Computer Fraud and Security
ISSN (online): 1873-7056

Advanced Verification Methodologies for Multi-Die Integrated Systems

Suri Babu Talla

Independent Researcher, USA

Abstract

The semiconductor industry is experiencing a fundamental transformation as traditional monolithic system-on-chip integration encounters insurmountable economic and physical limitations at advanced process nodes. Escalating mask costs, yield degradation on large dies, and the imperative for heterogeneous computing architectures have catalyzed the widespread adoption of chiplet-based design methodologies. By decomposing complex systems into smaller, modular dies interconnected through advanced packaging technologies, chiplets enable process-node specialization, improved manufacturability, architectural flexibility, and cost-effective scaling across diverse product segments. However, this architectural paradigm introduces unprecedented verification challenges that transcend traditional validation approaches. Multi-die systems must ensure correctness across high-bandwidth die-to-die interconnects supporting multiple concurrent protocol layers, distributed cache coherency mechanisms operating across chiplet boundaries, asynchronous clock domain crossings with variable latencies, and complex power-thermal interactions in three-dimensional stacked configurations. These system-level interdependencies manifest only under sustained execution of realistic software workloads, creating validation requirements that conventional simulation methodologies cannot feasibly address due to prohibitive cycle count demands and limited observability across distributed architectures. Hardware emulation has emerged as the indispensable cornerstone of chiplet verification, uniquely providing the execution speed, multi-billion gate capacity, full-system visibility, and protocol monitoring capabilities necessary to validate complex multi-die interactions. This article presents a comprehensive examination of chiplet architectures, the multifaceted verification complexities they introduce across physical, protocol, and temporal domains, and the advanced emulation methodologies that enable successful validation of next-generation heterogeneous integrated systems.

Keywords: Chiplet Architectures, Multi-Die Verification, Hardware Emulation, Die-To-Die Interconnects, Distributed Coherency Protocols.

Introduction

The costs of monolithic integration, yield limitations on large dies, and the increasing use of heterogeneous computing architectures have brought the semiconductor industry to a practical limit, and modular multi-die systems are emerging as a solution. At nodes below 7 nm, cost constraints have already grown massively, while the reticle size has become a limiting factor in monolithic die size (monolithic die sizes have typically been limited to 800 mm² (1.2 sq in)), which leads to chiplet-based architecture. According to members from ResearchGate who have examined chiplet-based architecture, the economic feasibility of mass-scale implementation of chiplet-based architecture is becoming doubtful because mask set cost can easily occupy an important fraction of the development budget of new circuitry at advanced nodes, especially for small design teams or niche circuits. By obstructing economic yield degradation in a manner that asymmetrically affects larger dies, this handicap has considerably impacted industry directions in semiconductor architecture design.

In this approach, heterogeneous functions are each divided into chiplets, which are in turn interconnected using 2.5D silicon interposers, organic interposer substrates with embedded bridges, or 3D implementations. The most appropriate process node for each function can be selected, based on its cost and suitability for that function. In practice, chiplets have been used in semiconductor packages to create automotive compute processors for autonomous driving. In this application, advanced semiconductor nodes are used for compute processors, while memory, analog interfaces, and legacy IP blocks are fabricated with mature nodes [2]. Chiplet architectures make manufacturability easier, with small die sizes within the optimum yield range, and allow for architecture customization or scaling to different products and market segments.

This modularity adds verification challenges that cannot easily be handled with customary approaches, as chiplet systems span multiple domains. In addition, some multi-die systems require verification of die-to-die communication fabrics, like multi-protocol high-bandwidth mesh, that include multiple protocol layers like cache coherency for memory consistency, peripheral interconnects for input-output (I/O), or streaming I/O from accelerators to general-purpose cores. Chiplet architecture studies have also stressed cross-die clock domain crossing verification challenges due to independent frequency sources, interconnect latencies that vary with package parasitics, and thermal profiles that vary with the implementation (cross-die thermal gradients) [1]. Distributed cache coherency protocols that cross chiplets can create race conditions, deadlock, and ordering violations due to particular traffic patterns and long instruction sequences from mainstream workloads.

