International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 12 | Dec - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Design and Implementation of BIST Architecture for Low Power VLSI Applications using Verilog

Kavya M P[1], Bharath C M[2],Bhavana D M[3], Sadhana K R[4], Vijayalakshmi D[5]

[1]Assistant Professor, [2]Final year Student, [3]Final year Student , [4]Final year Student, [5]Final year Student
Department of Electronics and Communication Engineering,P E S Institue of Technology and Management,Shimoga

***

Abstract :A low-power Built-In Self-Test (BIST) scheme for a 32-bit multiplier, with an emphasis on enhancing fault coverage while m1inimizing power consumption. The proposed method utilizes a minimizing power usage is a key focus of the proposed approach, which employs a Linear Feedback Shift Register (LFSR) as part of its design, where the seed values are updated after every 2m 2^m 2m cycles. A counter is employed to monitor the number of cycles, facilitating this dynamic seed change. The objective is to achieve efficient power management in BIST while maintaining high fault detection capabilities. Different pattern generation methods are evaluated based on their power consumption. An mmm-bit binary counter and a Gray code generator are used in this processFault detection is carried out through signature analysis using a Multiple Input Signature Register (MISR), which identifies defects in the Circuit Under Test (CUT), specifically a Vedic multiplier. The design is simulated and implemented using Xilinx ISE and a Virtex 5 Field Programmable Gate Array (FPGA), with results provided for comparison.

Key Terms: 32-bit multiplier, test pattern generation, MISR, CUT, FPGA, BIST, low-power LFSR (LP-LFSR), switching activity.

1.INTRODUCTION

The rapid advancements in System-on-Chip (SoC) technology have made it possible to integrate a wide range of cores, from basic memory units to advanced Digital Signal Processors (DSPs), onto a single chip. However, as the number of processing elements within these chips continues to grow, managing communication between the various on-chip components has become increasingly challenging. To address these challenges, Network-on-Chip (NoC) architectures have emerged as a promising solution. NoC provides an efficient framework for data transmission across the chip while meeting key design objectives such as Power, Performance, and Area (PPA). By utilizing a network of routers and interconnects to connect processing elements, NoC overcomes the scalability and performance limitations of traditional bus-based communication systems.

The efficiency of a NoC is significantly influenced by the router, which plays a key role in controlling data exchange between network nodes. As on-chip networks grow in size and complexity, the router's design becomes increasingly critical. Several factors impact router performance, including the type of NoC architecture (such as Synchronous, Asynchronous, or Globally Asynchronous Locally Synchronous, GALS), the design of buffers, the structure of the arbiter, the choice of network topology, and the employed routing strategies. Optimizing these components can improve system performance and ensure seamless communication across cores.

In Synchronous NoCs, routers operate based on a global clock, which can result in increased power consumption. While these designs are typically compact and high-performing, they encounter challenges at higher operating frequencies, such as Electromagnetic Interference (EMI). To overcome the drawbacks of global clock distribution, approaches like Globally Asynchronous Locally Synchronous (GALS) systems have been developed. GALS divides the NoC into smaller synchronous regions, enabling each region to function independently without relying on a global clock. Additionally, Hybrid NoC designs have been investigated, combining Synchronous and Asynchronous communication to enhance energy efficiency and minimize latency. Some hybrid designs incorporate sub-routers for Synchronous control and Asynchronous data transfer to optimize performance.

Asynchronous NoCs, which operate without relying on a global clock, are known for their superior power efficiency. However, they generally have slower performance compared to Synchronous NoCs. Asynchronous designs are particularly well-suited for real-time applications that prioritize low power consumption and involve smaller data packets. In contrast, Synchronous NoCs are more appropriate for applications requiring continuous data transmission and handling large data packets, such as multimedia processing. Various Asynchronous NoC designs have been developed, including those based on Bundled Data Logic, which can achieve high throughput with relatively simple hardware but may be affected by timing variations. To improve latency, a two-phase clocked Mesh NoC using bonded Bundled Data Logic has also been introduced.

The selection of a routing protocol plays a crucial role in determining the performance of a NoC. While more advanced routing algorithms can enhance traffic management, they often increase router complexity, leading to higher power consumption and greater chip area requirements. In contrast, simpler routing protocols are typically more energy-efficient and cost-effective but may compromise traffic handling efficiency. Another important aspect of NoC design is the buffer size, which serves to store data packets and mitigate issues like packet loss or misrouting. However, larger buffers come with trade-offs, including higher power consumption and increased chip area. For example, input buffers in some NoC designs can occupy a substantial portion of the total network area, sometimes accounting for as much as 75%.

