中介层结构,并探索非侵入式 测试方法。

(6) 硅光子中介层

硅光子中介层作为未来的重要研究方向,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<sup>[37,100]</sup>。

(7) 基于 Chiplet 设计的仿真和电子设计自动 化 (electronic design automation, 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、 更全面的仿真和模拟工具的迫切性。这些工具应 能准确模拟不同 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,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。

(8) 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,以实现快速匹配可 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<sup>[101]</sup> 预测工具链正在研发和优化中。

这些方向的深入研究和发展不仅能提升芯片 间的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,加速芯片产品的上市。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

与传统的 SoC 设计相比, Chiplet 异构集成 技术更具灵活性、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 对构建高 性能的系统至关重要。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。此外,本文还总结了当前最 先进的 Nol 架构,并详细探讨了 NoC 架构对 NoI+NoC 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。这些研究不仅能帮助读 者理解这一领域的最新进展和面临的挑战,还能 为进一步优化基于 Nol 和 NoC 的架构提供重要 参考。同时,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

#### 参考文献

- Jangam SC, Pal S, Bajwa A, et al. Latency, bandwidth and power benefits of the SuperCHIPS integration scheme [C] // Proceedings of the 2017 IEEE 67th Electronic Components and Technology Conference, 2017: 86-94.
- [2] Yin JM, Lin ZF, Kayiran O, et al. Modular routing design for chiplet-based systems [C] // Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture, 2018: 726-738.
- [3] Loh GH, Naffziger S, Lepak K. Understanding chiplets today to anticipate future integration opportunities and limits [C] // Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition, 2021: 142-145.
- [4] Pande PP, Grecu C, Jones M, et al. Performance evaluation and design trade-offs for network-onchip interconnect architectures [J]. IEEE Transactions on Computers, 2005, 54(8): 1025-1040.
- [5] Sharma H, Pfromm L, Topaloglu RO, et al. Florets for chiplets: data flow-aware high-performance and energy-efficient network-on-interposer for CNN inference tasks [J]. ACM Transactions on Embedded Computing Systems, 2023, 22(5s): 1-21.
- [6] Bharadwaj S, Yin JM, Beckmann B, et al. Kite: a family of heterogeneous interposer topologies enabled via accurate interconnect modeling [C] // Proceedings of the 2020 57th ACM/IEEE Design Automation Conference, 2020: 1-6.

- [7] Liu YF, Li XY, Yin SY. 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 packagescale interconnects for general-purpose, domainspecific and quantum computing systemsoverview, challenges, and opportunities [J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2024, 14(3): 354-370.
- [9] Goyal V, Wang X, Bertacco V, et al. Neksus: an interconnect for heterogeneous system-in-package architectures [C] // Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium, 2020: 12-21.
- [10] OCP. ODSA Wiki [EB/OL]. [2024-09-14]. https:// www.opencompute.org/wiki/Server/ODSA.
- [11] Farjadrad R, Kuemerle M, Vinnakota B. A bunchof-wires (BoW) interface for interchiplet communication [J]. IEEE Micro, 2019, 40(1): 15-24.
- [12] UCI Express. UCI Express document [EB/OL]. (2024-06-12)[2024-08-15]. https://uciexpress.org/ \_files/ugd/0c1418\_c5970a68ab214ffc97fab16d 11581449.pdf.
- [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] Das Sharma D. UCIe white paper [EB/OL]. (2022-07-25)[2024-06-12]. https://www.uciexpress. org/general-8.
- [15] Feng YX, Xiang D, Ma KS. Heterogeneous dieto-die interfaces: enabling more flexible chiplet interconnection systems [C] // Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture, 2023: 930-943.
- [16] OIF. OIF-FD-CEI-224G-01.0 [EB/OL]. (2022-01-07)[2024-06-12]. https://oiforum.com/wp-content/ uploads/OIF-FD-CEI-224G-01.0.pdf.
- [17] CHIPS Alliance. AIB specification 2.0 [EB/OL].
   (2022-06-17)[2024-06-12]. https://github.com/ chipsalliance/AIB-specification/blob/master/ AIB\_Specification%202\_0.pdf.

