计算机研究与发展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(11): 2651−2662, 2025

本期推荐

芯粒互联技术综述

王 浩[1,2] 王 勇[2] 冯长磊[2] 盖伟新[1] 吴 鹏[2] 钱 江[1,2]
[1](北京大学集成电路学院 北京 100871)
[2](北京微电子技术研究所 北京 100076)
(ahao@pku.edu.cn)

Review of Chiplet Interconnection Technology

Wang Hao[1,2], Wang Yong[2], Feng Changlei[2], Gai Weixin[1], Wu Peng[2], and Qian Jiang[1,2]
[1](*School of Integrated Circuits*, *Peking University*, *Beijing* 100871)
[2](*Beijing Microelectronics Technology Institute*, *Beijing* 100076)

Abstract As the breaker of Moore's Law, Chiplet technology has high expectations in the integrated circuit industry. Chiplet technology can combine multiple small chips with specific functions into a Chiplet integrated chip through high-speed interconnection technology, whose core technology is the Chiplet interconnection technology that can achieve Chiplet combination and expansion. We analyze and discuss the Chiplet interconnection protocol, interconnection architecture, fault-tolerant mechanism, typical interconnection Chiplets, and testability design based on the interconnection Chiplet. Firstly, we provide a detailed comparison and analysis of domestic and foreign Chiplet interconnection protocols, and provide the layers and functions of each protocol. Secondly, we introduce three typical Chiplet interconnection architectures, and analyze the characteristics and advantages of each architecture. Afterwards, the Chiplet fault-tolerant mechanism is introduced, including fault-tolerant encoding of interconnection interfaces, fault-tolerant topology, and fault-tolerant routing. Then, three types of interconnection Chiplet design schemes are presented, including programmable interconnection Chiplets, path programmable interconnection Chiplets, and fully customized interconnection Chiplets. Finally, a testability design testing scheme based on the interconnection Chiplet is introduced. This paper focuses on Chiplet interconnection and aims to help readers have a systematic understanding of Chiplet interconnection technology.

Key words Chiplet; interconnection; interconnection Chiplet; architecture; fault tolerance; design for testability

摘 要 作为摩尔定律的“破局者”,芯粒(Chiplet)技术被业界寄予了厚望.芯粒技术能够将多个具有特定功能的“小芯粒”通过高速互联技术组合成一个“小芯粒”集成芯片,其技术核心是能够实现芯粒组合扩展的芯粒互联技术.从芯粒互联协议、互联架构、容错机制、典型互联芯粒、基于互联芯粒的可测性设计5个方面进行了分析与讨论.首先详细对比分析了国内外芯粒互联协议,给出了各协议的分层及功能.然后介绍了3种典型的芯粒互联架构,分析了各种架构的特点及优势.之后介绍了芯粒容错机制,介绍了互联接口容错编码、容错拓扑和容错路由等容错途径.接着给出了可编程互联芯粒、路径可编程互联芯粒以及专用互联芯粒3种设计方案.最后介绍了基于互联芯粒的可测试性设计与测试方案.以芯粒互联为主题,旨在帮助读者对芯粒互联技术进行系统性了解.

关键词 芯粒;互联;互联芯粒;架构;容错机制;可测性设计

中图法分类号 TP302

收稿日期:2024-07-01;**修回日期**:2025-02-18

通信作者:王勇(wangy@mxtronics.com)

DOI: 10.7544/issn1000-1239.202440585 CSTR: 32373.14.issn1000-1239.202440585

当前,集成电路产业已步入“后摩尔时代”[1-4],先进制程工艺逼近物理极限,摩尔定律呈现放缓趋势.单芯片算力逐渐无法满足人工智能、自动驾驶、大数据、云计算等国家战略应用对边缘算力日益激增的需求.随着物联网时代进入新发展时期,市场对芯片的功能多样化、开发敏捷化要求不断提升,对其集成度的要求也进一步提升,集成芯片技术发展面临面积墙、互联墙、存储墙和成本墙的问题.作为摩尔定律的“破局者”,芯粒(Chiplet)技术被业界寄予了厚望.芯粒技术是一种集成电路设计制造新方法,将多个具有特定功能(计算、存储、互联、加速等)的“小芯粒”通过芯粒间(die-to-die, D2D)高速互联技术组合到一起,再通过先进封装技术(如2.5D, 3D封装)集成封装在一起,从而建立“小芯粒”集成芯片[5-7].芯粒技术的核心特征是具有极高的互连密度和带宽,具有良率高、成本低、灵活度高、迭代周期短的特点.

芯粒技术旨在建立一个健康开放的行业生态,它非常注重芯粒系统的可扩展性和灵活性.然而当前芯粒技术面临诸多挑战,首先需要实现协议标准的统一.当前芯粒互联协议种类繁多、技术细节获取困难、兼容性难以统一.芯粒间高速互联是芯粒技术的核心[8-11],它指的是通过高带宽、低延迟的片间互联接口或互联芯粒实现芯粒的组合扩展.当前,芯粒互联技术主要掌握在Intel, AMD, ARM等厂商及协议制定者手中.D2D互联接口信号多、传输速率高,对设计技术、制造工艺、验证评估能力等均提出了较高的要求.此外,集成度的大幅度提升带来了良率下降、可访问性下降、测试覆盖率下降等问题,芯粒及集成芯片的测试问题[12]也成为了芯粒技术领域亟待解决的关键核心问题.

2023年3月,集成芯片前沿技术科学基础专家组提出了集成芯片十大技术难题[13],为当前集成芯片的发展路线提供了牵引和指明了方向.本文以芯粒互联技术为主题,聚焦“多芯并行体系结构和互联接口”和“芯粒的测试和容错问题”两大技术难题,从芯粒互联协议、互联架构、容错机制、典型互联芯粒、基于互联芯粒的可测性设计5个方面进行了分析与讨论.

1 芯粒互联协议

芯粒技术的关键是生态建设,竞争的焦点是互联协议标准[14-17].在IC China 2020大会上,互联和封装引起了专家学者的足够重视,被确定为芯粒领域的关键技术,其中芯粒互联需要满足协议的一致性,因此亟需制定统一的芯粒互联协议标准.当前国内外有大量的机构团体致力于芯粒接口的标准化工作,短时间出现了大量芯粒互联协议.

2022年3月,集成电路行业十大巨头(Intel、AMD、ARM、高通、台积电等)联合成立了通用芯粒互联技术(Universal Chiplet Interconnect Express, UCIe)产业联盟,并正式推出了UCIe协议标准[14],旨在定义一个开放的、可互操作的芯粒生态系统标准,满足业内需要统一的芯粒互联标准.目前“UCIe”得到了x86和ARM阵营的广泛支持,已经成为国际主流的芯粒互联标准.

如图1所示UCIe协议是一个分层协议,每一层都执行不同的功能.协议层支持PCIe、CXL以及流协议,充分利用现有技术生态保证无缝的互操作性.D2D适配层协调协议层与物理层之间的数据传输,为芯粒提供链路状态管理和参数协商,通过选择循环冗余校验(cyclic redundancy check, CRC)和链路级重试保证传输数据的可靠性.物理层负责管理电信号、时钟、链路训练、边带等,采用边带、DDR前向时钟设计使得在单应用场景下的模块设计复杂度相对更低,模块验证更加容易.