The router designs presented in this study have been carefully analyzed for their area efficiency and operating frequency. By incorporating distributed control mechanisms, these routers operate independently, eliminating the need for complex handshake protocols. This approach improves both scalability and operational efficiency. The architecture of the proposed routers, which includes Input Channels, a Crossbar Switch, and Output Channels, is crucial for efficient data routing and communication within the NoC. This design ensures seamless interaction between the various cores and routers in the system.

.

2.METHADOLOGY

The proposed a Linear Feedback Shift Register (LFSR) is utilized in this work because of its simplicity and efficient use of circuit area for generating test patterns. This paper introduces an innovative architecture designed to produce test patterns with minimized switching activity. The structure of the low-power test pattern generator (LP-TPG), as illustrated which incorporates a modified low-power LFSR (LP-LFSR), an \(m\)-bit counter, a Gray code generator, a NOR gate network, and an XOR array.

The m-bit counter is initialized to all zeros and produces a sequence of \(2^m \) test patterns. A gray code generator and the m-bit counter are both synchronized using a common clock signal (CLK). The counter's output is connected to a NOR gate and a gray code generator. When the counter's output is entirely zero, the NOR gate outputs a high signal. This high signal enables the clock to activate the LP-LFSR, which generates the next sequence. The output sequence from the LP-LFSR is then XORed with the sequence produced by the gray code generator. The resulting patterns from this XOR operation form the final output sequence.

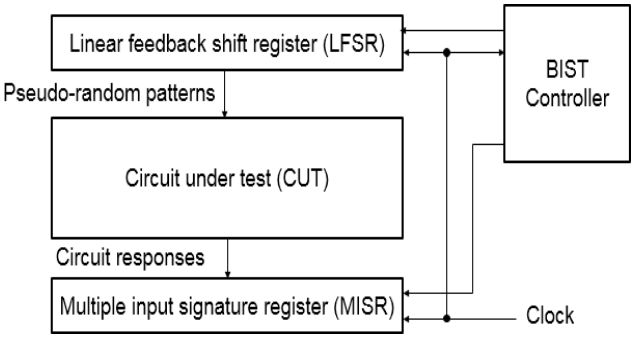

Figure.:1 BIST Architecture

The proposed method can be outlined in the following steps:

1. A pseudorandom test sequence is generated using an LFSR, and its single stuck-at fault coverage for the CUT is evaluated through both forward and reverse fault simulations. Let SSS represent the test sequence up to the last effective vector, beyond which the fault coverage shows no significant improvement.
2. Identify the set UUU of patterns in SSS that do not contribute to any fault detection.
3. For all ordered pairs of test vectors in the reduced set Sr=S\US_r = S \setminus USr=S\U, calculate the switching activity (SA) in the scan path and the CUT.
4. Rearrange the vectors in SrS_rSr to determine an optimal sequence order that minimizes energy consumption.
5. Modify the LFSR's state table so it generates the reordered sequence, denoted as S′S′S′.
6. Design a low-cost mapping logic (ML) to enhance the LFSR. This logic modifies the LFSR's state transitions under certain conditions to achieve two objectives: (a) to skip states that generate useless patterns and (b) to reorder SrS_rSr into S′S′S′. In all other conditions, the LFSR follows its original state transition function. Figure 2 illustrates a simple multiplexer-based design that can implement this logic. A similar concept of skipping LFSR states has been previously applied for embedding deterministic test sets.

