International Journal of Business Quantitative Economics and Applied Management Research
Volume-5, Issue-9, February-2019 ISSN No: 2349-5677

POST-SILICON VALIDATION STRATEGIES FOR SEMICONDUCTOR DESIGNS

Niranjana Gurushankar
MEng Electrical Eng, George Washington University
gniranjana96@gmail.com

Abstract

Post-silicon validation is a critical phase in semiconductor design post the Pre-Silicon validation phase, ensuring the functionality and performance of fabricated chips meets the specifications and the expectations. This paper explores the challenges, complexities and also the benefits of post-silicon validation, examining various standard approaches and emerging trends. I discuss various strategies, including on-chip instrumentation, emulation, and formal verification, automated test equipment (ATE) highlighting their strengths and limitations. The paper also analyzes the impact of increasing design complexity and shrinking time-to-market windows on validation methodologies. Finally, I present a comprehensive outlook on the future of post-silicon validation, emphasizing the need for adaptable and efficient techniques to address evolving industry demands.

Keywords: Post-silicon validation, chip validation, semiconductor design, silicon bugs, software validation holes, emulation, formal verification, power and performance, automated test equipment.

I. INTRODUCTION

The semi-conductor industry thrives on continuous innovation, pushing the boundaries of chip complexity and performance with the increase in chip usage. As the designs become more complicated, ensuring their correctness and reliability has grown increasingly challenging with Power, Performance and Cost playing a huge role. Post-silicon validation, the process of verifying a chip's functionality after manufacturing, plays a crucial role in guaranteeing product quality and mitigating costly errors [1].

Traditional validation methods often struggle to keep pace with the escalating complexity and shrinking time-to-market constraints [2]. This necessitates a shift towards more sophisticated and efficient validation approaches [3]. This paper delves into the intricacies of post-silicon validation, examining its challenges, exploring innovative solutions, and discussing its future trajectory.

II. CHALLENGES IN PRE-SILICON VALIDATION

Before diving into various challenges faced during bringing up a chip until Tape-Out which is a term used when the pre-silicon chip work has been completed. There are several stages in a chip

production starting from the Architecture phase which ends with the Tapeout phase. One of the phases is Silicon Validation, a modern semiconductor chip is a formidable undertaking, fraught with challenges that arise from the sheer complexity of these devices and the demanding requirements they must meet. There could be a question on why we need this validation when we have already tested out the design in the Software world or Simulation world? That's where this paper helps you with knowing some knowledge on why Post-Silicon validation is a critical step. This section focuses on the key challenges,

1. Massive Scale and Integration

Modern chips have billions of transistors, multiple IP blocks, and intricate interconnects. This makes it difficult to create comprehensive test environments and achieve sufficient coverage [4].

2. Hardware/Software Co-verification

Validating the interaction between hardware and software before silicon is available is challenging. Accurate models and efficient co-simulation techniques are crucial [5].

3. Verification Environment Limitations

Simulating complex designs at the RTL level can be extremely time-consuming, slowing down the validation process. This necessitates techniques like emulation and formal verification to speed up verification [6].

4. Corner Case Identification

Many bugs occur only under specific and rare combinations of events. Identifying these corner cases requires sophisticated test generation and coverage analysis techniques [7].

III. NEED FOR POST-SILICON VALIDATION

With all the challenges which were discussed above, this section of the paper shows In the face of these challenges, post-silicon validation emerges as an indispensable stage in the semiconductor design lifecycle. It bridges the gap between pre-silicon verification and real-world deployment by:

1. Power and Performance Validation

Meeting the Power Budgets especially in mobile and embedded systems, exceeding power constraints can drastically reduce battery life, cause overheating, and even damage the device. This is a major concern for consumer satisfaction and product viability [8].These two above benchmarks that can be validated only on a Silicon, which gives you the real time frequency and timing. Unexpected power spikes or performance drops can occur under specific conditions not easily simulated. Post-silicon validation, especially with prolonged testing, is more likely to uncover these [9].

Power and performance are also heavily influenced by software. Post-silicon validation reveals how software workloads impact these metrics, enabling optimization across the hardware-software stack [10].

2. Uncovering Hidden Bugs

Post-silicon validation has real hardware and realistic workloads, which could help engineers find subtle bugs that may have eluded pre-silicon verification efforts [11]. There are also limited test scenarios, even the best pre-silicon test benches can't cover every possible scenario a chip might encounter in the real world e.g. environmental factors etc.