Validation of multi-die systems includes correctness of inter-die communications and distributed protocols, as well as the correct behavior of the complete system. This is at a scale and a level of detail that cannot be adequately validated with a simulation flow due to the intrinsically sequential nature and low throughput of all current simulation flows. For example, simulation flows cannot boot an OS, cannot verify the firmware initialization sequence across chiplets, and cannot stress an interconnect protocol with realistic traffic to expose corner-case protocol violations and timing-induced failures [2].

Multi-Die System Architecture and Design Partitioning

Multi-die architectures avoid core scaling limitations by partitioning functionality onto separate dies, allowing each to be fabricated in the best process technology, improving die yield through reduced die area, and reusing IP across multiple products. They are successful when the partitioning takes advantage of architectural cleanliness and simplicity while balancing inter-die communication overhead.

Functional partitioning is a model that separates compute, memory, input-output, and acceleration into different dies. While this establishes clear boundaries, it can cause bandwidth bottlenecks. This allows for each domain to be optimized as best as possible, i.e., chiplets with compute can be manufactured with more advanced nodes for better performance and power efficiency, while memory dies could be manufactured on older nodes with better analog characteristics and manufacturing maturity. With chiplet-based architectures, functional partitioning allows heterogeneous integration, because different intellectual property blocks could be produced with the most suitable process technology in terms of performance and cost to the system [3]. Partitioned architectures still face the issue of communication between dies, but system architecture can be optimized to handle this case, particularly for time-critical communications, because physical separation between dies increases communication latency beyond that of an on-die interconnect.

Die-level partitioning, mapping each of the logical blocks to a physical die, is more difficult to package and requires high bandwidth when the workload is memory- or computation-bound. In such a co-package, compute dies are populated with multiple high-performance cores and caches, and memory chiplets are connected with high-bandwidth interfaces using package-level interconnects such as Intel Foveros and AMD Infinity Fabric. The yield impact of chiplet technology has been studied and found to be positive. A chiplet design is expected to have a higher yield than a monolithic die of the equivalent function because smaller dies are statistically less likely to have defects. In contrast, yield degradation is more pronounced for small feature size manufacturing, where defect density limits semiconductor manufacturing capability [4]. Partitioning at the die level allows flexibility in how systems are constructed from chiplets in terms of both type and quantity of chiplets per package for a market segment/application.

These solutions, known as interconnect-centric solutions, organize each system around their communications fabrics. Protocol verification is efficient in these systems, but system-level verification is more difficult. Architectures in this category have a dedicated communications fabric, or complex communications structure, that also provides centralized protocol transcoding, coherency transactions, and quality-of-service arbitration. Studies on future chiplet architectures mention that interconnect architecture must support a wide range of traffic classes with different latency and bandwidth requirements, such as cache coherency transactions requiring low latency, peripheral communications with medium latency tolerance, and high sustained throughput for streaming transfers for accelerator workloads [3]. Each traffic class has its own latency, power, thermal, and verification trade-offs that vary depending on the requirements of the design due to the dependencies of the distributed nature of multi-die systems and differing operating conditions and workload scenarios [4].

Partitioning Strategy	Functional Approach	Architectural Characteristics	Primary Benefits	Key Challenges	Process Node Flexibility
Functional Partitioning	Separates compute, memory, I/O, and acceleration into distinct dies	Clear architectural boundaries; Domain-specific optimization	Independent functional optimization, Process node specialization; cost-effectiveness	Potential bandwidth bottlenecks; Cross-boundary communication overhead; Longer signal propagation paths	Very High - Each function uses optimal node
Die-Level Partitioning	Maps each logical block directly to physical component	Modular component-based architecture; Direct logical-to-physical mapping	Enhanced manufacturing yield; Flexible system configurations; Scalable computational resources	Requires high-speed interconnects; Substantial aggregate bandwidth demands	High - Component-specific optimization
Interconnect-Centric Partitioning	Organizes around communication fabrics as central hubs	Dedicated interconnect structures; Centralized protocol management	Optimized protocol validation; Efficient QoS arbitration; Multi-traffic support	Complex system-level verification; Distributed performance dependencies; Fabric bottlenecks	Medium - Constrained by fabric requirements

