Performance of Generic and Recursive Pseudo Exhaustive Two-Pattern Generator

Akanksha Pandey[1], Pravin Tiwari [2]

Electronics & Communication Department, Takshshila Institute of Technology, Jabalpur (RGPV), India

Abstract: ***The main objective of this research is to design a Built-in self-test (BIST) technique based on pseudo-exhaustive testing. Two pattern test generator is used to provide high fault coverage. To provides fault coverage of detectable combinational faults with minimum number of test patterns than the conventional exhaustive test pattern generation, increases the speed of BIST and may posses minimum Hardware Utilization. A Built-in self-test (BIST) is a commonly used design technique that allows a circuit to test itself. Built-in self-test (BIST) technique based on pseudo-exhaustive testing, two pattern test generator is used to provide high fault coverage. It is widely known that a large class of physical defects cannot be modeled as stuck-at faults. For example, a transistor stuck- open fault in a CMOS circuit can convert a combinational circuit under test (CUT) into a sequential one, while a delay fault may cause circuit malfunction at clock speed, although it does not affect the steady-state operation. Detection of such faults requires two-pattern tests. It is proposed to design a BIST circuit for fault detection using pseudo exhaustive two pattern generator using minimum hardware utilization and increasing the speed of BIST.***

Keywords: BIST, RTL, TPG, CUT, RPET, GPET, VHDL.

1. Introduction

In this paper, we start by presenting a generic and recursive pseudo exhaustive two-pattern generation scheme. Where as, A generic pseudo exhaustive two-pattern generation scheme generates an (n, k) pseudo exhaustive two-pattern test for any value of k, by enabling a proper input signal PE[k], $1\leq k\leq n$. The generic pseudo-exhaustive two-pattern generator consists of a generic counter, 1’s complement adder, a controller and a carry generator (C gen). The n-bit pattern PE [n: 1] is given as an input to the controller, generic counter and C_gen and the output A [n:1] is taken from the Accumulator**.**

VLSI circuits, containing millions of transistors, the utilization of the same BIST pattern generator to test more than one module can drive down the cost of BIST hardware. Modules whose inputs are driven (during BIST) from the same pattern generator may have different cone sizes. Two solutions have been proposed to this direction. One solution is to utilize recursive pseudo exhaustive testing with recursive pseudo exhaustive testing, (n, k) PETS are generated for all *k*=1, 2, 3…... n. An alternative solution is generic pseudo exhaustive testing, introduced in generic pseudo exhaustive generator can generate a (n, k) PETS for any value of by enabling a respective signal PE[k]. BIST pattern generators are commonly discerned into one pattern and two-pattern. One-pattern generators target the detection of combinational (mainly stuck-at) faults. For the correct behavior of the circuit delay fault is detected by the two pattern generator [1][2].

There is no need for fault simulation for exhaustive testing and pseudo-exhaustive testing. However, such a method requires extra design effort to partition the circuits into pseudo-exhaustive testable sub-circuits.

2. BIST Structure

The general Built-in Self-Test structure consists of three main parts see Figure1 the TPG (Test Pattern Generator) produces test patterns that are fed to the inputs of a Circuit under Test (CUT) and the responses of a circuit are then evaluated in a Response Evaluator (RE).[4][6]

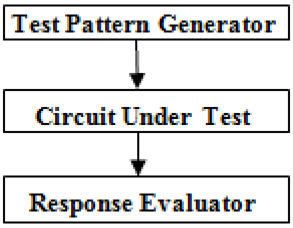

Figure 1: Basic BIST structure

BIST is performed when the functional circuitry is in normal operational mode. It can be done either concurrently or non concurrently. In concurrent online BIST, testing is conducted simultaneously during normal functional operation. The functional circuitry is usually implemented with coding techniques or with duplication and comparison. When an intermittent or transient error is detected, the system will correct the error on the spot, rollback to its previously stored system states, and repeat the operation, or generate an interrupt signal for repeated failures. In nonconcurrent online BIST, testing is performed when the functional circuitry is in idle mode. This is often accomplished by executing diagnosis software routines (macrocode) or diagnosis firmware routines (microcode). The test process can be interrupted at any time so that normal operation can resume. Offline BIST is performed when the functional circuitry is not in normal mode. This technique does not detect any real-time errors but is widely used in the industry for testing the functional circuitry at the system, board, or chip level to ensure product quality.[2][3]

