**J**ournal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1: 2025
ISSN : **1906-9685**


# IMPLEMENTATION OF EFFICIENT AND LOW POWER TEST PATTERN GENERATORS

**[1]Dr.S.Selvakumar Raja, [2]Mr.M.Mahipal**

[1]Principal & Professor Department of ECE, Kakatiya Institute of technology and science for women, Manikbhandar, Nizamabad-503003, Telangana State, India

HOD, Department of ECE, Kakatiya Institute of technology and science for women, Manikbhandar, Nizamabad-503003, Telangana State, India

***ABSTRACT***

We are aware that during testing, the gadget dissipates around 200% more power than it does in regular working mode when it is not in normal functioning mode.A 32-bit test pattern generator has been proposed for this project to validate the VLSI design. This 32-bit test pattern generator consumes less power due to its efficient LFSR and extra combinational circuitry. Low power consumption is due to decreased switching activity between test vectors.

**Key words:** LFSR, low power VLSI architecture,The Bipartite LFSR designand Verilog HDL.

## 1. INTRODUCTION

In modern integrated circuit (IC) design and development, ensuring the reliability, functionality, and performance of digital systems is a crucial task. With the increasing complexity of Very Large-Scale Integration (VLSI) circuits, testing has become a fundamental aspect of the design cycle to detect faults, ensure correctness, and improve yield. One of the essential components of digital testing is the generation of test patterns, which are used to stimulate the circuit under test (CUT) and observe its response to identify any defects. The efficiency of this process significantly impacts the overall performance, power consumption, and reliability of the testing environment. Traditional methods of test pattern generation (TPG) often focus on maximizing fault coverage but tend to overlook key constraints such as power consumption, hardware overhead, and speed of operation.

This is particularly critical for contemporary VLSI designs, where low power consumption and energy efficiency are paramount considerations due to the rise of portable and battery-operated devices, as the scaling down of technology nodes. Additionally, as IC complexity grows, the number of patterns required to test a circuit also increases, leading to greater power dissipation during testing, which can affect the lifespan of the device or even cause unintended functional behavior during testing. This project, "Implementation of Efficient and Low Power Test Pattern Generators," addresses the challenges associated with traditional test pattern generation methods by proposing novel techniques that not only ensure high fault coverage but also emphasize efficiency in power consumption and resource utilization. The primary focus is to develop test pattern generators that are optimized for low power operation while maintaining or enhancing fault detection capabilities. By implementing these efficient TPGs, we aim to minimize the dynamic power dissipation during testing, reduce switching activity, and maintain the integrity of the testing process without compromising on fault coverage.

## 2. LITERATURE SURVEY

Numerous research initiatives aimed at lowering power consumption, increasing fault coverage, and boosting overall efficiency have been prompted by the difficulty of optimising test pattern generators (TPGs) for VLSI circuits. The goal of this project is to address the current trends and issues in the development of low-power and efficient TPGs, which may be understood thanks to these studies. Girard et al. (2000) – "Low-Power Testing: A Survey" One of the first thorough analyses of low-power testing techniques was given by Girard et al., who described the causes of power consumption during test procedures, especially the excessive switching activity brought on by conventional test pattern generators such as LFSRs. Their research highlighted the need of lowering testing power dissipation, particularly in bigger circuits. The poll did not, however, provide any suggestions for reducing TPG power use, indicating a need for further study in this field.

Wang et al. (2001) – "Low-Transition Test Pattern Generation" Low-Transition Linear Feedback Shift Registers (LT-LFSRs), which minimised switching activity during test pattern application, were adopted in this work and decreased power consumption. By decreasing the number of transitions between successive test vectors, Wang et al.'s method significantly reduced dynamic power compared to conventional LFSRs. The authors' failure to investigate the influence on fault coverage in more intricate fault models, such as bridging or transition failures, restricts the overall application in wider testing settings, even if LT-LFSRs are successful at lowering power.

## 3. EXISTING SYSTEM:

To create test patterns for digital circuit testing, the current approach for test pattern creation in Built-In Self-Test (BIST) implementations usually makes use of traditional methods and algorithms. These approaches often produce test vectors

within the digital circuit itself using deterministic algorithms or pseudorandom techniques. Although BIST systems have gained popularity due to their capacity for self-testing and their potential to eliminate the need for external test equipment, their efficiency and power consumption may be limited. Low power consumption and energy efficiency may not be given priority by conventional BIST-based test pattern generators, which might result in higher power dissipation during testing activities.Additionally, it's possible that the current BIST systems aren't optimised to minimise switching activity and lower power consumption while creating test patterns. This may not be appropriate for battery-operated devices or power-sensitive applications, and it may lead to less than ideal energy efficiency performance.