3. Ensuring System-Level Functionality

It verifies the chip's operation within its intended system environment, encompassing interactions with software, peripherals, and other components [12].

4. Enabling Design Optimization

The insights gained from post-silicon validation can inform design optimizations for future revisions or product generations, contributing to continuous improvement [2].

IV. CHALLENGES IN POST SILICON VALIDATION

This section focuses on the key challenges when it comes to Post Silicon Validation technique,

1. Scale and Integration

System-on-Chips (SoCs) integrate billions of transistors with diverse functionalities, creating a massive state space that defies exhaustive testing [13]. As Ben-Porath and Mitra highlight, "the sheer complexity of modern designs makes it impossible to guarantee correctness through pre-silicon verification alone." This challenge is further exacerbated by the increasing prevalence of multi-core processors and heterogeneous architectures [14].

2. Emerging Technologies

Moore's Law states the number of transistors on an integrated circuit [IC] will double every two years leading to unpredictable behaviors and subtle interactions that are difficult to model accurately before fabrication [15]. As the International Technology Roadmap for Semiconductors (ITRS) emphasized, new technology nodes bring new failure mechanisms, further complicating the validation process.

3. The Software-Hardware Intertwining

This is one of the key problems of the co-verification challenge which I was talking about earlier, the intricate interplay between hardware and software in modern computing systems necessitates robust co-verification strategies [16]. Traditional approaches, which often treat hardware and software verification as separate tasks, struggle to address the complex interactions and dependencies between them [17].

4. Real-World Workloads

Post-silicon validation often relies on real silicon, which may make it even harder to debug certain bugs since the debug ports are very limited. Capturing waveforms is not possible on real silicon which makes it even harder. Validating the chip's interaction with real operating systems and application software is crucial to uncover hidden bugs and performance bottlenecks [18].

5. Shrinking Product Cycles

The semiconductor industry faces immense pressure to deliver products quickly, necessitating efficient validation methodologies that can keep pace with shrinking product development cycles. As highlighted in a 2014 industry study by Malik et al. [2], the increasing complexity and shrinking time-to-market windows demand a paradigm shift in validation methodologies.

V. SOLUTION & STRATEGIES

The strategies mentioned below can help overcome the challenges mentioned in the previous section.

1. On-Chip Instrumentation

Embedding dedicated logic for monitoring and controlling chip behavior provides valuable insights during validation. Techniques like scan chains, trace buffers, and assertion checkers enable observability and debug capabilities. For e.g reproducing and isolating the bugs since they occur only under specific conditions or in a particular rare corner case. On-chip instrumentation helps you reproduce these conditions by allowing you to force specific states or inject controlled stimuli. This targeted control, combined with the ability to observe internal signals, makes it far easier to isolate the root cause of failures.

2. Emulation and Prototyping

Pre-silicon emulation platforms offer a high-fidelity representation of the chip's behavior, allowing software development and early validation before fabrication [16, 19]. FPGA-based prototypes provide another avenue for hardware-software co-verification. FPGA validation gives some extra confidence before silicon comes back from the manufacturing company that the basic corner cases work as expected since you dump the entire design on FPGA and for time being that works as the silicon.

3. Formal Verification

Mathematical techniques rigorously prove the correctness of specific design properties, complementing traditional simulation-based approaches [20, 21]. Formal methods are particularly useful for verifying critical modules and identifying corner-case bugs.

4. Post-Silicon Debug and Diagnosis

Sophisticated tools and techniques aid in identifying and isolating silicon bugs. Utilizing scan chains, logic analyzers, and advanced debug probes enables efficient fault localization.

5. Automated Test Equipment (ATE)

ATE systems always incorporate industry-standard test suites and methodologies, ensuring comprehensive coverage of common failure modes. That is done by enabling rapid testing of a large number of chips, increasing the likelihood of uncovering statistically rare bugs. Automated test execution also eliminates any human error and ensures consistent test conditions across multiple devices.

VI. FUTURE DIRECTIONS

As engineering technology or semiconductor industry has become increasingly prevalent, post silicon validation strategies continue growing, with opportunities for future research. Prominent areas are captured below.

1. High-Performance Computing

Validating complex CPUs, GPUs, and AI accelerators requires rigorous testing to ensure performance, power efficiency, and reliability.

2. Mobile and Embedded Systems

SoCs for smartphones, IoT devices, and automotive applications demand thorough validation to guarantee functionality and security.