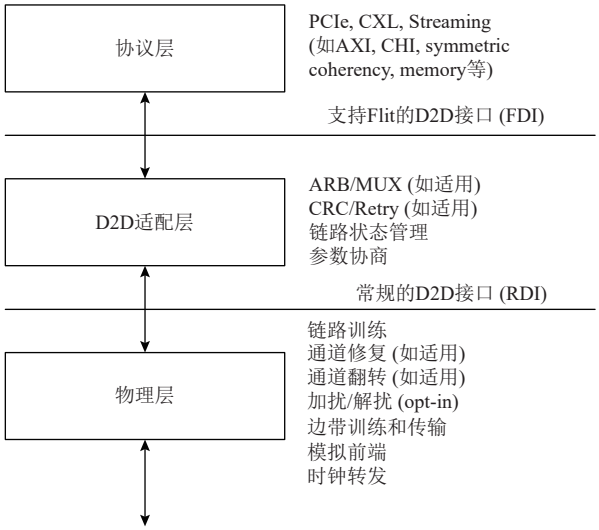


Fig. 1 Layering and functions of UCIe

图1 UCIe的分层及功能

2021年6月,中华人民共和国工业和信息化部立项了《小芯片接口总线技术要求》[15],由中国科学

院计算技术研究所、中华人民共和国工业和信息化部电子工业标准化研究院和国内多个芯片厂商联合组建的中国计算机互联技术联盟(China Computer Interconnect Technology Alliance, CCITA)合作开展了国内首个芯粒互联协议的制定工作，协议于2022年12月在第2届中国互连技术与产业大会上正式对外发布，结合当前我国技术发展和应用的现状，制定和应用计算机系统芯片内、芯片间、系统间互联技术的协议规范和标准.

《小芯片接口总线技术要求》规定的互联接口层次结构如图2所示，该规范支持flit包格式的数据传输，支持计算到存储(computing to memory, C2M)、计算到计算(computing to computing, C2C)、计算到输入输出(computing to input/output, C2IO)、计算到其他(computing to others, C2O)四种应用场景，支持MCM, RDL, CoWoS, Silicon Bridge这4种封装方式，支持并行单端、串行单端和串行差分3种物理层接口，接口适应范围更广且更具灵活性.

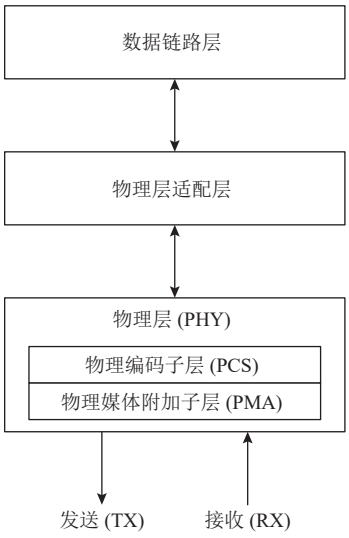


Fig. 2 System hierarchy structure of interconnected interfaces

图2 互联接口的体系层次结构

2020年9月，中国芯粒产业联盟(China Chiplet league, CCLL)宣布启动成立. 2023年2月，中国芯粒产业联盟、国内外IP厂商、国内领先系统与应用厂商联合发布了《芯粒互联接口标准》(Advanced Cost-driven Chiplet Interface, ACC)[16]，该标准由交叉信息核心技术研究院牵头，由中国芯粒产业联盟共同起草制定.

《芯粒互联接口标准》的层次结构如图3所示，该协议是32 Gbps以上大带宽的高速串行接口规范，侧重于国产基板及封装供应链优化和适用性，以成

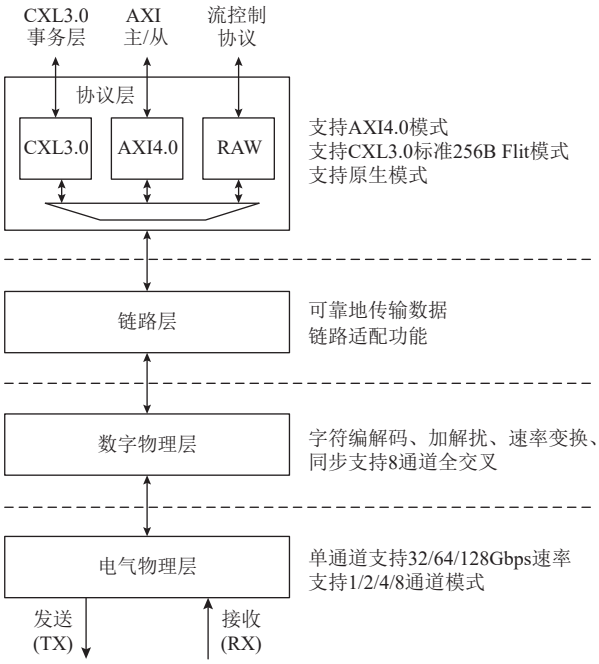


Fig. 3 Hierarchical structure of ACC standard

图3 ACC标准的层次结构

本可控及商业合理性为核心导向，因此其成本更低、产能更充足稳定. 该规范支持AXI4.0、CXL3.0以及原生模式，支持C2C, C2IO这2种应用场景，链路层支持自动检测、出错重传2种模式，可以确保传输数据的高可靠性.

国内外芯粒互联协议对比如表1所示，通过对比可以看出，各种协议均具有各自的优势. UCIe协议支持PCIe/CXL的无缝对接，利用PCIe的强大生态由板级互联扩展到封装内部芯粒间互联，不仅支持PCIe向CXL扩展，还支持用户自定义流协议模式；该协议具有更低的接口传输延迟和数据传输功耗，在功耗与性能平衡方面更具优势. 虽然UCIe协议是当前最为完善的芯粒互联协议，但它仍处于发展和完善阶段，也存在缺乏完整定义的问题.《小芯片接口总线技术要求》支持并行单端、串行单端、串行差分3种物理层接口，提出了多种速率要求，能够灵活应对更为广泛的应用场景和不同能力的供应商，兼顾了对PCIe等现有协议的支持；不足之处是该协议存在延迟较大的问题.《芯粒互联接口标准》支持更高的传输速率；采用串行差分数据传输方式，管脚数少；支持更为广泛的链路报文格式；不足之处是该协议的标准化和技术成熟度仍需进一步完善.

当前，UCIe协议正逐渐赢得国内外主流IC厂商的认可，它主要面向高性能计算领域，其完善程度、规范程度都相对较高，更新速度快，从2022年UCIe1.0

Table 1 Comparison of Interconnection Protocols for Domestic and International Chiplet

表 1 国内外芯粒互联协议对比

对比项	UCIe	小芯片接口总线技术要求	芯粒互联接口标准
国家	美国	中国	中国
制定者	UCIe 联盟	中国计算机互连技术联盟（CCITA）	中国芯粒产业联盟（CCLL）
发布年份	2022	2022	2023
应用场景	C2C，C2IO	C2M（后续）、C2C、C2IO、C2O	C2C，C2IO
协议层次	协议层、D2D 适配层、物理层	数据链路层、适配层、物理层	协议层、链路层、物理层
兼容性	支持 PCIe6.0、CXL2.0、流协议	支持 flit 包格式传输	支持 AXI4.0、CXL3.0、原生模式
链路报文格式（大小等）/B	68，256	70，256	128，256，384，512，640
物理层种类	并行单端	并行单端、串行单端、串行差分	串行差分
传输速率/GTps	4，8，12，16，24，32	2.5，5，8，16，32	32，64，128
接口位宽	$K\times N$	$16\times2\times N$（并行）、$4\times N$（串行）	$8\times N$
传输距离/mm	≤10	≤25	≤50
误码率	1E−15(传输速率≥12GTps) 1E−27(传输速率≤8GTps)	1E−15	1E−15，1E−12，1E−9
延迟/ns	≤2	≤13	≤6
功耗/（pJ/b）	0.25~1.25	1.5，2.5	2.5
封装方式	2D，2.5D	2D，2.5D	2D，2.5D

发布至今已实现 2 个版本(UCIe1.1, UCIe2.0)的更新，生态建设与推广势头迅猛. 针对高性能计算、国产基板及封装供应链等特定应用场景，国内芯粒组织/团体也纷纷提出了符合国内芯片工艺制程条件的芯粒互联协议，这对于短期内解决芯粒国产化问题具有重要意义. 然而，大到研究团体/组织，小到部门，纷纷提出了自己的协议，使得国内短时间涌现了大量的芯粒互联协议. 这些协议一定程度上反映了各个组织/团体对于芯粒技术的指标需求，但是过量的协议会阻碍国内芯粒生态的形成，而且部分协议过于精简，并不能发挥较好的指导示范作用. 因此，未来国内亟需制定一套统一且完备的芯粒互联标准，充分吸收国内现有主流协议的优点，摒弃不合理的部分，在满足国内芯粒产业发展需求的同时，尽量兼容国外主流协议标准(UCIe 协议)，做到接口适配和协议兼容，实现国内统一且与国际接轨的目标.