Furthermore, in order to obtain thorough test coverage, traditional BIST-based test pattern generators could depend on laborious testing techniques, which can be costly in terms of both time and resources. For applications where quick testing and low power consumption are crucial factors, this strategy may not be the best option.Overall, BIST-based test pattern generators may not completely meet the needs for efficiency and low power consumption in contemporary digital circuit testing, even if they provide self-testing capabilities and less need for external test equipment. The energy economy and sustainability of test pattern generating methods in BIST implementations need improvements and optimisations.

**Basics of LFSR(Linear Feedback Shift Register)**

One kind of digital circuit that produces a series of binary integers is called a Linear Feedback Shift Register (LFSR). In order to create the next bit in the sequence, the contents of a register—a group of flip-flops that hold binary values—are shifted, and the feedback from some of the bits is used.Among other uses, LFSRs are often used in digital signal processing, encryption, and error correction coding. The fundamentals of LFSRs, including their composition, functionality, and characteristics, will be covered in this article.An LFSR's structure A register of N flip-flops coupled in a linear feedback loop makes up an LFSR. A binary sequence known as the seed value is placed into the flip-flops to initialise the register. A series of taps, which are chosen bits from the register and utilised to create the subsequent bit in the sequence, make up the feedback loop. An XOR gate, which is linked to the taps, uses an exclusive-or (XOR) operation to combine the values of the chosen bits. The flip-flops' values move one place to the right when the XOR gate's output is sent back into the register. The output of the XOR gate becomes the next bit in the sequence and replaces the rightmost bit in the register. An LFSR's functioning is significantly influenced by the tap selection. Selecting taps appropriately allows the feedback loop to create a maximum-length sequence with a period of 2N - 1, where N is the number of flip-flops in the register. Maximum length sequences should contain all possible 1s and 0s exactly once.

It also has high statistical qualities that make it ideal for a variety of applications.

The way an LFSR operates A mathematical equation that specifies the sequence produced by the register may be used to explain how an LFSR operates. Using the register's rightmost bit, x(t), x(t-1),..., x(t-N+1) indicate the flip-flop values at time t. Next, the equation gives the next bit:

$$x(t+1) = x(t-N+1) \oplus x(t-N+2) \oplus \ldots \oplus x(t) \oplus f(x(t),$$

where f is a Boolean function that defines feedback loop taps and XOR. The values of the chosen bits in the register are sent into the Boolean function, which outputs a single binary value that serves as feedback. A binary polynomial, or polynomial over the binary field GF(2) with coefficients of 0 or 1, may be used to represent the value of f. The number of taps in the feedback loop equals the polynomial's degree. For instance, an LFSR with three taps, with taps at locations 0, 1, and 3 in the register, is represented by the polynomial $f(x) = x^3 + x + 1$. An LFSR's characteristics LFSRs are helpful in a variety of applications due to a number of intriguing features. One of their most significant characteristics is that, although being deterministic, they produce a sequence that seems random. This characteristic results from the feedback loop's generation of a maximum length sequence, which possesses long period and flat power spectrum, among other favourable statistical characteristics. The ease with which LFSRs may be implemented in software or hardware using basic logic gates and shift registers is another crucial feature. This makes them a desirable option for applications where simplicity, speed, and low power consumption are crucial considerations.In cryptography, LFSRs are also used to generate pseudorandom numbers that serve as encryption and decryption keys.

The cryptosystem's security depends on the unpredictability of the sequence produced by an LFSR, and many methods have been devised to verify that LFSR sequences are random.TheBerlekamp-Massey algorithm is one such method for figuring out the shortest feedback polynomial that can produce a certain binary number sequence. By contrasting the feedback polynomial with the predicted polynomial for a maximum length sequence, this approach may be used to determine if an LFSR sequence is random.The linear complexity test is an additional method that counts the number of linearly independent subsequences of a certain length in the LFSR sequence. This test may be used to estimate the LFSR's period and identify any non-randomness in the sequence.

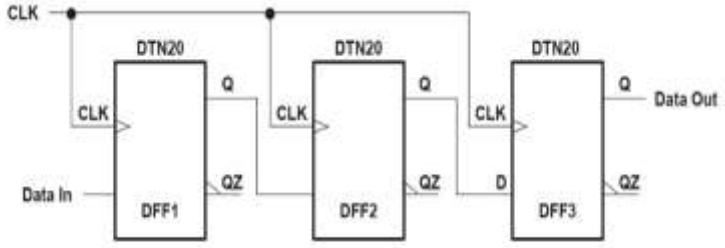



Figure 1. A 3-Bit Shift Register

**Pseudorandom Pattern Generation**

Linear feedback shift registers generate pseudorandom patterns well. When the flipflops' outputs are loaded with a seed value, the LFSR creates a pseudorandom pattern of 1s and 0s.
Keep in mind that the clock is the sole signal required to produce the test patterns.