未来发展。

- [18] OCP. BoW specification v2.0d [EB/OL]. (2023-03-01)[2024-06-12]. https://www.opencompute. org/documents/bow-specification-v2-0d-1-pdf.
- [19] Ardalan S, Cirit H, Farjad R, et al. Bunch of wires: an open die-to-die interface [C] // Proceedings of the 2020 IEEE Symposium on High-Performance Interconnects, 2020: 9-16.
- [20] 中国 Chiplet 产业联盟. 芯粒互联接口标准-草案
  [EB/OL]. (2022-12-01)[2024-06-12]. https://www.chinahva.com/ACC1.0.pdf.
  IIISCT. Advanced Cost-driven Chiplet Interface
  (ACC 1.0) [EB/OL]. (2022-12-01)[2024-06-12].
  https://www.chinahva.com/ACC1.0.pdf.
- [21] Lin MS, Huang TC, Tsai CC, et al. A 7-nm 4-GHz Arm<sup>1</sup>-core-based CoWoS<sup>1</sup> chiplet design for highperformance 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] // Proceedings of the 2023 60th ACM/IEEE Design Automation Conference, 2023: 1-6.
- [23] Ehrett P, Austin T, Bertacco V. SiPterposer: a fault-tolerant substrate for flexible system-inpackage design [C] // Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition, 2019: 510-515.
- [24] Nasrullah J, Luo ZQ, Taylor G. Designing software configurable chips and SIPs using chiplets and zGlue [C] // Proceedings of the International Symposium on Microelectronics, 2019: 27-32.
- [25] Kannan A, Jerger NE, Loh GH. Enabling interposer-based disintegration of multi-core processors [C] // Proceedings of the 48th International Symposium on Microarchitecture, 2015: 546-558.
- [26] Stow D, Akgun I, Xie Y. Investigation of costoptimal network-on-chip for passive and active interposer systems [C] // Proceedings of the 2019 ACM/IEEE International Workshop on System Level Interconnect Prediction, 2019: 1-8.
- [27] Zheng Y, Shan G, Liu Z, et al. A reusable interposer for multi-chiplet microsystems reconfigura-

tion [C] // Proceedings of the 2023 IEEE MTT-S International Wireless Symposium, 2023: 1-3.

- [28] Stow D, Xie Y, Siddiqua T, et al. Cost-effective design of scalable high-performance systems using active and passive interposers [C] // Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design, 2017: 728-735.
- [29] Feng YX, Ma KS. 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 2.5D/3D integration [C] // Proceedings of the 2016 IEEE/ACM International Conference on Computer-Aided Design, 2016: 1-6.
- [31] Jiao B, Xu L, Yu XY, et al. FPIA: communication-aware multi-chiplet integration with fieldprogrammable interconnect fabric on reusable silicon interposer [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2024, 71(9): 4156-4168.
- [32] 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.
- [33] Tan ZH, Cai HY, Dong RP, et al. NN-Baton: DNN workload orchestration and chiplet granularity exploration for multichip accelerators [C] // Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture, 2021: 1013-1026.
- [34] Hao X, Ding Z, Yin J, et al. Monad: towards costeffective specialization for chiplet-based spatial accelerators [C] // Proceedings of the 2023 IEEE/ ACM International Conference on Computer Aided Design, 2023: 1-9.
- [35] Thonnart Y, Bernabé S, Charbonnier J, et al.

POPSTAR: a robust modular optical NoC architecture for chiplet-based 3D integrated systems [C] // Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition, 2020: 1456-1461.