Table 1: Multi-Die System Partitioning Strategies - Characteristics and Design Trade-offs [3, 4]

Interconnect Technologies and Protocol Considerations

The communication links between die forming the backbone of multi-die systems are the key enablers of heterogeneous integration and modular semiconductor system architecture. The latest interconnect standards provide a complete physical layer and protocol specification to support all types of traffic, e.g. memory coherency traffic, peripheral communications, and streaming data between chiplets. Research in high-performance power-efficient 3D system-in-package (SiP) solutions has shown UCIe to be the die-to-die market standard, with a complete protocol stack architecture definition. This defines physical layer signaling, die-to-die adapter functions, and protocol layer definitions for meaningful interoperating Assured Interconnects between heterogeneous chiplet vendors and process nodes [5]. UCIe's raw physical layer bandwidth per channel is 12 to 32 GT/s per lane for standard and broadband packaging implementations [5]. The UCIe specification defines packaging densities of 2 to 4 Tb/s/mm die edge for higher levels of integration and data movement between and among compute, memory, and accelerator chiplets [5]. Quantitative and qualitative analysis of system-in-package designs has led to the conclusion that a vendor-neutral interconnect technology specification effort focused on heterogeneous computing platforms with artificial intelligence accelerators, high-performance processors, and specialized memory controllers will meet a range of bandwidth, latency, and power requirements in a packaged assembly.

Short-reach interfaces often place emphasis on low power consumption and low latency. Many employ wide interface buses, eliminating the need for expensive serialization and deserialization circuits. Most use source-synchronous clocking, where the data is accompanied by a forwarded clock signal, allowing the receiving circuit to be simple, without requiring additional clock recovery and phase-locked loop circuitry. Reading the power consumption characteristics of chiplet and interconnect

technology data, short-reach parallel interfaces are extremely low power, and the latest implementations of short-reach parallel I/O have performance-per-watt results that allow continuous high-bandwidth transport within thermal design power (TDP) envelopes. These characteristics support three-dimensional stacking topologies [6]. The physical realization of these I/Os is possible through microbump interconnection technology with 40-55 micrometer pitch bump arrays, enabling thousands of I/Os to be routed in parallel with a small die footprint. Bandwidth scaling is achieved through bump pitch widening rather than implementing progressively higher frequencies, since high frequencies have to be routed over large physical distances, making it difficult for signal integrity and power consumption.

Unlike short-reach, long-reach implementations must also include complex equalization and training algorithms that must compensate for the high loss of packaging substrates, solder bumps, and PC board traces used to carry the signals. In a survey of chiplet technology and SoC roadmaps, serializer-deserializer-based implementations must include multi-tap transmitter pre-emphasis circuits and adaptive receiver equalization engines that compensate for frequency-dependent channel loss [6]. The adaptive circuits run wide-ranging training sequences at link initialization, sweeping through coefficient spaces to determine the optimum transmitter and receiver settings that maximize eye opening and minimize bit error rates given all the actual conditions of the operating channel, such as crosstalk from adjacent signal lanes and power supply noise coupling.

The interconnects must support an arbitrary traffic pattern while still complying with the requirements for protocol compliance, flow control correctness, and error recovery for multiple flows with different priority and quality of service levels. CCI uses a credit-based flow control scheme, in which the sending agent tracks the amount of buffer space available at receiving agents, preventing buffer overflow while maximizing the utilization of the channel [5]. Error detection and recovery, such as cyclic redundancy check calculations for verifying packets and retry buffers that automatically resend dropped frames, can also complicate verification of the hardware layers. Further communication layers in the stack (initialization sequences, timeouts, sequence numbers, and bandwidth arbitration schemes to share limited bandwidth among virtual channels) can also be difficult to verify and require system-level validation with real workloads.