2 芯粒互联架构

本节主要介绍典型芯粒互联架构及其工作原理.

当前芯粒互联有 3 种典型的架构，分别是多芯粒架构[17]、中心 IO 芯粒架构[18-19]、侧接口芯粒架构[20].

多芯粒架构[17]的典型产品包括 AMD 的 EPYC 一代处理器、Intel 的 Sapphire Rapids 和 Emerald Rapids 等，多芯粒架构设计简单，通过 D2D 互联接口连接多个相同的芯粒，可以实现高性价比的芯粒组合.

AMD 的 EPYC 一代处理器架构如图 4 所示，它集成了 4 颗同构计算芯粒，每颗芯粒包含 8 个 Zen 内核，采用了 Global Foundries 的 14 nm 制程工艺，实现了 32 核 64 线程. 相对于 SoC 实现方式，第 1 代 EPYC 处理器采用芯粒的实现方式，虽然带来了 10% 的额外面积开销，但是提高约 20% 的产出率，节约成本约 59%.

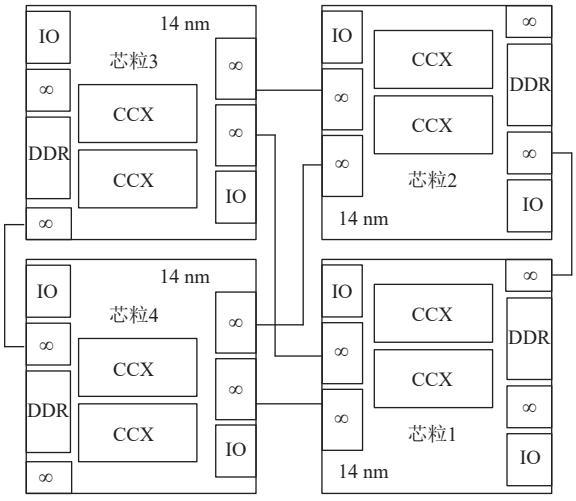


Fig. 4 AMD processor based on multi-Chiplet architecture[17]

图 4 基于多芯粒架构的 AMD 处理器[17]

中心 IO 芯粒架构[18-19]的典型产品包括 AMD 霄

龙二代处理器、Intel Ponte Vecchio 等. 中心 IO 芯粒架构的实现基于先进封装，其所有芯粒都通过中心 IO 芯粒实现集中互联，可减少多芯粒架构中各芯粒间直接互联引入的大密度布线，显著提升芯粒间互联性能. 中心 IO 芯粒对加工制程不敏感，可选择较低的加工制程，降低量产的成本.

如图 5 所示，以 AMD EYPC 二代处理器为代表的“功能芯粒+互联芯粒”架构成为芯粒互联的典型架构. 它采用 1 颗 14 nm 制程的 IO 芯粒（IO die, IOD）实现 8 颗 7 nm 制程的 CPU 计算芯粒（CPU compute die, CCD）的互联. IF（Infinity Fabric）是 AMD 公司和 IBM 公司为片上片间高效互联设计的并行互联总线接口，是一种用于实现 IOD 与 CCD 以及 CCD 与 CCD 之间互联的缓存一致性总线，是实现算力集成所必须的. IOD 与所有 CCD 平等互联，带宽、延迟均保持一致，易于集成芯片的配置管理.

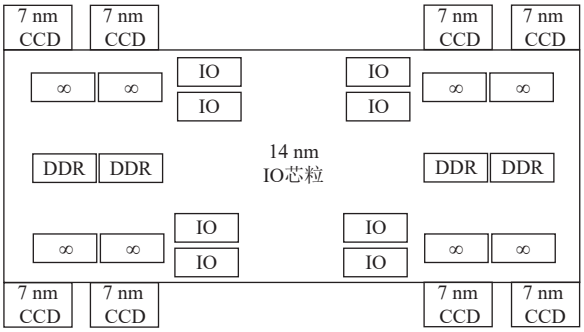


Fig. 5 AMD processor composed by CCD and IOD[18-19]

图 5 由 CCD 和 IOD 组成的 AMD 处理器[18-19]

侧接口芯粒架构典型产品包括华为昇腾 910 处理器、AMD 的 RX7000、NVIDIA 的 Ampere 等. 相对于中心 IO 芯粒架构，侧接口芯粒架构的侧面存在一个接口芯粒，通过该接口芯粒实现与主要功能芯粒互联以及对外互联扩展，通过主要的功能芯粒实现与其他功能芯粒的互联.

华为昇腾 910 处理器架构[20] 如图 6 所示，该处理器包含 8 个芯粒、4 个 HBM 芯粒、2 个 Dummy 芯粒、1 个 SoC 芯粒、1 个 Nimbus 芯粒. HBM 芯粒的总带宽可达 1.2 TBps, Dummy 芯粒用来保持芯片的整体机械应力平衡. SoC 芯粒采用了 TSMC 7 nm 工艺，通过一个 4×6 的 2 D mesh 片上网络（network on chip, NoC）实现对功能组件的互联，功能组件包含 32 个 Ascend-Max 内核、16 个 Arm V8 TaiShan CPU 内核和 CPU LCC、视频编解码器. Nimbus 芯粒实现与 SoC 芯粒以及处理器外部的扩展互联.

典型互联架构的对比如表 2 所示，多芯粒架构将

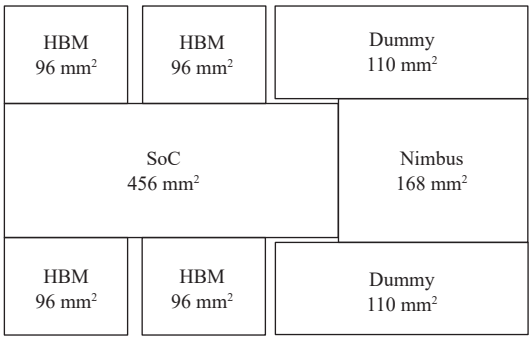


Fig. 6 Huawei Ascend 910 processor[20]

图 6 华为昇腾 910 处理器[20]

多个独立的芯粒组合在一起，形成更大的系统，适用于高度灵活性和可扩展性的应用场景，比如高性能计算、人工智能加速器、大数据处理等. 此外，多芯粒架构可以通过并行处理，提高整体的计算能力，其生产成本相对较低. 然而，多芯粒架构也面临一些挑战，比如为了保证芯粒间的两两互联，需要在芯粒之间或芯粒内部进行复杂的布线规划，可能导致较高的布线密度，带来芯粒之间的信号传输延迟和功耗问题，此外如何有效管理调度芯粒等也是重要的挑战.