- [36] Narayan A, Thonnart Y, Vivet P, et al. Systemlevel evaluation of chip-scale silicon photonic networks for emerging data-intensive applications [C] // Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition, 2020: 1444-1449.
- [37] Wang TQ, Feng F, Xiang SL, et al. Application defined on-chip networks for heterogeneous chiplets: an implementation perspective [C] // Proceedings of the 2022 IEEE International Symposium on High-Performance Computer Architecture, 2022: 1198-1210.
- [38] Shao YS, Clemons J, Venkatesan R, et al. Simba: scaling deep-learning inference with multi-chipmodule-based architecture [C] // Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019: 14-27.
- [39] Wang M, Wang Y, Liu C, et al. Network-on-interposer design for agile neural-network processor chip customization [C] // Proceedings of the 2021 58th ACM/IEEE Design Automation Conference, 2021: 49-54.
- [40] Feng YX, Xiang D, Ma KS. A scalable methodology for designing efficient interconnection network of chiplets [C] // Proceedings of the 2023 IEEE International Symposium on High-Performance Computer Architecture, 2023: 1059-1071.
- [41] Krishnan G, Mandal SK, Pannala M, et al. SIAM: chiplet-based scalable in-memory acceleration with mesh for deep neural networks [J]. ACM Transactions on Embedded Computing Systems, 2021, 20(5s): 1-24.
- [42] Orenes-Vera M, Tureci E, Martonosi M, et al. DCRA: a distributed chiplet-based reconfigurable architecture for irregular applications [Z/OL]. arXiv Preprint, arXiv: 2311.15443, 2023.
- [43] Jerger NE, Kannan A, Li ZM, et al. NoC architec-

tures for silicon interposer systems: why pay for more wires when you can get them (from your interposer) for free? [C] // Proceedings of the 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture, 2014: 458-470.

- [44] Sharma H, Mandal SK, Doppa JR, 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.
- [45] Li FP, Wang Y, Cheng YQ, 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.
- [46] Kadomoto J, Irie H, Sakai S. Design of shapechangeable chiplet-based computers using an inductively coupled wireless bus interface [C] // Proceedings of the 2020 IEEE 38th International Conference on Computer Design, 2020: 589-596.
- [47] Kadomoto J, Mitsuno S, Irie H, et al. An inductively coupled wireless bus for chiplet-based systems [C] // Proceedings of the 2020 25th Asia and South Pacific Design Automation Conference, 2020: 9-10.
- [48] Green C, Thottethodi M. NetSmith: an optimization framework for machine-discovered network topologies [Z/OL]. arXiv Preprint, arXiv: 2404.02357, 2024.
- [49] Kim DH, Athikulwongse K, Healy MB, 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.
- [50] Zhang JM, Fan X, Ye YY, et al. INDM: chipletbased interconnect network and dataflow mapping for DNN accelerators [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, 43(4): 1107-1120.
- [51] Yang JQ, 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,

2024, 35(2): 349-361.

- [52] Gem5 Community. Gem5 documentation [EB/OL]. (2025-03-11)[2025-03-11]. https://www.gem5.org/ documentation/.
- [53] Binkert NL, Beckmann BM, Black G, et al. The Gem5 simulator [J]. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1-7.
- [54] GT-CHIPS. Gem5\_chips [EB/OL]. (2019-04-19)
   [2024-07-31]. https://github.com/GT-CHIPS/ gem5 chips.
- [55] Gem5 Community. HeteroGarnet: a detailed simulator for diverse interconnect systems [EB/OL]. (2025-03-11)[2025-03-11]. https://www.gem5.org/ documentation/general\_docs/ruby/heterogarnet/.
- [56] FCAS-LAB. Chiplet sniper [EB/OL]. (2020-10-09)[2024-07-31]. https://github.com/FCAS-ZJU/ chiplet\_sniper.
- [57] FCAS-LAB. Chiplet-Gem5-sharedMemory [EB/ OL]. (2024-06-04)[2024-07-31]. https://github. com/FCAS-SCUT/chiplet-Gem5-sharedMemory.
- [58] FCAS-LAB. Chiplet-GPGPU-Sim-massgepassing [EB/OL]. (2022-09-17)[2024-07-31]. https:// github.com/FCAS-SCUT/chiplet-GPGPU-Simmassgepassing.
- [59] Illbbbyyy. Chiplet-sim [EB/OL]. (2023-09-02)[2024-07-31]. https://github.com/lllbbbyyy/chipl et-sim.
- [60] Jiang N, Becker DU, Michelogiannakis G, et al. A detailed and flexible cycle-accurate network-onchip simulator [C] // Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software, 2013: 86-96.
- [61] Sun C, Chen CHO, Kurian G, et al. DSENT-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling [C] // Proceedings of the 2012 IEEE/ACM sixth International Symposium on Networks-on-Chip, 2012: 201-210.
- [62] Kahng AB, Lin B, Nath S. ORION3.0: a comprehensive NoC router estimation tool [J]. IEEE Embedded Systems Letters, 2015, 7(2): 41-45.
- [63] 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.