Interconnect Standard	Packaging Type	Raw Channel Bandwidth per Lane	Physical Implementation	Signaling Type
Universal Chiplet Interconnect Express (UCIe)	Standard Packaging	12 to 32 GT/s per lane	Industry standard protocol stack	Configurable PHY
UCIe	Advanced Packaging	12 to 32 GT/s per lane	Dense die-edge connections	High-density parallel
Short-Reach Parallel Interfaces	2.5D/3D Advanced Packaging	Aggregate through width scaling	Microbump: 40 to 55 micrometer pitch	Source-synchronous parallel
Long-Reach SerDes Interfaces	Standard/Advanced Packaging	Multi-gigahertz per differential pair	Standard solder bumps and PCB traces	Serialized with equalization

Table 2: UCIe and Die-to-Die Interconnect Bandwidth Specifications [5, 6]

Verification Challenges in Distributed Systems

Multi-die architectures introduce verification complexities absent in monolithic designs, fundamentally transforming the validation landscape and demanding novel methodologies to ensure functional correctness across distributed computing fabrics. Clock domain crossing verification becomes significantly more challenging due to independent phase-locked loops operating on each chiplet with inherent frequency variations, variable inter-die latency introduced by package routing topologies and transmission line effects, and packaging-induced timing uncertainties arising from manufacturing process variations, thermal gradients, and dynamic voltage fluctuations. Research on quantitative analysis of state-of-the-art

synchronizers demonstrates that these distributed clocking challenges are exacerbated by the need to maintain synchronization across multiple independent timing domains, where each chiplet may operate at different frequencies optimized for its specific workload characteristics, requiring complex clock domain crossing structures with metastability resolution circuits and synchronizer chains that must account for worst-case timing scenarios across all possible phase relationships.

Studies examining clock domain crossing implementations in 90 nm CMOS technology reveal critical performance metrics that illustrate the verification complexity. Level synchronizers operating at a frequency of 1.5 GHz demonstrate latency of approximately 711 picoseconds, while edge-detecting synchronizers exhibit latency around 738 picoseconds, and pulse-generating synchronizers show latency of 750 picoseconds [7]. These timing measurements underscore the asynchronous nature of die-to-die communication, which introduces non-deterministic latencies that can vary by multiple clock cycles depending on instantaneous queue occupancy, arbitration decisions, and flow control state at the moment transactions enter the interconnect fabric. Furthermore, power consumption analysis indicates that level synchronizers consume approximately 1.09 milliwatts, edge-detecting synchronizers require 1.12 milliwatts, and pulse-generating configurations also consume 1.12 milliwatts during operation [7]. These factors create non-deterministic behavior that simulation struggles to capture comprehensively, as the state space of possible timing interactions expands exponentially with the number of clock domains and the depth of synchronizer pipelines distributed throughout the system.

When handshaking protocols are implemented to manage clock domain crossing, the verification complexity increases substantially. Full handshake protocols demonstrate a latency of 4012 picoseconds, while partial handshake implementations exhibit latencies of 3380 picoseconds for the first variant and 3340 picoseconds for the second variant [7]. The corresponding power consumption for these protocols ranges from 2.85 milliwatts for a full handshake to 2.59 milliwatts and 2.72 milliwatts for the two partial handshake variants [7]. These quantitative metrics illustrate that traditional block-level verification cannot expose these system-level interactions because the emergent failure modes depend on precise temporal relationships between events occurring on different dies, with timing variations and power characteristics that only manifest under sustained operational conditions spanning extended execution periods far beyond typical simulation windows.