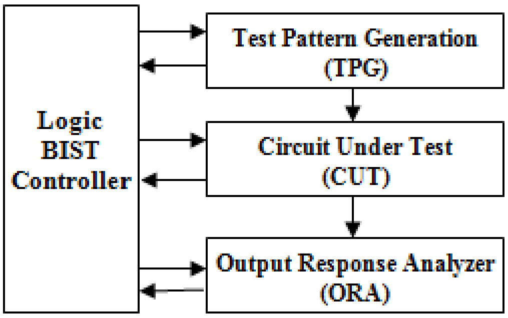

Figure 2: Typical logic BIST system

Figure 2 shows a typical logic BIST system using the structural offline BIST technique. The test pattern generator (TPG) automatically generates test patterns for application to the inputs of the circuit under test (CUT). [4] The output response analyzer (ORA) automatically compacts the output responses of the CUT into a signature. Specific BIST timing control signals, including scan enable signals and clocks, are generated by the logic BIST controller for coordinating the BIST operation among the TPG, CUT, and ORA. The logic BIST controller provides a pass/fail indication once the BIST operation is complete. It includes comparison logic to compare the final signature with an embedded golden signature, and often comprises diagnostic logic for fault diagnosis.

3. BIST Test Pattern

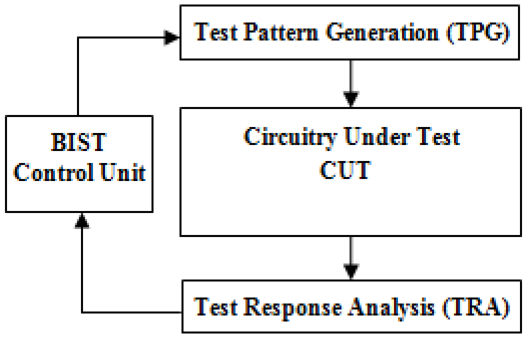

Figure 3: Test Pattern Generator

The fault coverage that we obtain for various fault models is a direct function of the test patterns produced by the Test. Pattern Generator (TPG) and applied to the CUT shown in Figure 3. This section presents an overview of some basic TPG implementation techniques used in BIST approaches.[6]

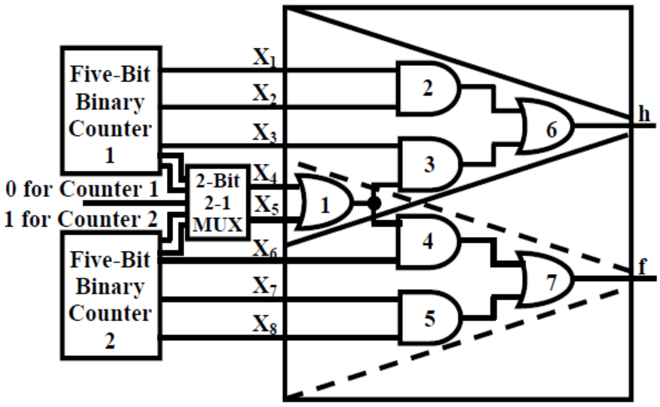

Figure 4: Pseudo-exhaustive pattern generator

Figure 4 shows the circuit partitioning for pseudo-exhaustive pattern generation can be done by *cone segmentation* as shown in Figure 4 Here, a cone is defined as the fan-ins of an output pin. If the size of the largest cone in K, the patterns must have the property to guarantee that the patterns applied to any K inputs must contain all possible combinations. In Figure 4.2, the total circuit is divided into two cones based on the cones of influence. For cone 1 the PO *h* is influenced by X_1, X_2, X_3, X_4 and X_5 while PO *f* is influenced by inputs X_4, X_5, X_6, X_7 and X_8. Therefore the total test pattern needed for exhaustive testing of cone 1 and cone 2 is (2^5 +2^5) = 64. But the original circuit with 8 inputs requires 2^8 = 256 test patterns exhaustive test.[3][8]

4. Design and Implementations

Pseudo exhaustive test pattern generators provide for complete fault coverage without the need for fault simulation. In pseudo exhaustive less number of test patterns is used.