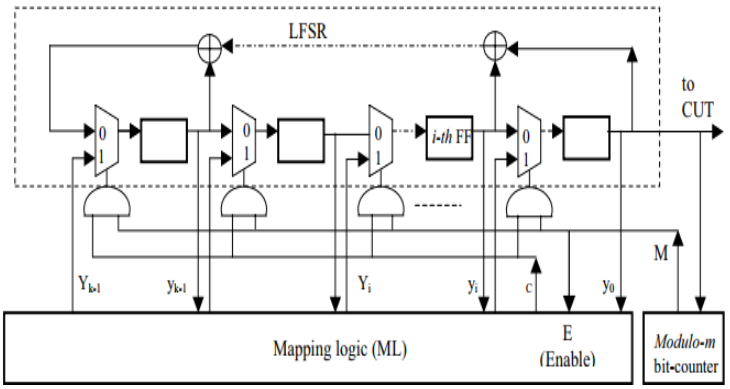

Figure:2 Example LFSR

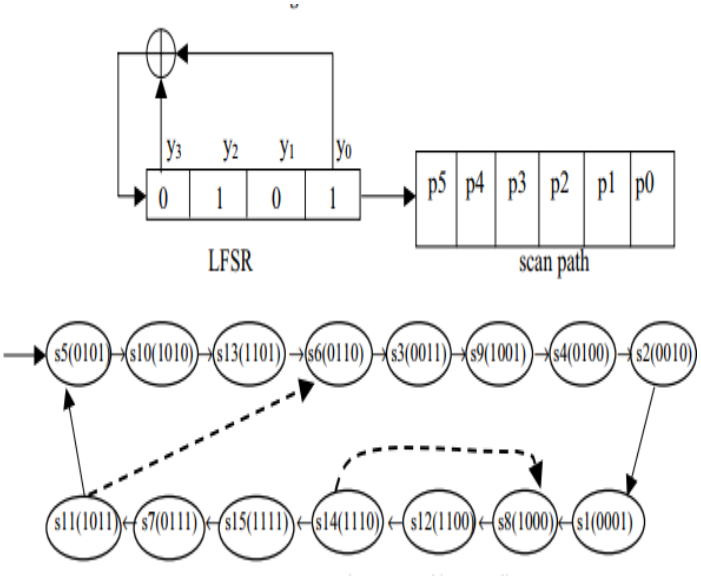

Figure: 3 State Diagram

Implementation:

Pattern Generator or LFSR:

An LFSR is an n-bit shift register that cycles through 2^n-1 values in a pseudo-random manner. It is built using standard flip-flops, with outputs from specific flip-flops being fed back (using modulo-2 arithmetic) to the register's inputs. Similar to a binary counter, it generates all 2^n-1 states, but in a reproducible "random" order. The combination of exclusive-OR gates and the shift register creates a pseudorandom binary sequence (PRBS) at the outputs of the flip-flops. By carefully selecting feedback points from the n-bit shift register, it is possible to generate a maximal-length PRBS of 2^n-1, which covers all possible nnn-bit patterns except the all-zero pattern. The all-zero state cannot occur in this type of LFSR because, if initialized with an all-zero seed, the LFSR remains in the all-zero state due to the feedback logic being based on XOR operations.

When used as a Test Pattern Generator (TG), an LFSR cycles rapidly through a large number of its states. These states, determined by the specific design parameters of the LFSR, serve as the test patterns. The concept of Pseudo-Random Pattern Generation is applied here. A sequence of 0s and 1s is referred to as a pseudo-random binary sequence when the bits exhibit randomness in a local context, yet the sequence is repeatable. This type of pattern generation typically requires less hardware, has lower performance overheads, and involves reduced design complexity compared to other methods. In pseudo-random test patterns, each bit is approximately equally likely to be a 0 or a 1.

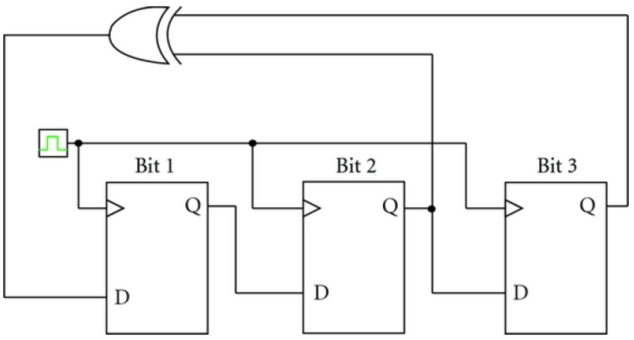

Figure:4 3-bit LFSR

Response Analayzer or MISR:

A Multiple Input Signature Register (MISR) is used for performing signature analysis in a system. This process is crucial because a properly functioning circuit and a faulty one will produce different signatures. These signatures are compared at the end to verify whether the CUT passes the test.

The Pseudo-Random Pattern Generator provides the required patterns, which are then input to the CUT. The CUT's response is fed into the MISR, which compresses multiple output streams into a single LFSR, thereby minimizing hardware requirements. There are various configurations for connecting the inputs of LFSRs to construct an MISR. Due to the linear and associative properties of XOR operations, such as (A⊕B)⊕C=A⊕(B⊕C) C = A (A⊕B)⊕C=A⊕(B⊕C), different configurations yield equivalent results as long as the outcomes of the additions remain the same. An nnn-bit MISR can process up to nnn inputs to generate a signature. The same polynomial equation used for LFSRs, such as x3+x2+1x^3 + x^2 + 1x3+x2+1, can also be utilized to design an MISR.