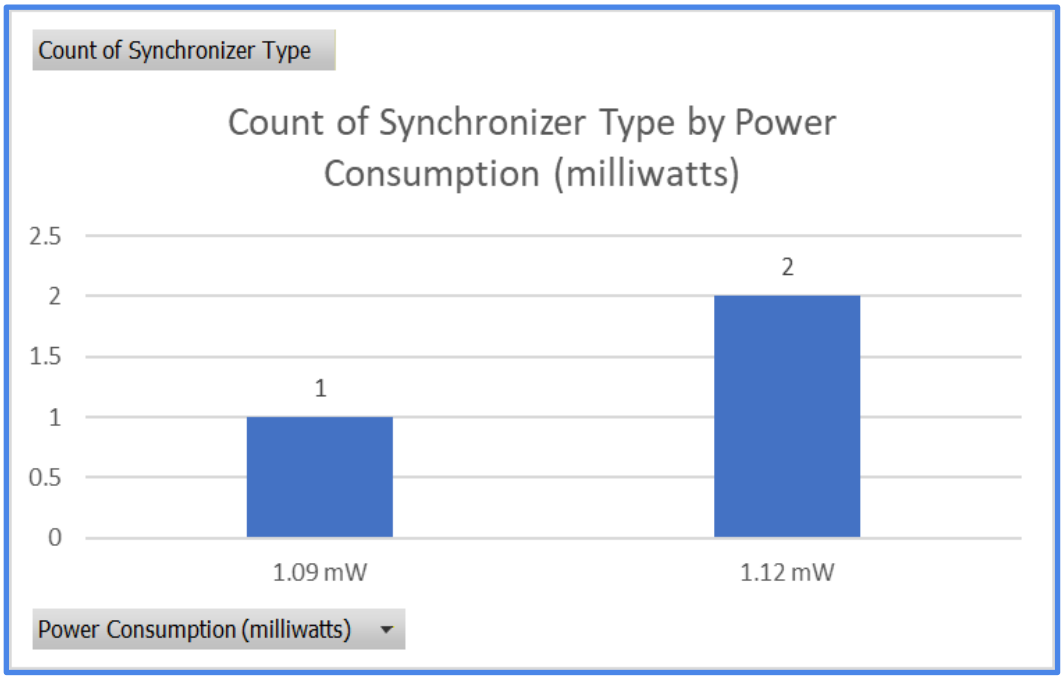

Fig. 1: Clock Domain Crossing Synchronizer Performance Metrics in 90nm CMOS Technology [7, 8]

Emulation as the Verification Foundation

Hardware emulation addresses the verification gap left by simulation and prototyping approaches, emerging as the indispensable methodology for validating complex multi-die systems that demand full-system execution and comprehensive observability across distributed architectures. Multi-die systems may consist of various configurations, including 2.5D packages containing interposer-mounted chiplets; 3D stacks with regular structures such as memory and field-programmable gate arrays; heterogeneous stacks mounted on interposers or bridges; and recursive composition formulations, which are essentially stacks of stacks where the system is partitioned to balance throughput and energy [10].

Simulation cannot execute the billions of cycles required to boot operating systems, stress interconnect protocols under sustained traffic loads, or uncover race conditions in distributed coherency schemes that only manifest after extended execution periods involving realistic software workloads. Each of these multi-die configurations represent a combination of independently manufactured dies interconnected through communication fabrics, which enable designs with massive size. Given the increased scale and complexity associated with multi-die architectures, the incremental refinement design flow typical for monolithic system-on-chips will not work effectively. After the architecture design of monolithic systems, teams typically write the register-transfer level code and tests, find issues, and possibly change the architecture, going back and forth between these steps as needed through synthesis, timing analysis, and power estimation iterations. Such a flow is not possible with multi-die system architectures because the dies are already manufactured and all components must be verified from a system-level perspective [10].

Prototyping platforms struggle with capacity limitations that restrict implementable design sizes, timing closure challenges for complex multi-gigahertz designs with critical path constraints spanning multiple dies and package interconnects, and accurate modeling of analog components including phase-locked loops, power management integrated circuits, and high-speed serializer-deserializer transceivers that are fundamental to die-to-die communication but cannot be directly synthesized into field-programmable gate array logic fabric. System verification for multi-die architectures must validate assumptions made during the architecture design phase, considering parameters including die-to-die communication, delay, jitter, coherency, power, guaranteed delivery, and errors, whereas delay is the only consideration for monolithic systems [10].