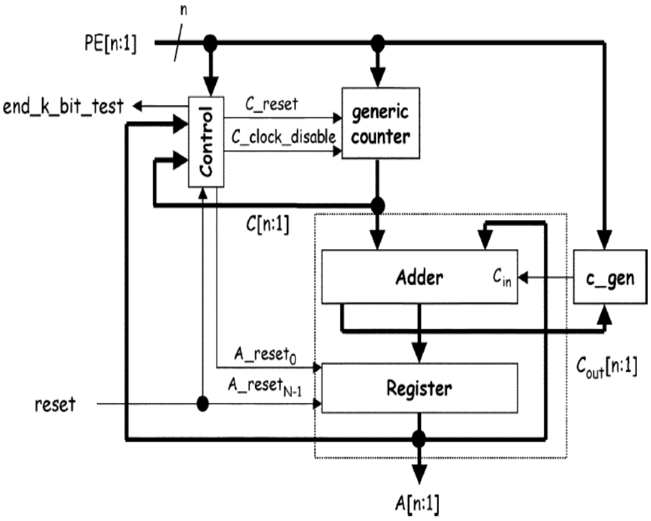

Figure 5: Generic exhaustive pattern generator

In Figure 5, a generic pseudo-exhaustive two-pattern generator is a module with n inputs (PE [n:1]) and n outputs (A [n: 1]) that can generator a two pattern (n, k) pseudo exhaustive test set for any value of k, (k < n).

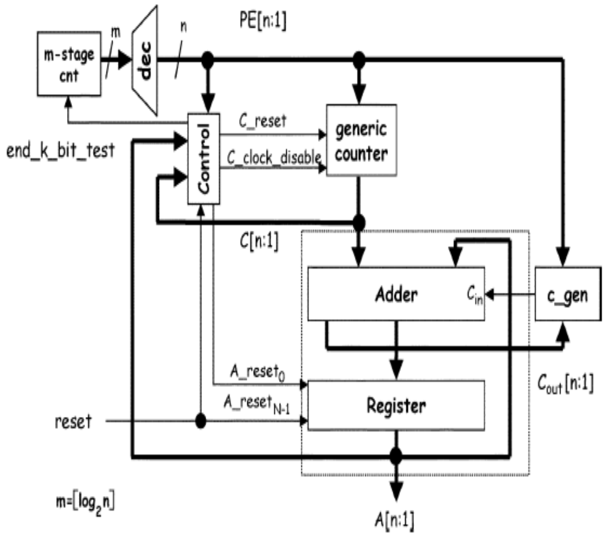

Figure 6: Recursive exhaustive pattern generator

The generic pseudo exhaustive two-pattern generator can be generalized into progressive two-pattern generator that generates all (n, k)-pseudo exhaustive two-pattern test vectors for all values of k. This technique is called as recursive pseudo exhaustive two-pattern generation. Recursive pseudo exhaustive two-pattern generator (RPET) shown in Figure 6 are also having the architecture similar to generic pseudo exhaustive two pattern generator. It's having y-stage counter and decoder added to generic pseudo exhaustive two-pattern generator's architecture.

5. Comparison of test strategies

Implementing a BIST strategy, the main issues are fault coverage, hardware overhead, test time overhead, and design effort. These four issues have very complicated relationship. Table 1 summarizes the characteristics of the test strategies mentioned earlier based on the four issues.

Table 1: Comparison of different test strategies

Test Generation Methodology	Fault Coverage	Hardware Overhead	Test Time Overhead	Design Effort
Stored Pattern	High	High	Short	Large
Exhaustive	High	Low	Long	Small
Pseudo-Exhaustive	High	High	Medium	Large
Pseudo-Random	Low	Low	Long	Small
Weighted Pseudo-Random	Medium	Medium	Long	Medium

6. Tool used for Simulation

Using a Hardware Description Language (HDL) allows you to test different design implementations early in the design flow. Use the synthesis tool to perform the logic synthesis and optimization into gates. Xilinx® FPGA devices allow you to implement your design at your computer. Since the synthesis time is short, you have more time to explore different architectural possibilities at the Register Transfer Level (RTL) .You can reprogram Xilinx FPGA devices to test several design implementations.[5]

7. Simulation Results Conclusion

BIST plays a vital role in modern VLSI technology.BIST two pattern generators play an increasingly important role in testing VLSI circuits and systems, due to a number of reasons, including higher coverage of sequential faults and assurance of correct temporal behaviour. Exhaustive and pseudo-exhaustive test generation schemes are commonly utilized to provide for extremely high fault coverage in combinational circuits. However, with pseudo-exhaustive generators, test generation time is greatly reduced.[2][7]

Table 2: Recursive pseudo exhaustive two pattern generator comparison

Scheme	*Exiting modules*	*Transformation*	*Gate equivalent*
[6]	Two n-stage register	Transform two registers into counters + XOR gates + n 2-input multiplexer + m-stage group counter + Detect	7 x n +XOR gates + 3 x n + 8 x m
[7]	Two n-stage register	Transform two registers into + n 2-input multiplexer + m-stage group counter + Detect	7 x n +XOR gates + 3 x n + 8 x m
[5]	Accumulator + register	m-stage counter + m-to-n decoder + Control + Detect + c_gen + Transform registers into generic counters	18 x n + 8 x m
[Proposed]	Accumulator + Counter	m-stage counter + m-to-n decoder + Control +c_gen + generic counters	15 x n + 8 x m