Table 2 Comparison of Typical Interconnected Architectures

表 2 典型互联架构对比

类别	多芯粒架构	中心 IO 芯粒架构	侧接口芯粒架构
应用场景	高性能计算、人工智能加速器、大数据处理等	数据中心、高性能计算等	高速数据传输、高性能 GPU 等
典型产品	EPYC (7001), Sapphire Rapids, Emerald Rapids 等	EPYC (7002，7003，9004)，MI300，Ponte Vecchio 等	昇腾 910、RX7000、Ampere 等
互联能力	最弱	最强	适中
通信效率	较低	最高	较低
生产成本	最低	最高	较高
布线密度	最高	低	低
面积大小	最大	较小	最小
延迟	最高	低	低
功耗	最高	低	低

中心 IO 芯粒将所有 IO 互联功能集中在该芯粒上，将计算功能分布到其他功能芯粒上，减少了芯粒间信号的传输距离，降低了延迟和功耗，适用于对延迟和功耗有严格要求的场景，比如数据中心、高性能计算等. 然而，中心 IO 芯粒位于中心位置，需要与周围的功能芯粒进行互联，会导致其周围布线密度较

高, 采用先进的封装工艺实现芯粒互联会造成较高的生产成本.

侧接口芯粒架构将功能芯粒通过侧向接口连接, 提高了系统的灵活性和可扩展性, 每个芯粒都可以独立进行输入输出操作, 减少了中心 IO 芯粒的负担. 然而, 侧接口芯粒架构也面临着接口一致性和兼容性以及如何管理这些独立的侧向接口问题. 因此, 选择哪种架构取决于对性能、成本、设计复杂度以及扩展性的综合考量, 如果注重成本和简化设计, 多芯粒架构可能是合适的选择; 如果更注重高性能、高灵活性和可扩展性, 中心 IO 芯粒架构和侧接口芯粒架构可能是更优的选择.

3 芯粒容错机制

本节主要介绍典型芯粒互联的容错技术.

3.1 互联接口容错编码

UCIe 协议最高支持每通道 32 GTps 的传输速率, 国内芯粒互联协议最高可支持每通道 128 GTps 的传输速率, 为确保芯粒间的可靠数据传输, UCIe 协议中给出了 2 种主要模式: 循环冗余校验(CRC)+重传、ECC 纠错. 循环冗余校验是通过对发送端数据添加校验位来实现对传输数据引入的错误检测. 前向纠错编码(FEC)能够通过加入冗余实现错误纠正, 提高传输数据的可靠性.

常见的 FEC 编码有 BCH 码[21]、RS(Reed-Solomon)码[22-24]、LDPC 码[25]、Polar 码[26] 等, 其中 BCH 码、RS 码具有译码延时短、纠错性能好、硬件吞吐率高等优点, 适合应用于芯粒间互联容错场景. 对于芯粒间超高速数据传输应用场景, 信号在传输过程中同样会受到更复杂的噪声干扰, 导致误码率(BER)增加. FEC 技术成为保证数据可靠性的关键技术之一. 国际上, 多家领先的半导体公司及研究机构已经在 FEC 领域取得了显著进展. 例如, IEEE 802.3 标准已经包含了多种 FEC 算法, 用以保证高速传输速率下的信号可靠性. 然而, 现有的面向芯粒互联的 FEC 技术在面对更高速率、更高可靠度和更低延迟的需求时, 仍面临着巨大的挑战.

RS 码技术具备很好的适配 UCIe 协议容错模式的特性, 其典型译码器关键模块包括: 校验子计算(syndrome calculation, SC)、关键方程求解(key equation solver, KES)、钱搜索和福尼算法(Chien search and Forney algorithm, CSFA). 校验子计算模块能够判断接收码字中是否引入错误, 如果校验子全 0, 则证明接收码字是正确的, 否则引入了错误. 通过对比钱搜索得到的根的个数是否等于错误位置多项式的阶数来判断最终译码是否成功. 图 7 为一种可应用于芯粒互联的 RS 码译码器架构, 码字判决在完成校验子计算之后判决码字是否出错, 重传判决在钱搜索求解出错误位置多项式的根之后判决码字是否能够译码成功. 如果不能成功, 则请求发送端进行数据重传.

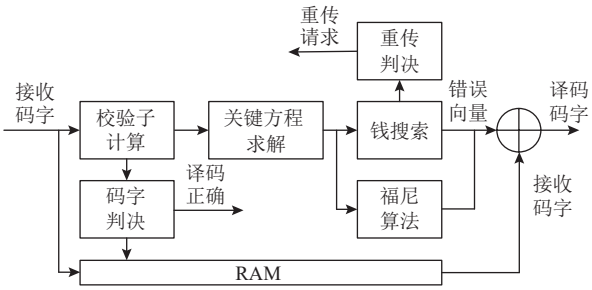


Fig. 7 Reed-Solomon decoder architecture for Chiplet interconnection

图 7 面向芯粒互联的 Reed-Solomon 译码器架构

3.2 容错拓扑和容错路由

在集成芯片的互连网络中, 单芯粒集成了海量的晶体管, 在先进制程下芯粒故障率高, 永久性故障会给集成芯片系统带来性能损失, 采用容错拓扑和容错路由[27-33] 在一定程度上可提升系统的容错性能.

容错拓扑是指针对芯粒互联设计所采用能够容忍故障与提供冗余路径的物理布局和连接方式, 它对于提升网络系统的容错性能有相当重要的作用. 典型的容错拓扑有 2 条实现途径: 1)采用冗余网络, 通过多条路径建立芯粒间连接关系, 当某条链路发生故障时, 通过冗余通信路径实现数据通信, 确保高可靠、无障碍的数据传输. 2)采用高连接性网络, 对于重要节点须确保其具有较大的节点基数, 为互连网络提供更多路径选择, 提升系统容错性能, 次要节点可减少其节点基数, 降低硬件资源开销.

容错路由是芯粒互联设计中针对永久性故障导致的网络变化提出的路由算法, 用以提升集成芯片系统的容错能力. 当片上网络出现故障时, 容错路由算法需具备一定的网络适应能力, 能够自适应执行路由策略, 绕开故障区域进行通信. 容错路由算法可以依据以下方法: 1)动态路径选择算法可以根据实时的网络状况选择最佳路径绕过故障区. 路由算法依据链路状态、节点负载、延迟等信息做出路由决策, 将数据流导向可用路径. 2)基于负载均衡的路由算法可以在互连网络出现故障时使用负载均衡策略来选择最佳路径, 根据节点的负载情况, 通过较空闲

的路径实现通信，避免更多的数据流导向过载区域或故障区域.

现有芯粒互联容错技术的对比如表 3 所示. 容错编码主要适用于芯粒间数据传输场景，通过添加冗余信息增强接口数据传输的抗误码能力，因此会牺牲一定的编码效率来换取高编码增益. 容错拓扑和容错路由主要适用于集成芯片的芯粒间互联. 其中，容错拓扑通过采用高连接性网络和添加冗余网络的方式来提高容错率，但也增加了冗余节点和链路；容错路由通过执行路由策略和绕开故障区域的方法规避故障，其延迟较高，同时也增加了硬件开销.

Table 3 Comparison of Fault Tolerant Technologies for Chiplet Interconnection

表 3 芯粒互联容错技术对比

类别	容错编码	容错拓扑	容错路由
适用场景	芯粒间数据传输	集成芯片的芯粒互联	集成芯片的芯粒互联
容错方法	添加冗余信息	采用高连接性网络，添加冗余网络	执行路由策略，绕开故障区域
容错性能	编码增益高	容错率高	故障规避率高
代价	牺牲编码效率	增加冗余节点和链路	增加硬件开销
延迟	低	低	高

4 典型互联芯粒

互联芯粒[34]可用于芯粒扩展、片间级联和数据传输，可实现对不同工艺节点以及不同材质功能芯粒的集成，适用于高性能计算、人工智能、航空航天、硬件加速等场景. 它通常应具备数据交换、协议适配、系统管理、芯粒测试、集成芯片测试、缓存一致性中的部分或全部功能，支持快速组建集成芯片产品. 为满足芯粒间高速、高密度的互联需求，依据应用场景及定制化程度的差异，本文提供了 3 种互联芯粒设计方案供设计开发者选择.