- [64] Nardi L, Souza A, Koeplinger D, et al. Hypermapper: a practical design space exploration framework [C] // Proceedings of the 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2019: 425-426.
- [65] Ehrett P, Austin T, Bertacco V. Chopin: composing cost-effective custom chips with algorithmic chiplets [C] // Proceedings of the 2021 IEEE 39th International Conference on Computer Design, 2021: 395-399.
- [66] Feng YX, Ma KS. Chiplet actuary: a quantitative cost model and multi-chiplet architecture exploration [C] // Proceedings of the 59th ACM/IEEE Design Automation Conference, 2022: 121-126.
- [67] 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.
- [68] Zheng H, Wang K, Louri A. A versatile and flexible chiplet-based system design for heterogeneous manycore architectures [C] // Proceedings of the 2020 57th ACM/IEEE Design Automation Conference, 2020: 1-6.
- [69] Dally WJ, Seitz CL. Deadlock-free message routing in multiprocessor interconnection networks
   [J]. IEEE Transactions on Computers, 1987, c-36(5): 547-553.
- [70] Ramrakhyani A, Gratz PV, Krishna T. Synchronized progress in interconnection networks (SPIN): a new theory for deadlock freedom [C] // Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture, 2018: 699-711.
- [71] 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.
- [72] Taheri E, Pasricha S, Nikdast M. DeFT: a dead-

[73] Wu YB, Wang L, Wang XH, et al. Upward packet popup for deadlock freedom in modular chipletbased systems [C] // Proceedings of the 2022 IEEE International Symposium on High-Performance Computer Architecture, 2022: 986-1000.

Conference & Exhibition, 2022: 1047-1052.

- [74] Ebrahimi M, Daneshtalab M. EbDa: a new theory on design and verification of deadlock-free interconnection networks [C] // Proceedings of the 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture, 2017: 703-715.
- [75] Al Faruque MA, Ebi T, Henkel J. Configurable links for runtime adaptive on-chip communication [C] // Proceedings of the 2009 Design, Automation & Test in Europe Conference & Exhibition, 2009: 256-261.
- [76] Ma S, Wang ZY, Liu ZL, et al. Leaving one slot empty: flit bubble flow control for torus cachecoherent NoCs [J]. IEEE Transactions on Computers, 2013, 64(3): 763-777.
- [77] Majumder P, Kim S, Huang JY, et al. Remote control: a simple deadlock avoidance scheme for modular systems-on-chip [J]. IEEE Transactions on Computers, 2020, 70(11): 1928-1941.
- [78] Xiang XY, Sigdel P, Tzeng NF. Bufferless network-on-chips with bridged multiple subnetworks for deflection reduction and energy savings
   [J]. IEEE Transactions on Computers, 2019, 69(4): 577-590.
- [79] Farrokhbakht H, Kao H, Hasan K, et al. Pitstop: enabling a virtual network free network-on-chip [C] // Proceedings of the 2021 IEEE International Symposium on High-Performance Computer Architecture, 2021: 682-695.
- [80] Ejaz A, Papaefstathiou V, Sourdis I. HighwayNoC: approaching ideal NoC performance with dual data rate routers [J]. IEEE/ACM Transactions on Networking, 2020, 29(1): 318-331.

