FPGA Prototyping as a Verification Tool in Semiconductor Design

Niranjana Gurushankar

Abstract

The increasing complexity of integrated circuits (ICs) has made verification a critical aspect of the semiconductor design flow. Traditional simulation-based verification methodologies are struggling to keep pace with the growing design size and software complexity. FPGA prototyping offers a compelling alternative by providing a hardware-based platform for early software development, real-time system validation, and accelerated verification. This paper explores the role of FPGA prototyping as a powerful verification tool in modern semiconductor design. It examines the benefits, challenges, and methodologies associated with FPGA prototyping, highlighting its ability to bridge the gap between pre-silicon verification and post-silicon validation. Furthermore, the paper discusses various use cases and advanced techniques, such as hybrid emulation and virtual prototyping, to demonstrate the versatility and effectiveness of FPGA prototyping in tackling the verification challenges of today's complex IC designs.

Keywords: Chip Verification, FPGA Prototyping, Chip design, SoC Verification, System-Level Validation, Pre-Silicon Validation

Introduction

The semiconductor industry is driven by an insatiable demand for higher performance, lower power consumption, and increased functionality in electronic devices. This has led to a surge in the complexity of integrated circuits (ICs), pushing the limits of traditional design and verification methodologies. Verification, the process of ensuring that a design meets its specifications, has become a critical bottleneck in the semiconductor design flow. It consumes a significant portion of the overall design cycle, often exceeding 70% [1].

Traditionally, verification relied heavily on simulation-based techniques. While valuable for verifying individual modules and basic functionality, simulation struggles to handle the complexities of modern system-on-chip (SoC) designs. Simulating intricate interactions between hardware and software, real-time behavior, and performance characteristics at the system level becomes computationally expensive and time-consuming. This limitation has fueled the adoption of hardware-based verification approaches, with FPGA prototyping emerging as a prominent solution.

FPGA prototyping involves implementing a design on a field-programmable gate array (FPGA) platform to create a physical prototype of the target IC. This prototype enables real-time execution of the design, facilitating early software development, comprehensive system validation, and accelerated verification. By operating at speeds orders of magnitude faster than simulation, FPGA prototypes allow for extensive testing with real-world stimuli and realistic workloads.

This paper delves into the role of FPGA prototyping as a powerful verification tool in modern semiconductor design. It explores the advantages, challenges, and methodologies associated with this approach. The paper also examines various use cases and advanced techniques, such as hybrid emulation and virtual prototyping, to showcase the versatility and effectiveness of FPGA prototyping in addressing the verification challenges of today's complex IC designs. However, deeper challenges and advantages exist in using FPGA for chip validation, as discussed in subsequent sections in this paper.

Challenges in Silicon Validation:

The relentless pursuit of higher performance and increased functionality in System-on-Chips (SoCs) has led to unprecedented design complexity, pushing traditional verification methodologies to their limits. Key challenges include:

Simulation Bottlenecks:

Verifying intricate SoCs, often containing billions of transistors and complex software stacks, through simulation becomes computationally prohibitive, resulting in lengthy verification cycles and delayed time-to-market.

Limited System-Level Validation:

Simulation struggles to accurately capture the real-time behavior and intricate interactions between diverse IP blocks, embedded processors, and software components within a complex SoC.

Inadequate Software Development Support:

Traditional pre-silicon verification environments offer limited opportunities for early software development and integration, hindering the ability to identify and address software-related issues before silicon arrival.

FPGA Prototyping

FPGA prototyping provides a compelling solution by offering a hardware-based platform for accelerated verification and early system validation. This approach addresses the limitations of traditional methods through:

Hardware Acceleration:

Implementing the SoC design on an FPGA enables real-time execution speeds, orders of magnitude faster than simulation, facilitating extensive testing with real-world stimuli and realistic workloads.

Early Software Development:

FPGA prototypes provide a pre-silicon platform for software development, enabling software teams to begin developing, testing, and debugging software well before silicon availability.

System-Level Validation:

Prototypes allow for comprehensive system-level validation, capturing the complex interactions between hardware and software components in a realistic environment, closely resembling the final silicon.

Uses of FPGA Prototyping

FPGA prototyping finds diverse applications across the SoC design flow:

Early Software Bring-up: Provides a platform for software developers to develop, test, and debug software on a representative hardware platform, long before silicon is available.

Hardware/Software Co-verification: Enables concurrent hardware and software development, facilitating the identification and resolution of integration issues early in the design cycle.

System Validation with Real-World Interfaces: Allows for comprehensive system-level validation with real-world data and interfaces, ensuring the SoC meets performance and functionality requirements in its intended application.

Performance Analysis and Optimization: Facilitates performance analysis and optimization by running real-time benchmarks and workloads on the prototype, identifying performance bottlenecks and guiding design refinements.