4.1 可编程互联芯粒

如图 8 所示，可编程互联芯粒通常用于产品定义的早期开发阶段，需要高度灵活性和快速原型实现的场景. 它提供了大量可编程资源，用于实现复杂的逻辑功能，通过硬件描述语言定义互联接口协议、协议适配以及接口间的数据交换模式，支持多路高速端口、通道实现芯粒间互联，每路高速端口都配有 Bank 资源. JTAG 接口用于对芯粒内部节点以及集成芯片进行调试、配置和测试. 可编程互联芯粒具有广泛的适应性和较高的灵活性，它支持芯粒功能设计的快速迭代，可满足长期的设计开发需求. 此外，绝

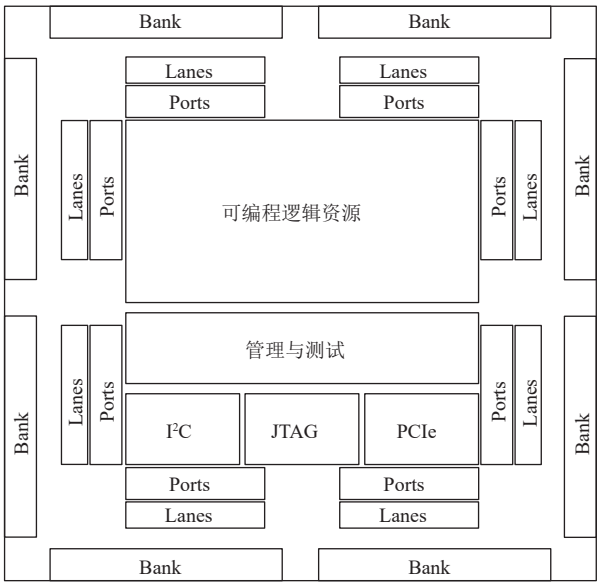


Fig. 8 Programmable interconnection Chiplets

图 8 可编程互联芯粒

大部分的芯粒互联接口协议能够通过编程实现，具有很强的接口适配能力. 然而，相比其他 2 种设计类型，可编程互联芯粒的设计实现具有更高的复杂度，相应的芯粒的面积、功耗和设计风险更大，其设计成本和规模化生产成本也更高.

4.2 路径可编程互联芯粒

如图 9 所示，路径可编程互联芯粒通过可编程模块控制 D2D 互联接口间的数据处理与交换，常用的可编程模块包括以指令集为核心的 MCU 以及以 LUT 门电路为核心的 eFPGA. 此外，该种互联芯粒包含定制化的 D2D 互联接口、外部互联接口、DDR 接口、FLASH 控制器、JTAG 控制器、系统总线、中断控制、时钟管理等模块. 可编程模块用于接收指令并执行对互联管理芯粒、集成芯片的协同管理，实现不同功能芯粒间的数据交换.

该芯粒具有传输、存储、管理等定制化的资源

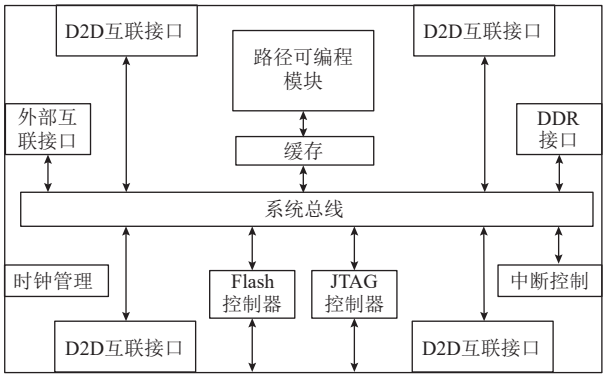


Fig. 9 Path programmable interconnection Chiplets

图 9 路径可编程互联芯粒

以及可用于数据路由配置的可编程资源,通过编程实现对通信链路的逻辑功能配置,为用户提供芯粒互联自定义的空间,以适应多变的工作负载和通信模式.定制化的D2D互联接口用于实现互联芯粒与功能芯粒的数据传输.外部互联接口用于集成芯片与片外功能电路的互联.DDR接口用于可编程模块的指令、程序、数据缓存,以及接收可编程模块的存储访问指令,实现对存储器的连接控制和数据的存储与访问.Flash控制器用于管理Flash存储器的读写操作,包括擦除、编程以及监控Flash的状态.JTAG控制器用于调试和测试可编程模块,以及访问FPGA内部寄存、状态和时钟信息,观察和控制信号的状态,实现集成芯片调试和功能验证.系统总线用于各功能模块的内部互联与数据通信.中断控制用于监控可编程模块的运行状态,防止程序因为某些原因陷入死循环或异常状态.时钟管理包括全局时钟和标准接口协议时钟,全局时钟用于控制互联芯粒内部的数据通信,标准接口协议时钟用于控制互联芯粒与功能芯粒(或外部模块)之间的通信.

路径可编程互联芯粒的灵活性介于可编程互联芯粒和专用互联芯粒之间,具有很好的容错性,可以根据路径的故障情况动态配置使用其他路径,能够适应系统需求变化,快速实现新的通信策略而不需要改变硬件.相比可编程互联芯粒,路径可编程互联芯粒采用了定制化的外围电路,相比专用互联芯粒,它具备了路径可配置的能力,因此兼备了两者的优势,一定程度上降低了接口的数据传输延迟,提高了接口数据交换的灵活性,同时也降低了芯粒的面积和功耗以及芯粒的规模化生产成本.然而,其接口配置的灵活性相比可编程互联芯粒明显降低,设计和维护可编程互联路径比专用互联芯粒更为复杂,其动态的路由决策会增加延迟和降低通信效率.

4.3 专用互联芯粒

如图10所示,专用互联芯粒通过定制化的互联结构和逻辑实现芯粒互联,适用于对性能要求极高和量产规模足够大可以抵消开发成本的场合.专用互联芯粒具有针对特定应用的最佳性能,满足小面积、低功耗、高性能、规模化的成本优势,其设计开发周期最短,开发成本最低,设计风险也最低.然而,专用互联芯粒的灵活性最低,一旦设计制造完成,无法更改其功能以适应新的需求.

典型互联芯粒的对比如表4所示,总的来说,可编程互联芯粒、路径可编程互联芯粒和专用互联芯粒各有其独特的特点和应用场景,其专用化程度依次升高,灵活性逐渐减弱,选择哪一种设计模式取决于项目具体的实施阶段和应用需求,以及对性能、功耗、设计成本、设计周期和对未来灵活性的重视程度.一般而言,在产品的早期开发阶段,可编程设计具有很强的可配置性,可以充分发挥其高度灵活性和快速原型制作的优势,通过编程方式调整芯粒间的连接方式和数据传输路径,从而实现更高的性能优化和资源利用率.这种类型的互联芯粒适用于需要高度灵活性和可扩展性的应用,比如云计算、大数据处理等.与可编程互联芯粒相比,路径可编程设计更侧重于优化数据传输路径的效率和可靠性,具有定制化的高速接口和可编程资源,它提供了灵活的系统设计选择,可以实现更高效的数据传输和处理器之间的通信,适用于对性能有较高要求的应用,比如高性能计算、AI计算等.专用互联设计是为特定应用或系统设计的,具有固定的连接方式和功能,主要应用于特定的硬件平台或系统,比如服务器架构、GPU集群等.其