3. Networking and Communications

Chips powering network infrastructure and data centers necessitate robust validation to maintain high availability and data integrity.

4. Cloud-Based Validation

Utilizing cloud computing resources can provide scalable and cost-effective solutions for managing the increasing computational demands of validation.

5. Security Validation

With the rise of hardware security threats, integrating security validation into the post-silicon process is becoming increasingly important.

VII. CONCLUSION

In this paper, an overview of how post-silicon validation remains a critical challenge and a dynamic field in semiconductor design. As chip complexity continues to escalate, embracing innovative strategies and methodologies is crucial. Combining on-chip instrumentation, emulation, formal verification, and advanced debug techniques empowers engineers to effectively validate designs and ensure product quality. The future of post-silicon validation lies in adaptable, efficient, and intelligent solutions that address the evolving demands of the semiconductor industry.

REFERENCES

1. A. Adir et al., "Challenges and Solutions in Post-Silicon Validation of High-End Processors," Proc. FMCAD, 2019.
2. S. Mitra et al., "Post-Silicon Validation Opportunities, Challenges and Recent Advances," Design Automation Conference (DAC), 2010.
3. A. Ben-Porath and S. Mitra, "Post-Silicon Bug Detection," Foundations and Trends® in Electronic Design Automation, vol. 4, no. 4, pp. 257-381, 2011.
4. F. Corno et al., "A Survey of Hardware/Software Co-Verification Techniques," Journal of

Systems Architecture, vol. 54, no. 3-4, 2008.
5. F. Corno et al., "A Survey of Hardware/Software Co-Verification Techniques," Journal of Systems Architecture, vol. 54, no. 3-4, 2008.
6. J. Bergeron, "Writing Testbenches: Functional Verification of HDL Models," Springer, 2003
7. M. Abramovici et al., "Digital Systems Testing and Testable Design," Wiley-IEEE Press, 2000
8. J. Srinivasan et al., "The Case for Lifetime Reliability-Aware Design and Validation of SoCs," Proc. Design Automation Conference (DAC), 2010.
9. P. Bernardi et al., "On the Effectiveness of Post-Silicon Validation for Modern Microprocessors," IEEE Transactions on Computers, vol. 65, no. 10, 2016
10. V. Bertacco and I. Ghosh, "Software Validation of Hardware Features," IEEE Design & Test of Computers, vol. 22, no. 4, 2005
11. A. Ben-Porath and S. Mitra, "Post-Silicon Bug Detection," Foundations and Trends® in Electronic Design Automation, vol. 4, no. 4, pp. 257-381, 2011.
12. J. Baumgartner et al., "Scalable Post-Silicon Validation of Heterogeneous Systems," IEEE Design & Test, vol. 34, no. 1, 2017.
13. A. Ben-Porath and S. Mitra, "Post-Silicon Bug Detection," Foundations and Trends® in Electronic Design Automation, vol. 4, no. 4, pp. 257-381, 2011.
14. E. J. Marinissen et al., "A Taxonomy of Bugs in Digital Systems," ACM Computing Surveys, vol. 48, no. 4, 2016.
15. P. Bernardi et al., "On the Effectiveness of Post-Silicon Validation for Modern Microprocessors," IEEE Transactions on Computers, vol. 65, no. 10, 2016.
16. F. Corno et al., "A Survey of Hardware/Software Co-Verification Techniques," Journal of Systems Architecture, vol. 54, no. 3-4, 2008.
17. V. Bertacco and I. Ghosh, "Software Validation of Hardware Features," IEEE Design & Test of Computers, vol. 22, no. 4, 2005.
18. M. Abramovici et al., "On-Chip Debugging of Network-on-Chips," IEEE Design & Test of Computers, vol. 23, no. 4, 2006.
19. C. Huang et al., "Emulation-Based Prototyping for Software Development," IEEE Design & Test of Computers, vol. 22, no. 1, 2005
20. T. Kropf, "Introduction to Formal Hardware Verification," Springer, 1999.
21. E. M. Clarke et al., "Model Checking," MIT Press, 1999.
22. J. Bergeron, "Writing Testbenches: Functional Verification of HDL Models," Springer, 2003.
23. H. Foster, "Assertion-Based Design," Springer, 2004.
24. L. W. Nagel et al., "Design for Testability - A Survey," IEEE Transactions on Computers, vol. C-30, no. 6, 1981