Emulation platforms provide the capacity, execution speed, and observability necessary for multi-die validation, offering a unique combination of near-real-time execution performance with comprehensive internal state visibility that enables verification engineers to debug complex system-level interactions. Design size and complexity exacerbate verification challenges in multi-die systems. Scalable simulation and emulation models along with a system integration methodology can provide the capacity and performance required for comprehensive validation [10]. Modern emulation architectures deliver execution speeds that are orders of magnitude faster than software simulation while maintaining the ability to capture detailed signal traces and assert protocol compliance across all chiplet interfaces simultaneously.

The full-system visibility provided by emulation frameworks allows verification teams to observe internal signal states across distributed chiplets while the system executes at speeds approaching real-time operation, enabling the detection of race conditions, protocol violations, and timing-dependent failures that would be extremely difficult to reproduce in controlled simulation environments. Knowing when verification is complete can be tough to determine for multi-die systems, as die-level bugs cannot be fixed at the system level, necessitating exhaustive verification of individual dies with comprehensive functional coverage. This allows system-level verification to focus on scenarios using an explicit coverage model that ensures data arrives at the right place and with the expected throughput and latency [10]. Advanced emulation methodologies incorporate models, transactors including virtual testers, and speed adapters that can be used with emulators to accelerate system validation. The ability to partition designs across multiple emulation engines while maintaining cycle-accurate determinism enables validation of systems exceeding single-platform capacity limits, supporting verification of large-scale multi-chiplet architectures that would be impossible to validate through simulation alone.

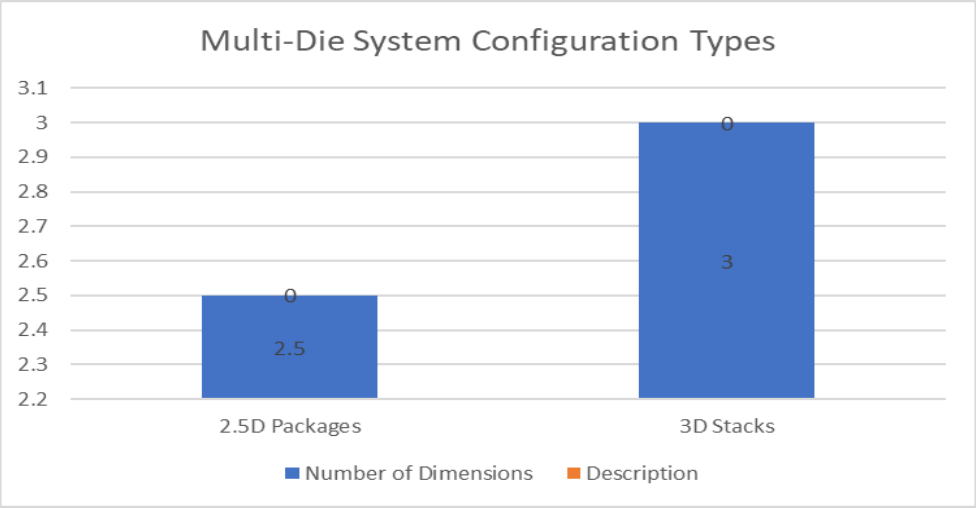

Fig. 2: Multi-Die System Configuration Dimensional Classifications [9, 10]