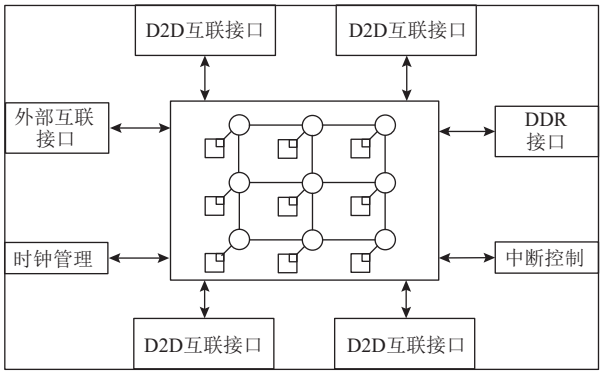


Fig. 10 Application specific interconnection Chiplets

图10 专用互联芯粒

Table 4 Comparison of Typical Interconnected Chiplets

表4 典型互联芯粒对比

类别	可编程互联芯粒	路径可编程互联芯粒	专用互联芯粒
应用场景	云计算、大数据处理等	高性能计算、AI计算等	服务器架构、GPU集群
灵活性	最高	适中	最低
设计复杂度	最高	适中	最低
设计费用	最高	适中	最低
设计周期	最长	适中	最短
设计风险	最高	适中	最低
规模化生产成本	最高	适中	最低
面积大小	最大	较小	最小
性能	较好	良好	优异
延迟	最高	适中	最低
功耗	最高	适中	最低

优势在于其针对特定应用的优化，能够提供更高的性能和可靠性，但是灵活性相对较低，因此其适用于对性能要求极高但又对灵活性要求不高的场景.

5 基于互联芯粒的可测性设计与测试技术

当前具有“碎片化”特点的集成芯片的可测性与测试面临着诸多挑战. 随集成度的大幅度提升，芯片的可访问性下降，测试覆盖率下降，定位缺陷位置变得愈加困难，可测性设计与测试技术对于提升集成芯片的制造良率和可靠性具有十分重要的意义.

互联芯粒的可测性设计与测试技术是指在集成芯片设计过程中，为了确保芯粒在制造、封装过程中以及最终集成芯片的质量和可靠性，同时使集成芯片在生产、集成后以及运行期间都能够进行有效的测试和故障诊断. 互联芯粒的可测性设计既要解决互联芯粒自身测试，也要服务于所构建的集成芯片测试.

互联芯粒的可测性设计需充分考虑集成芯片内各个芯粒的可测试性，在芯粒设计中加入测试和诊断相关的架构设计，采取一系列的测试方法，实现对互联芯粒、功能芯粒、集成芯片等的功能和性能测试.

基于互联芯粒的可测性设计可依据 IEEE 1149.1（JTAG 接口）、IEEE 1838 等标准[35-41]. IEEE 1149.1 标准适用于集成电路测试和故障诊断，能够实现对芯片内部逻辑的访问与控制. 基于 IEEE 1149.1 标准改进的边界扫描链路设计[42] 能够实现更便捷的互联芯粒外测能力，支撑互联芯粒开展对外低速 IO 构建的互连网络测试. IEEE 1838 标准专门针对 2.5D 和 3D 堆叠集成电路提供了测试解决方案，该标准提供了 2.5D 和 3D 堆叠设备的标准化测试端口.

互联芯粒的测试设计通常包括 3 方面：基于 1149.1 的边界扫描链路设计可实现更便捷的互联芯粒外测能力，支撑互联芯粒开展对外低速 IO 构建的互连网络互联测试；基于高速接口内嵌 BERT 设计可实现传输误码与眼图测试，支撑互联芯粒开展高性能传输监测；基于可扩展自适应的测试压缩技术可实现测试编码压缩，支撑互联芯粒全扫描自检测试编码高效率加载与移入移出[43].

互联芯粒为功能芯粒提供测试访问性. 功能芯粒分为支持最大公约芯粒测试集的芯粒与不支持最大公约芯粒测试集的芯粒 2 种. 对于支持最大公约芯粒测试集的芯粒，可直接基于可编程逻辑构建的最大公约测试集测试结构完成芯粒测试，由测试结构提供测试激励输入与测试响应采集. 对于尚未支持最大公约芯粒测试集的芯粒，在集成芯片设计阶段可将其可测性引脚连接至可编程逻辑，实现功能芯粒专用测试集的测试激励输入与测试响应采集. 基于可编程逻辑实现的测试结构，具有测试向量可直接生成或移植、测试控制难度小、测试开发效率高的特点，可全面提升集成芯片的测试覆盖率.

集成芯片的制造良率需要考虑单颗芯粒自身的良率和多芯粒封装过程的良率. 为保证集成芯片的良率，集成芯片除了在封装前保障芯粒良率之外，同样需要在封装后针对封装过程中引入的故障进行良率测试，排除封装过程中产生的缺陷. 在基于互联芯粒的集成芯片中，互联芯粒承担将多颗芯粒集成互联为具备海量数据通信、大规模计算等高性能的功能簇，同时互联芯粒还承担其他具备独立功能的芯粒与集成芯片内其他功能簇的通信、互联与数据交换. 因此，互联芯粒更适合承担集成芯片内互连网络故障排查任务. 基于互联芯粒构建的互连网络可开展高速端口协议初始化自检与超高速互连传输链路的低延迟纠错编码设计、低速端口互连线自检与互连网络故障定位算法、启动自检专用测试控制结构设计等技术研究，实现互连网络覆盖物理连接、通信协议等多个层面的启动自检，同时具备可编程的自检范围设计与直接的自检结果反馈.

6 未来趋势及挑战

当前芯粒集成技术潜在的功能多样性、构建灵活性并未得到深入挖掘，突破以芯粒互联技术为支撑的自下而上的设计方法十分重要. 由于不同芯粒的异构特性，芯粒互联技术是多功能集成的关键. 芯粒技术的发展依赖于生态建设，芯粒接口的标准化是形成芯粒技术生态的必然要求，标准化的形成取决于芯粒互联协议的统一. 当前国内短时间涌现了大量的芯粒互联协议，这一定程度上对芯粒技术的发展起到了促进作用，但是过多的协议会阻碍技术生态的形成，不利于芯粒技术的长远发展. 因此，制定一套国内统一且完备的芯粒互联标准势在必行，既要充分吸收国内主流协议的优点，又要兼容国外主流协议标准（UCIe 协议），实现国内统一且与国际接轨的目标. 芯粒互联架构可能会逐渐呈现以中心 IO 芯粒架构为主、多种互联架构并存的局面，选择哪种架构取决于设计开发者对性能、成本、设计复杂度以及扩展性的综合考量. 容错机制对于提高集

成芯片的可靠性和稳定性至关重要,未来更多的设计将通过引入冗余编码、容错拓扑和容错路由,实施动态可重构策略,确保系统在出现硬件故障时能够继续保持正常运行.互联芯粒的设计模式取决于项目具体的实施阶段和应用需求,未来一段时间内,路径可编程互联芯粒以其较高的灵活性可能成为系统设计的重要选择,专用互联芯粒以其高性能和高可靠性也可能为集成系统设计提供重要解决方案.未来,基于互联芯粒的可测性设计与测试将聚焦提高覆盖率和准确性以实现自动化测试.

芯粒互联技术的发展当前仍处于初级阶段,芯粒互连密度的提升将引发一系列科学问题.随着芯粒集成度的大幅提升,需要设计新型高并行效率的体系架构,设计满足主流通用互联协议的接口电路,充分释放芯粒集成的算力潜能.大规模芯粒集成也带来了集成芯片可访问性下降和测试覆盖率下降的问题,使得定位缺陷位置变得愈加困难.当前芯粒互联也存在一些待解决的关键技术难题,比如芯粒的标准化接口的设计与生态构建、高速互联技术、高速互联接口的容错技术等,因此需要建立统一的芯粒互联标准,突破芯粒互联高速接口设计技术,实现芯粒设计、制造、封装、测试全产业链生态构建.