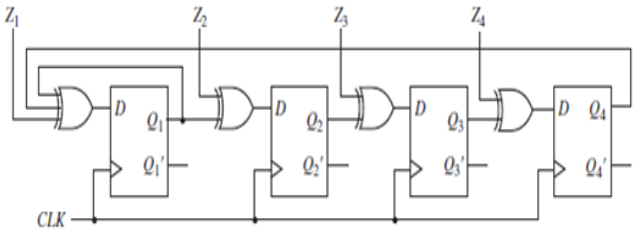

Figure :5 MISR with 3 input and 3 output bits

CIRCUIT UNDER TEST:

The Circuit Under Test refers to the test circuit being evaluated. Initially, a fully functional circuit is installed in the chip socket, located off-chip relative to the main chip that contains the LFSR, MISR, signature register, and BIST controller. When testing is required, the relevant circuitry is connected to the main chip and operated accordingly.

BIST CONTROLLER:

The BIST (Built-In Self-Test) Controller in managing LFSR (Linear Feedback Shift Register) and MISR (Multiple Input Signature Register) operations. By detailing how the controller pre-configures LFSRs and MISRs of various bit sizes and how it dynamically selects the appropriate configuration based on the number of inputs and outputs of the Circuit Under Test , it establishes the flexibility and adaptability of the system. This process ensures that the testing mechanism is tailored to the specific characteristics of the CUT, promoting efficiency and accuracy. Additionally, the explanation underscores the systematic and automated nature of the BIST process, which is crucial for optimizing test coverage and minimizing manual intervention.

SIGNATURE REGISTER:

The Signature Register is responsible for storing the signature values generated by the MISR. To determine whether the CUT is functional or faulty, the signature value of a known good circuit is compared to that of the CUT. The parent chip receives clock and reset signals, along with input specifying the required bit lengths for the Controller. Initially, a fault-free circuit is connected in place of the CUT. The system runs, and a signature value is generated for this circuit. Subsequently, another circuit of the same type—potentially faulty—is connected as the CUT, and the system runs again to produce a new signature value. If the signature values from the two circuits match, the CUT is deemed fault-free. If the values differ, it indicates a fault in the newly tested circuit.

CONCLUSIONS:

The designed BIST architecture is developed using Verilog and evaluated on various circuits with deliberately injected faults. The design is synthesized and simulated using Modelsim to evaluate its performance. A novel BIST approach is presented to conserve energy in both the LFSR and the CUT within a random testing framework. A significant portion of switching activity (SA) is inherent and cannot be minimized solely by vector reordering. To address this, alternative strategies include selecting a new set of effective test vectors from the random sequence or reconfiguring the scan path architecture. Another area of interest is ensuring the reusability of BIST hardware and mapping logic for multiple cores within a chip. This paper demonstrates a BIST implementation using Verilog, where the LFSR generates pseudorandom sequences, and signature analysis verifies circuit functionality.A discrepancy between the generated signature. Although there is a small chance of a faulty circuit producing the correct signature, using longer test sequences significantly enhances fault coverage

ACKNOWLEDGEMENT:

We would like to express our heartfelt gratitude to Mrs. Kavya M P, for their exceptional guidance, support, and expertise throughout this research project. We are deeply indebted to P E S institute of technology and management and its faculty members for providing us with the necessary resources and facilities that enabled us to conduct our study. We also appreciate the insightful discussions and feedback from our colleagues and peers, which significantly contributed to the advancement of this research. Lastly, we extend our sincere appreciation to our families and friends for their unwavering support and encouragement throughout this endeavor. Their love and motivation played a vital role in our succ

REFERENCES:

1. Chakrabarty and colleagues, “Generating Deterministic Built-in Test Patterns for High-Performance Circuits Using Twisted-Ring Counters,” IEEE Transactions on VLSI Systems, Volume 8, Issue 5, pages 633-636, October 2000.

2. Hakmi A. W., “Programmable Deterministic Built-In Self-Test,” presented at the IEEE International Test Conference (ITC), November 2007.

3. Katti R. S., Ruan X. Y., and Khattri H., “Design of Multiple Output Low-Power Linear Feedback Shift Registers,” IEEE Transactions on Circuits and Systems I, Volume 53, Issue 7, pages 1487-1495, July 2006.

4. Miron Abramovici, "Design for Testability: A Testability-Driven Approach," Revised Edition, November 1997.

5. Poornima M., "Implementation of a Multiplier Using the Vedic Algorithm,s" International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 2, No. 6, May 2013