- [81] Srivastava S, Shaikh MA, Shivaneetha G, et al. Intelligent congestion control for NoC architecture in gem5 simulator [C] // Proceedings of the 2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, 2022: 353-360.
- [82] Chen LZ, Pinkston TM. Worm-bubble flow control [C] // Proceedings of the 2013 IEEE 19th International Symposium on High Performance Computer Architecture, 2013: 366-377.
- [83] Ramrakhyani A, Krishna T. Static bubble: a framework for deadlock-free irregular on-chip topologies [C] // Proceedings of the 2017 IEEE International Symposium on High Performance Computer Architecture, 2017: 253-264.
- [84] Parasar M, Jerger NE, Gratz PV, 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.
- [85] Parasar M, Farrokhbakht H, Jerger NE, et al. DRAIN: deadlock removal for arbitrary irregular networks [C] // Proceedings of the 2020 IEEE International Symposium on High Performance Computer Architecture, 2020: 447-460.
- [86] Murphy BT. Cost-size optima of monolithic integrated circuits [J]. Proceedings of the IEEE, 1964, 52(12): 1537-1545.
- [87] Xu Q, Jiang L, LI HH, et al. Yield enhancement for 3D-stacked ICs: recent advances and challenges [C] // Proceedings of the 17th Asia and South Pacific Design Automation Conference, 2012: 731-737.
- [88] Chang YC, Gong CSA, Chiu CT. Fault-tolerant mesh-based NoC with router-level redundancy [J]. Journal of Signal Processing Systems, 2020, 92: 345-355.
- [89] Lehtonen T, Wolpert D, Liljeberg P, et al. Selfadaptive system for addressing permanent errors in on-chip interconnects [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 18(4): 527-540.

- [90] Chen CL, Fu YW, 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.
- [91] Kia HS, Ababei C. Improving fault tolerance of network-on-chip links via minimal redundancy and reconfiguration [C] // Proceedings of the 2011 International Conference on Reconfigurable Computing and FPGAs, 2011: 363-368.
- [92] Ren Y, Liu LB, Yin SY, 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.
- [93] Taheri E, Pasricha S, Nikdast M. ReD: a reliable and deadlock- f ree routing for 2.5-D chipletbased interposer networks [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, 43(12): 4599-4612.
- [94] Ouyang YM, Wang Q, Ru MX, et al. A novel low-latency regional fault-aware fault-tolerant routing algorithm for wireless NoC [J]. IEEE Access, 2020, 8: 22650-22663.
- [95] Naffziger S, Beck N, Burd T, et al. Pioneering chiplet technology and design for the amd epyc<sup>™</sup> and ryzen<sup>™</sup> processor families: industrial product [C] // Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer

Architecture, 2021: 57-70.

- [96] Li HY, Shao CP, Wang Z. Detecting fault injection attacks based on compressed sensing and integer linear programming [J]. IEEE Transactions on Dependable and Secure Computing, 2019, 16(3): 476-483.
- [97] Li HY, Hu XB, Shao CP, et al. SEU reliability evaluation of 3D ICs [J]. IET Electronics Letters, 2015, 51(4): 362-364.
- [98] Zhang XL, Li HY, Jiang L, et al. 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.
- [99] Jiang L, Huang XW, Xie HF, et al. A novel TSV probing technique with adhesive test interposer [C] // Proceedings of the 2015 33rd IEEE International Conference on Computer Design, 2015: 597-604.
- [100] Saint-patrice D, Malhouitre S, Assous M, et al. Process integration of photonic interposer for chiplet-based 3D systems [C] // Proceedings of the 2023 IEEE 73rd Electronic Components and Technology Conference, 2023: 5-12.
- [101] Iff P, Bruggmann B, Besta M, et al. Rapidchiplet: a toolchain for rapid design space exploration of chiplet architectures [Z/OL]. arXiv Preprint, arXiv: 2311.06081, 2023.