7 结束语

“后摩尔时代”先进制程的物理尺寸微缩发展路线受限,芯粒技术为应对面积墙、互联墙、存储墙和成本墙问题提供了解决方案,其核心特征是具有极高的互连密度和带宽.芯粒技术的核心是芯粒间高速互联,当前国内外芯粒互联标准尚未形成统一的标准,各类协议各具特色与优势,亟需形成国内统一、国际接轨的互联标准.芯粒互联架构对于组建集成芯片具有重要的指导意义,目前形成了多芯粒架构、中心 IO 芯粒架构和侧接口芯粒架构 3 种模式.芯粒容错机制为确保数据可靠传输和应对电路故障提供了解决方案.互联芯粒能够支持快速组建集成芯片产品,根据应用场景及定制化程度的差异可以选择可编程、路径可编程、专用互联芯粒设计方案.基于互联芯粒的可测性设计与测试技术为 2.5D、3D 集成芯片提供了一种测试解决方案.

总之,芯粒技术是行业技术、市场需求与经济效益协同选择的结果,在先进制程受限背景下,芯粒技术对于我国集成电路产业而言是一个关键机遇,也是一个重要挑战.生态建设是芯粒技术发展的基础,互联技术是芯粒技术实现的关键.本文从协议、架构、容错、互联、测试多个方面介绍了当前芯粒互联的技术细节与发展现状,希望帮助相关研究人员对芯粒互联技术加深了解.

作者贡献声明: 王浩确定综述选题,提出整体研究思路,撰写与修改论文;王勇和冯长磊对论文选题、组织结构和研究内容提供指导意见;盖伟新指导论文写作;吴鹏和钱江负责收集芯粒互联架构的相关资料,提出论文修改意见.

参 考 文 献

[1] Naffziger S, Beck N, Burd T, et al. Pioneering Chiplet technology and design for the AMD EPYC™ and Ryzen™ processor families: Industrial product[C]//Proc of the 48th ACM/IEEE Annual Int Symp on Computer Architecture (ISCA). Piscataway, NJ: IEEE, 2021: 57−70

[2] Loh G H, Naffziger S, Lepak K. Understanding Chiplets today to anticipate future integration opportunities and limits[C]//Proc of the 24th Design, Automation & Test in Europe Conf & Exhibition (DATE). Piscataway, NJ: IEEE, 2021: 142−145

[3] Feng Yinxiao, Ma Kaisheng. Chiplet actuary: A quantitative cost model and multi-Chiplet architecture exploration[C]//Proc of the 59th ACM/IEEE Design Automation Conf. Piscataway, NJ: IEEE, 2022: 121−126

[4] Suggs D, Subramony M, Bouvier D. The AMD ‘zen 2’ processor[J]. IEEE Micro, 2020, 40(2): 45−52

[5] Sharma D D, Pasdast G, Qian Zhiguo, et al. Universal Chiplet Interconnect Express (UCIe): An open industry standard for innovations with Chiplets at package level[J]. IEEE Transactions on Components, Packaging and Manufacturing Technology, 2022, 12(9): 1423−1431

[6] Liao Jie, Jiao Bo, Zhang Jinshan, et al. A scalable die-to-die interconnect with replay and repair schemes for 2.5D−3D integration[C]//Proc of the 56th IEEE Int Symp on Circuits and Systems (ISCAS). Piscataway, NJ: IEEE, 2023: 57−70

[7] Meng Hui, Zhao Qian, Yoshida T. A study of reconfigurable switch architecture for Chiplets interconnection[C]//Proc of the 10th Int Symp on Computing and Networking Workshops (CANDARW). Piscataway, NJ: IEEE, 2022: 69−75

[8] Chen Guilin, Wang Guanwu, Hu Jian, et al. Survey on Chiplet packaging structure and communication structure[J]. Journal of Computer Research and Development, 2022, 59(1): 22−30 (in Chinese)
(陈桂林, 王观武, 胡健, 等. Chiplet 封装结构与通信结构综述[J]. 计算机研究与发展, 2022, 59(1): 22−30)

[9] Jiang Jianfei, Wang Qin, He Guanghui, et al. Research and prospect

on Chiplet technology[J]. Microelectronics & Computer, 2022, 39(1): 1−6 (in Chinese)
(蒋剑飞, 王琴, 贺光辉, 等. Chiplet 技术研究与展望[J]. 微电子学与计算机, 2022, 39(1): 1−6)

[10] Li Yingxuan. The state-of-the-art of Chiplet and problems need be solved[J]. Microelectronics & Computer, 2022, 39(5): 1−9 (in Chinese)
(李应选. Chiplet 的现状和需要解决的问题[J]. 微电子学与计算机, 2022, 39(5): 1−9)

[11] Liu Yafei, Li Xiangyu, Yin Shouyi. Review of Chiplet-based design: System architecture and interconnection[J]. Science China Information Sciences, 2024, 67(10): 200401: 1−200401: 20

[12] Xie Weikun, Cai Zhikuang, Liu Xiaoting, et al. Overview of Chiplet testing technology[J]. Electronics &Packaging, 2023, 23(11): 110101: 1−110101: 11 (in Chinese)
(解维坤, 蔡志匡, 刘小婷, 等. 芯粒测试技术综述[J]. 电子与封装, 2023, 23(11): 110101: 1−110101: 11)

[13] China Computer Federation. White Paper on Integrated Chip and Chiplet Technology [EB/OL]. 2023 [2024-06-09]. https://www.gitlink.org.cn/zone/iChips/Source/12 (in Chinese)
(中国计算机学会. 集成芯片与芯粒技术白皮书)[EB/OL]. 2023 [2024-06-09]. https://www.gitlink.org.cn/zone/iChips/Source/12)

[14] Universal Chiplet Interconnect Express. Universal Chiplet Interconnect Express (UCIe) Specification Revision 1.1 [EB/OL]. (2023-07-10) [2024-06-10]. https://www.uciexpress.org/ucie-1-1-white-paper-download

[15] China Electronics Standardization Association. T/CESA 1248—2023 Techical Requirement for Chiplet Interface Bus[S]. Beijing: China Electronics Standardization Association, 2023 (in Chinese)
(中国电子工业标准化技术协会. T/CESA 1248—2023 小芯片接口总线技术要求[S]. 北京: 中国电子工业标准化技术协会, 2023)

[16] Chiplet Industry Alliance of China. T/ZGTXXH096−2024 Advanced Cost-driven Chiplet Interface [EB/OL]. 2022 [2024-06-12]. http://www.iiisct.com/smart/upload/CMS1/202303/ACC1.0.pdf(in Chinese)
(中国 Chiplet 产业联盟. T/ZGTXXH096−2024 芯粒互联接口标准[EB/OL]. 2022 [2024-06-12]. http://www.iiisct.com/smart/upload/CMS1/202303/ACC1.0.pdf)

[17] Gomes W, Khushu S, Ingerly D B, et al. 8.1 lakefield and mobility compute: A 3D stacked 10 nm and 22 FFL hybrid processor system in 12×12mm^2, 1mm package-on-package[C]//Proc of the 67th ISSCC. Piscataway, NJ: IEEE, 2020: 144−146

[18] Beck N, White S, Paraschou M, et al. “Zeppelin”: An SoC for multichip architectures[C]// Proc of the 65th ISSCC. Piscataway, NJ: IEEE, 2018: 40−41