Table 3: Comparison of hardware overhead in gates

Scheme	No. of gates for n=7 [Proposed]	No. of gates for n=7 [5]	No. of gates for n=7 [6]	No. of gates for n=7 [7]
GPET	84	112	-	-
RPET	129	150	202	229

Table 4: Calculation of hardware overhead of the presented scheme

Module	**Hardware Overhead**	**Gates**
m-stage counter	m x DFF	8 x m
m-to-n decoder	≤ 2 x n	2 x n(1)
Control + Detect	9 gates + 3 DFF + 5 x n gates	9 + 24 + 5 x n
Generic counter	n x DFF + n x MUX21 + OR-grid	8 x n + 3 x n +2 x n(2)
Accumulator	n x (DFF + 2 x XOR + 3* NAND)	19 x n
C_ gen	n transmission	n

The hardware overhead of the modules is presented in Table 2, 3 & 4, we present the hardware overhead of the generic and recursive pseudo exhaustive two-pattern generators for the following cases:

- None of the required modules exists;
- An accumulator exists in the data path;
- An accumulator whose inputs are driven by the outputs of a register exists; in this case, the register can be transformed into a generic counter by adding 2-way multiplexers;
- An accumulator exists whose inputs are driven by an m-stage counter.

8. Conclusion

Pseudo-exhaustive BIST generators provide 100% fault coverage for detectable combinational faults with much fewer test vectors than exhaustive testing. Various techniques have been proposed for pseudo exhaustive test pattern generation for combinational faults.Built-In Self Test (BIST) techniques have been utilized to employ on-chip test generation and response verification, reducing the need for expensive external testing equipment

9. Acknowledgement

I am deeply grateful to my guide Prof. Praveen Tiwari and Prof. Sobhit Verma for their valuable discussions and theoretical insights.

References

[1] K. Nivitha1, Anita Titus "A BIST Circuit for Fault Detection Using Recursive Pseudo-Exhaustive Two Pattern Generator" International Journal of Modern Engineering Research (IJMER) www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-676-681

[2] IoannisVoyiatzis, Dimitris Gizopoulos and Antonis Paschalis, "Recursive Pseudo-Exhaustive Two Pattern Generation," IEEE trans. Very Large Scale Integration (VLSI) sys, vol. 18, no. 1, Jan 2010

[3] P. Dasgupta, S. Chattopadhyay, P. P. Chaudhuri, and I. Sengupta, "Cellular automata-based recursive pseudo-exhaustive test pattern generation IEEE Trans. Comput., vol. 50, no. 2, pp. 177–185, Feb. 2001.

[4] C. Chen and S. Gupta, "BIST test pattern generators for two-pattern testing-theory and design algorithms," *IEEE Trans. Comput.*, vol. 45,no. 3, pp. 257–269, Mar. 1996.

[5] Dr. V.K. Agrawal, Group Director, ISRO, Bangalore, Asst Professor, Department of ECE, DSCE, Bangalore, "Implementation of BIST Structure using VHDL for VLSI Circuits"International Journal of Engineering Science and Technology (IJEST) 2009.

[6] R. Srinivasan, S. K. Gupta, and M. Breuer, "Bounds on pseudoexhaustive test lengths," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 6, no. 3, Sep. 1998.

[7] VOYIATZIS, A. PASCHALIS, D. NIKOLOS, JOURNAL OF ELECTRONIC TESTING: "An Accumulator-Based BIST Approach for Two-Pattern Testing" Theory and Applications 15, 267–278 (1999) 1999 Kluwer Academic Publishers. Manufactured in The Netherlands.

[8] J. Rajski and J. Tyszer, "Recursive pseudoexhaustive test pattern generation," *IEEE Trans. Comput.*, vol. 42, no. 12, pp. 1517–1521, Dec.1993.M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization," In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 1951-1957, 1999. (conference style)

Author Profile

Akanksha Pandey is a pursuing student of M.Tech (P.G. course) in Embedded System & VLSI Design from Takshshila Institute of Engineering and Technology and received the degree of B.E in electronics & Communication stream in 2012. She presently works to determine the performance of Performance of Generic and Recursive Pseudo Exhaustive Two-Pattern Generator in BIST circuit.

Pravin Tiwari is an assistant Professor in the department of electronics & Communication in Takshshila Institute of Engineering and Technology. He already achieved the degree of B.E. (ECE) and M.Tech (VLSI).