Conclusion

Chiplet-based architectures represent a paradigmatic shift in semiconductor design methodology, offering compelling solutions to the economic and technical challenges that have rendered traditional monolithic integration increasingly untenable at advanced process nodes. Through modular decomposition of complex systems into specialized dies interconnected via standardized high-bandwidth interfaces, chiplets unlock unprecedented flexibility in process selection, enable yield optimization through smaller die geometries, facilitate intellectual property reuse across product generations, and support scalable heterogeneous integration of compute, memory, and accelerator functions. As advanced packaging technologies mature and industry-standard interconnect specifications achieve widespread adoption, chiplets are positioned to become the dominant architectural paradigm for high-performance computing, artificial intelligence acceleration, automotive systems, and edge infrastructure applications. However, this architectural evolution demands equally sophisticated verification strategies capable of addressing the unique challenges inherent in distributed multi-die systems. The validation of chiplet-based designs requires methodologies that can effectively manage cross-die clock domain crossing verification with independent timing sources, distributed cache coherency protocols spanning multiple dies with complex ordering dependencies, high-bandwidth interconnect compliance across diverse traffic classes, and multi-physics interactions involving electrical, thermal, and architectural domains. Traditional simulation approaches, constrained by limited execution throughput and inability to sustain realistic workload conditions, cannot scale to meet these verification demands. Hardware emulation has therefore emerged as the essential foundation for chiplet validation, providing the unique combination of near-real-time execution performance, comprehensive observability across all chiplets simultaneously, protocol monitoring for interconnect compliance, and capacity to support complete system-on-chip designs with multiple dies. Looking forward, the continued evolution of emulation methodologies incorporating artificial intelligence-assisted debug, cloud-scale verification infrastructure, and hybrid modeling approaches will further strengthen the verification capabilities necessary for increasingly complex chiplet ecosystems, enabling the semiconductor industry to achieve first-pass silicon success and accelerate time-to-market for next-generation heterogeneous integrated systems.

References

[1] Michael Raymond et al., "Chiplet-Based Architectures Redefining the Future of System-on-Chip (SoC) Design," ResearchGate, December 2025. [Online]. Available: https://www.researchgate.net/publication/398419703

[2] Swathi Narsimhan et al., "Chiplets on Wheels: Review Paper on Holistic Chiplet Solutions for Autonomous Vehicles," ResearchGate, May 2024. [Online]. Available: https://www.researchgate.net/publication/381122272

[3] Murali Krishna Reddy Madalapu et al., "Emerging Chiplet-Based Architectures for Heterogeneous Integration," ResearchGate, March 2025. [Online]. Available: https://www.researchgate.net/publication/389855062

[4] Vivek Gujar, "Chiplet Technology Revolutionizing Semiconductor Design-A Review," ResearchGate, February 2024. [Online]. Available: https://www.researchgate.net/publication/378156097

[5] Debendra Das Sharma et al., "High-performance power-efficient three-dimensional system-in-package designs with universal chiplet interconnect express," ResearchGate, February 2024. [Online]. Available: https://www.researchgate.net/publication/378313501

[6] Quinfen Hao et al., "Survey of Chiplet Technology: SoC Architecture, Interconnect, EDA, and Advanced Packaging," ResearchGate, January 2025. [Online]. Available: https://www.researchgate.net/publication/397932176

[7] F. Khalid, N. Sharif, N. Ramzan, and O. Hasan, "Quantitative Analysis of State-of-the-Art Synchronizers: Clock Domain Crossing Perspective," in 2011 IEEE International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 2011, pp. 1-5, doi: 10.1109/ICET.2011.6048485. [Online]. Available: https://www.researchgate.net/publication/233388727

[8] Swathi Narashiman et al., "A Survey on Chiplet Interconnects," 31 May 2024. [Online]. Available: https://arxiv.org/abs/2406.00182

[9] Venkatavaraha Chakravarthy Kanumetta et al., "Design Verification of Multi-Chiplet AI Accelerators: Challenges and Solutions," ResearchGate, November 2025. [Online]. Available: https://www.researchgate.net/publication/398093696

[10] Arturo Salz & Johannes Stahl, "Chip Verification Insights for Multi-Die Systems," Synopsys Blog, 1 August 2023. [Online]. Available: https://www.synopsys.com/blogs/chip-design/multi-die-systems-chip-verification.html