[19] Naffziger S, Lepak K, Paraschou M, et al. 2.2 AMD Chiplet architecture for high-performance server and desktop products [C]//Proc of the 67th ISSCC. Piscataway, NJ: IEEE, 2020: 44−45

[20] Liao Heng, Tu Jiajin, Xia Jing, et al. Ascend: A scalable and unified architecture for ubiquitous deep neural network computing[C]//Proc of the 27th IEEE Int Symp on High-Performance Computer Architecture (HPCA). Piscataway, NJ: IEEE, 2021: 789−801

[21] Bossert M, Schulz R, Bitzer S. On hard and soft decision decoding of BCH codes[J]. IEEE Transactions on Information Theory, 2022, 68(11): 7107−7124

[22] Valls J, Torres V, Canet M J, et al. A test vector generation method based on symbol error probabilities for low-complexity Chase soft-decision Reed-Solomon decoding[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2019, 66(6): 2198−2207

[23] Zhang Xinmiao. VLSI architectures for Reed-Solomon codes: Classic, nested, coupled, and beyond[J]. IEEE Open Journal of Circuits and Systems, 2020, 1: 157−169

[24] Gao Jiajing, Zhang Wei, Liu Yanyan, et al. High-performance concatenation decoding of Reed-Solomon codes with SPC codes[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2021, 29(9): 1670−1674

[25] Guo Jianzhong, Li Ruoxi, Zhang Yuan, et al. Hard-decision decoding algorithms for LDPC codes based on matching pursuit[J]. IEEE Communications Letters, 2024, 28(3): 454−457

[26] Arikan E. Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels[J]. IEEE Transactions on Information Theory, 2009, 55(7): 3051−3073

[27] Wang Ying, Han Yinhe, Zhang lei, et al. Economizing TSV resources in 3-D network-on-chip design[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014, 23(3): 493−506

[28] Fu Binzhang, Han Yinhe, Li Huawei, et al. Zonedefense: A fault-tolerant routing for 2-D meshes without virtual channels[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2013, 22(1): 113−126

[29] Xiang Dong, Chakrabarty K, Fujiwara H. Multicast-based testing and thermal-aware test scheduling for 3D ICs with a stacked network-on-chip[J]. IEEE Transactions on Computers, 2015, 65(9): 2767−2779

[30] Ni Tianming, Xu Qi, Huang Zhengfeng, et al. A cost-effective TSV repair architecture for clustered faults in 3-D IC[J]. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2020, 40(9): 1952−1956

[31] Tosun S, Ajabshir V B, Mercanoglu O, et al. Fault-tolerant topology generation method for application-specific network-on-chips[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 34(9): 1495−1508

[32] Ren Pengju, Ren Xiaowei, Sane S, et al. A deadlock-free and connectivity-guaranteed methodology for achieving fault-tolerance in on-chip networks[J]. IEEE Transactions on Computers, 2016, 65(2): 353−366

[33] Chen Y, Chang E, Hsin H, et al. Path-diversity-aware fault-tolerant routing algorithm for network-on-chip systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2017, 28(3): 838−849

[34] Wang Hao, Wang Yong, Feng Changlei, et al. Aerospace high reliability and high-speed interconnection Chiplets. China, CN202311597700.9 [P/OL] . 2024-03-29 [2024-06-15]. https://d.

wanfangdata.com.cn/patent/ChhQYXRlbnROZXdTMjAyNDExMjIxNjU4MjISEENOMjAyMzExNTk3NzAwLjkaCDduM3R0YjM5(in Chinese)
(王浩, 王勇, 冯长磊, 等. 宇航高可靠高速互联芯粒: 中国, CN202311597700.9[P/OL]. 2024-03-29 [2024-06-15]. https://d.wanfangdata.com.cn/patent/ChhQYXRlbnROZXdTMjAyNDExMjIxNjU4MjISEENOMjAyMzExNTk3NzAwLjkaCDduM3R0YjM5)
[35] IEEE Computer Society. IEEE 1149.1-2013 Test Technology Standards Committee. Test Access Port and Boundary-Scan Architecture[S]. Piscataway, NJ: IEEE, 2013
[36] IEEE Computer Society. IEEE 1500-2005 Testability Method for Embedded Core-based Integrated Circuits[S]. Piscataway, NJ: IEEE, 2005
[37] IEEE Standards Association. IEEE 1687-2014 Access and Control of Instrumentation Embedded within A Semiconductor Device[S]. Piscataway, NJ: IEEE, 2014
[38] IEEE Computer Society. IEEE 1838-2019 Test Access Architecture for Three-Dimensional Stacked Integrated Circuits[S]. Piscataway, NJ: IEEE, 2020
[39] Abdennadher S. Testing inter-Chiplet communication interconnects in a disaggregated SoC design[C/OL]//Proc of the IEEE Int Conf on Design & Test of Integrated Micro & Nano-Systems (DTS). Piscataway, NJ: IEEE, 2021 [2024-06-13]. https://xplorestaging.ieee.org/document/9498132
[40] Chandra A, Khan M, Patidar A, et al. A case study on IEEE 1838 compliant multi-die 3DIC DFT implementation[C]//Proc of the IEEE Int Test Conf (ITC). Piscataway, NJ: IEEE, 2023: 11-20
[41] McLaurin T, Cron A. Applying IEEE test sandards to multidie designs[J]. IEEE Design & Test, 2022, 39(5): 7-16
[42] Zhang Zhen, Wei Jinghe, Yin Yujia, et al. An improved test structure of boundary scan designed for 2.5D integration[C]//Proc of the 8th Int Conf on Integrated Circuits and Microsystems (ICICM). Piscataway, NJ: IEEE, 2023: 643-648
[43] Syed U S, Chakrabarty K, Chanadra A, et al. 3D-scalable adaptive scan (3D-SAS)[C]//Proc of the IEEE Int 3D Systems Integration Conf (3DIC) . Piscataway, NJ: IEEE, 2012: 1-6



Wang Hao, born in 1993. PhD, engineer. His main research interests include Chiplet interconnection, digital signal processing, and VLSI implementation.
王 浩, 1993 年生. 博士, 工程师. 主要研究方向为芯粒互联、数字信号处理、VLSI 实现.



Wang Yong, born in 1977. PhD, professor. His main research interests include Chiplet and integrated chip design and testing, manufacturing process, packaging, testing, detection, and failure analysis of integrated circuits and semiconductor discrete devices.
王 勇, 1977 年生. 博士, 研究员. 主要研究方向为芯粒与集成芯片设计与测试, 集成电路和半导体分立器件的制造工艺、封装、测试、检测与失效分析.



Feng Changlei, born in 1984. PhD candidate, professor. His main research interests include Chiplet interconnection and integration, and microsystem design and testing.
冯长磊, 1984 年生. 博士研究生, 研究员. 主要研究方向为芯粒互联与集成、微系统设计与测试.



Gai Weixin, born in 1969. PhD, professor. His main research interests include ultra-high-speed wireline transceiver, Chiplet interface circuits, optical receiver, phase-locked loop, and high-speed analog-to-digital converter.
盖伟新, 1969 年生. 博士, 教授. 主要研究方向为超高速有线收发机、芯粒接口电路、光接收机、锁相环、高速模数转换器.



Wu Peng, born in 1988. Master, senior engineer. His main research interests include heterogeneous integration design, and signal integrity and power integrity research.
吴 鹏, 1988 年生. 硕士, 高级工程师. 主要研究方向为异质异构集成设计、信号完整性与电源完整性研究.



Qian Jiang, born in 1994. PhD, engineer. His main research interests include 2.5D integration design and interconnection structure design.
钱 江, 1994 年生. 博士, 工程师. 主要研究方向为 2.5D 集成设计、互连结构设计.