Regression Testing: Accelerates regression testing by executing large test suites at high speeds, ensuring design stability and functional correctness throughout the development process.

Impact of FPGA Prototyping

The adoption of FPGA prototyping has a profound impact on SoC design:

Reduced Time-to-Market: By accelerating verification and enabling early software development, FPGA prototyping helps shorten overall development cycles, leading to faster time-to-market.

Improved Silicon Quality: Comprehensive system-level validation and real-time debugging capabilities lead to higher silicon quality and reduced risk of costly post-silicon bugs and re-spins.

Lower Development Costs: Early detection and resolution of bugs through prototyping can significantly reduce expensive re-spins and design revisions, contributing to lower overall development costs.

Increased Productivity: FPGA prototyping empowers software and hardware teams to work concurrently, fostering collaboration and improving overall productivity and efficiency.

Scope of FPGA Prototyping

The scope of FPGA prototyping extends beyond traditional verification tasks:

Hybrid Emulation: Combining FPGA prototyping with simulation environments, such as emulation, allows for efficient verification of complex SoCs by leveraging the strengths of both approaches. This enables detailed analysis of specific modules while maintaining high-speed execution for system-level validation.

Virtual Prototyping: Integrating virtual models with FPGA prototypes enables early software development and architectural exploration even before RTL is available. This facilitates early software development and system-level performance analysis.

Hardware-in-the-Loop Testing: FPGA prototypes can be used for real-time testing with physical components and systems, ensuring seamless integration and interoperability of the SoC within its intended environment.

By addressing the limitations of traditional verification methodologies, FPGA prototyping is transforming the SoC design landscape. Its ability to accelerate verification, enable early software development, and provide comprehensive system-level validation makes it an indispensable tool for tackling the complexities of modern SoC design.

Future direction:

As discussed above with all the challenges and advantages in using FPGA prototyping, there are also possible future directions in using this methodology with opportunities for future research in Software and Hardware areas. Prominent areas are discussed below.

AI-Driven FPGA Prototyping:

Imagine AI algorithms optimizing FPGA partitioning, resource allocation, and even generating test cases. This could significantly accelerate the prototyping process and improve verification efficiency.

Cloud-Based FPGA Prototyping:

Accessing FPGA prototypes through the cloud could democratize this technology, making it available to smaller companies and researchers without the need for expensive hardware investments.

Formal Verification with FPGA Prototyping:

Integrating formal verification techniques with FPGA prototyping could provide stronger guarantees of correctness and completeness.

FPGA Prototyping for Security Verification: As security becomes increasingly critical, FPGA prototypes can be used to validate hardware security measures and identify vulnerabilities early in the design cycle.

Conclusion

In this paper, an overview of how FPGA prototyping is proving indispensable in verifying increasingly complex SoCs, along with a deep dive specifically into the challenges of using FPGA. The paper also covers how it accelerates verification, enables early software development, and provides comprehensive system-level validation, ultimately leading to faster time-to-market, improved silicon quality, and reduced development costs. As SoCs continue to evolve, FPGA prototyping will remain crucial for tackling design challenges and ensuring the creation of innovative and reliable electronic devices.

References

1.Bauer, J., et al. (2012). A Case Study on FPGA-Based Prototyping of a Complex DSP System. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20(12), 2224-2233.

2.Noda, T., et al. (2013). A Fast and Accurate HW/SW Co-Verification Environment Using Transaction Level Modeling and FPGA Prototyping. In Proceedings of the 50th Annual Design Automation Conference (pp. 1-6).

3.Villarreal, J., et al. (2015). Hybrid Emulation: Combining Virtual Prototypes and FPGA-Based Prototypes for System-Level Validation. IEEE Design & Test, 32(1), 74-83

4.Chang, H., et al. (2014). Overcoming the Challenges of FPGA-Based Prototyping for SoC Designs. IEEE Design & Test, 31(3), 68-77

5.Chen, W., Ray, S., Bhadra, J., Abadir, M., & Wang, L.C. (2017). Challenges and Trends in Modern SoC Design Verification. IEEE Design & Test, 34(1), 7-22.
6.Mishra, P., Morad, R., Ziv, A., & Ray, S. (2017). Post-Silicon Validation in the SoC Era: A Tutorial Introduction. IEEE Design & Test, 34(1), 68-92.
7.Goeders, J., & Wilton, S.J.E. (2017). Signal-Tracing Techniques for In-System FPGA Debugging of High-Level Synthesis Circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(1), 83-96
8.Hayashi, T., & Tomiyama, H. (2014). FPGA-Based Prototyping Methodology for Early Software Development. IEICE Transactions on Information and Systems, E97-D(4), 881-888