International Journal of Core Engineering & Management
Volume-6, Issue-06, 2019 ISSN No: 2348-9510

DFT IN SEMICONDUCTOR DESIGN VERIFICATION

Niranjana Gurushankar
MEng Electrical Engineering,
George Washington University

Abstract

Design for Testability (DFT) is crucial for verifying increasingly complex integrated circuits. This paper explores DFT's role in enhancing verification coverage and reducing time-to-market by facilitating fault detection and diagnosis. We examine various DFT methodologies, including scan insertion, ATPG, memory BIST, and logic BIST, and discuss their applications in modern semiconductor design. Furthermore, we analyze emerging DFT trends such as at-speed testing and low-power DFT, highlighting their contribution to achieving robust and reliable ICs.

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 semiconductor industry's relentless pursuit of Moore's Law has led to an exponential increase in the complexity and density of integrated circuits (ICs) [1]. This surge in complexity presents formidable challenges in ensuring the functional correctness and reliability of these designs [2]. This is where Design for Testability (DFT) emerges as a critical enabler, bridging the gap between design and verification.

DFT, in essence, is the practice of incorporating testability features into the design during the early stages of the design cycle [3]. While historically associated with manufacturing testing, DFT has evolved into a powerful tool for design verification [4]. By providing enhanced observability and controllability of internal circuit nodes, DFT facilitates comprehensive fault detection and diagnosis [5]. This paper delves into the multifaceted role of DFT in modern semiconductor design verification, examining its methodologies, applications, and emerging trends. However, deeper challenges exist in DFT methodology, as discussed in subsequent sections in this paper.

II. DFT METHODOLOGIES

This section explores the core DFT techniques employed in contemporary IC design, analyzing their strengths and limitations in the context of verification. We focus on four key methodologies: scan-based testing, ATPG algorithms, memory BIST, and logic BIST.

2.1 Scan-Based Testing

Scan-based testing is a widely adopted DFT technique that enhances the controllability and observability of internal circuit nodes [6]. It involves modifying registers within the design to form shift registers, collectively known as a scan chain. During testing, these scan chains allow test patterns to be shifted into the circuit and responses to be shifted out, effectively transforming a sequential circuit into a combinational one for testing purposes. The advantage of using this scan based testing is to have high fault coverage, where the scan chains provide excellent access to internal nodes, enabling high fault coverage for stuck-at faults and some transition faults. This testing needs very simplified test generation since its relatively straightforward, as it involves targeting combinational logic.

This testing comes with its own limitation which is going to be discussed in this paper as well, one of the major limitations being increase in area, adding scan chains introduces area overhead due to the extra logic and routing required. Area affects performance, due to the added delay in the signal paths.

2.2 Automatic Test Pattern Generation (ATPG)

ATPG algorithms automate the process of generating test patterns for detecting faults in digital circuits [7]. These algorithms analyze the circuit structure and generate input stimuli that propagate fault effects to observable outputs. ATPG tools are essential for achieving high fault coverage in scan-based testing. Looking at the strengths of using the above method is that it reduces the manual effort required to create test patterns which reduces a lot of human error as well. ATPG algorithms can achieve high fault coverage for various fault models. Computational complexity with larger and complex designs could be intensive.

2.3 Memory Built-In Self-Test (MBIST)

MBIST is a DFT technique that integrates test structures within memory arrays to enable self-testing [8]. These structures generate test patterns and evaluate responses on-chip, eliminating the need for external testing equipment for memory testing. This technique helps in achieving high coverage for various memory faults. This helps with getting more coverage with the memory testing and also this reduces memory test time compared to external testing.

Few of the limitations using this method is the area overhead since adding MBIST adds extra area to the existing memory design. Diagnosing the precise location of memory faults with this methodology can be challenging.

2.4 Logic Built-In Self-Test (LBIST)

LBIST is a DFT technique that employs on-chip circuitry to generate pseudo-random patterns and analyze responses for testing logic blocks [9]. It offers a cost-effective solution for testing complex logic circuits without requiring external test equipment. LBIST eliminates the need for any expensive external testing.Since the built in design can be used instead of any extra design. Also to test the functionality of the logic, the frequency has to be operational and this method helps with that which is a very important feature. Just like any other method, there are some limitations as well which are, fault coverage. To achieve high fault coverage with LBIST could be challenging especially with complex designs.

III. FUTURE DIRECTIONS WITH DFT METHODOLOGIES

As the semiconductor industry continues its relentless pursuit of miniaturization and performance, DFT methodologies must evolve to address emerging challenges and leverage new opportunities. Here are some key future directions:

3.1 AI-Driven DFT

Artificial intelligence and machine learning are poised to revolutionize DFT. AI algorithms can be used to optimize test pattern generation, reduce test time, and improve fault coverage. Machine learning can also be employed for fault diagnosis, enabling faster identification and rectification of design errors.

3.2 DFT for 3D ICs

Three-dimensional integrated circuits (3D ICs) present unique challenges for DFT due to their complex structures and increased interconnect density. New DFT techniques are needed to address issues like limited access to internal nodes and thermal challenges in 3D stacked dies.

3.3 Security-Aware DFT

With the growing concern for hardware security, DFT methodologies must consider security threats. Secure DFT techniques can be used to prevent malicious attacks, detect hardware Trojans, and ensure the integrity of ICs.

3.4 DFT for Emerging Technologies

New technologies like quantum computing and neuromorphic computing will require innovative DFT solutions. Research is needed to develop DFT techniques that can effectively test and verify these unconventional computing paradigms.

3.5 Low-Power DFT

With the increasing emphasis on power efficiency, DFT methodologies must minimize their power consumption. Low-power DFT techniques can reduce test power and contribute to the overall energy efficiency of ICs.

IV. CONCLUSION

This paper explored the crucial role of Design for Testability (DFT) in modern semiconductor design verification. The key DFT methodologies, their limitations and stregths were discussed, including scan-based testing, ATPG, MBIST, and LBIST. DFT significantly enhances fault coverage, reduces time-to-market, and improves the reliability of integrated circuits. As technology advances, DFT will continue to evolve, with emerging trends like AI-driven DFT and security-aware DFT shaping the future of IC design and verification. By embracing these advancements, the semiconductor industry can ensure the development of robust and reliable electronic systems for years to come. Further research is needed to evolve on optimizing DFT methodologies for complex design which involves both analog and digital world.

REFERENCES

1. Moore, Gordon E. "Cramming more components onto integrated circuits." Electronics, Vol. 38, No. 8, April 19, 1965.
2. Bahadur, R., et al. "System-on-a-chip verification challenges and solutions: a survey." Journal of Systems Architecture, Vol. 56, No. 11, 2010, pp. 519-534.
3. Abramovici, Miron, Melvin A. Breuer, and Arthur D. Friedman. Digital systems testing and testable design. Computer Science Press, 1990.

4. Benso, A., et al. "A survey on design for testability techniques for analog and mixed-signal integrated circuits." Journal of Electronic Testing: Theory and Applications, Vol. 25, No. 4, 2009, pp. 223-239.
5. Wang, Laung-Terng, Cheng-Wen Wu, and Xiaoqing Wen. VLSI test principles and architectures: design for testability. Morgan Kaufmann, 2006.
6. Bushnell, Michael L., and Vishwani D. Agrawal. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. Springer Science & Business Media, 2000.
7. Cheng, Kwang-Ting Tim, et al. "Test generation for sequential circuits." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 7, No. 10, 1988, pp. 1081-1093.
8. Zorian, Yervant. "Embedded-memory test and repair: infrastructure IP for SOC yield." IEEE Design & Test of Computers, Vol. 20, No. 3, 2003, pp. 58-66.
9. Wunderlich, Hans-Joachim. "LBIST—a special case of BIST." Journal of Electronic Testing: Theory and Applications, Vol. 18, No. 4, 2002, pp. 353-367.