**Maximal-Length LFSRs**

A maximal-length LFSR creates the most PRPG patterns with a pattern count of 2n – 1, where n is the number of register elements. It creates about equal numbers of 1s and 0s in patterns and runs.Designers may utilise Peterson and Weldon2's lists of maximum-length LFSRs because it's hard to predict their length. Table 1 shows the LFSR patterns in Figure 2 assuming 111 was the seed.

The user would build an LFSR greater than three bits in a practical ASIC design to generate several pseudorandom patterns before they repeated. Practical constraints restrict the LFSR's duration. A 32-bit maximal-length LFSR at 16 MHz may create almost 4 billion patterns in five minutes.

## 4. PROPOSED SYSTEM

The low-power LT-LFSR test pattern generator in this work can test sequential and combinatorial circuits. By enhancing correlation between test vectors, the recommended design reduces transitions. Power consumption may be reduced by reducing test vector switching. The typical LFSR architecture must be modified to incorporate intermediate patterns between its unique pairs of patterns. This is possible with a little toggling between two test vectors and two methods, random injection and bipartite injection, which are detailed in this section.

Our Low Transition Test Pattern Generator uses Random Injection (RI) and Bipartite LFSR to create test vectors. The RI technique employs a random-bit injection (R), which may be "0" or "1," to introduce a new pattern in the next bit of an intermediate pattern after a pattern pair transition.

**Low-Transition Pattern Generation Techniques**

Use the random injection strategy to introduce a new test vector, Ti1, between two test vectors such that the total switching activity between Ti and Ti1 and Ti1 and Ti+1 equals Ti and Ti+1. Introducing the Ti1 bit pattern between Ti and Ti+1 reduces switching activity.Injection occurs if two equal-bit places between Ti and Ti+1 are the same. As shown in fig., the RI injection is taking place when a transition takes place between Ti and Ti+1.

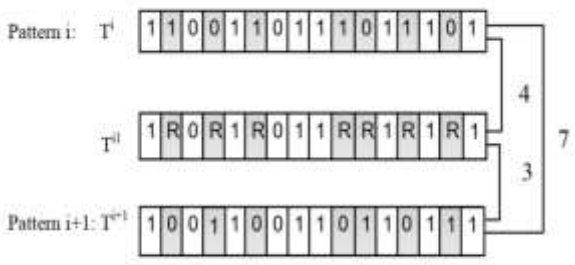
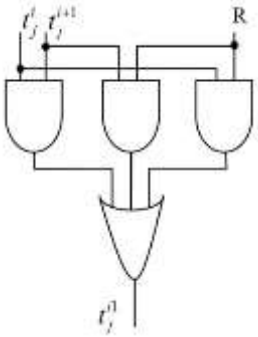



Fig. 2: Example of RI injection (R=0)




An illustration of the creation of an intermediate design may be seen in the above figure. The number of transitions (1<=>0) between the Ti and Ti+1 bit patterns prior to injection is seven, The darkened bit shows. After injecting the Ti1 bit pattern with R=0 or R=1, the transitions are decreased to four or three. We considered four transitions the worst-case situation. Figure shows the random injection circuit.

**Bipartite LFSR Technique**

LFSR realisation may be altered during testing to improve design performance like dissipated power. However, changing the patterns' arrangement or adding new ones might influence unpredictability. An intermediate test pattern between two successive random patterns lowers transitions. Two nonoverlapping Clk (clock) signals partition an LFSR into two equal halves. Thus, half of LFSR is idle and half is running. Fig. depicts an LFSR with several flipflops in sequence with a clock signal. Figure shows the Bipartite LFSR's basic construction for creating pattern Ti1.

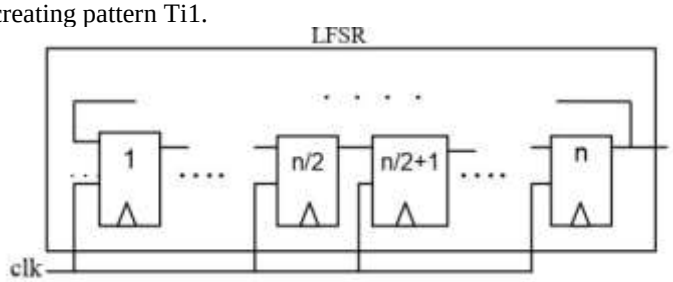



Fig2: LFSR

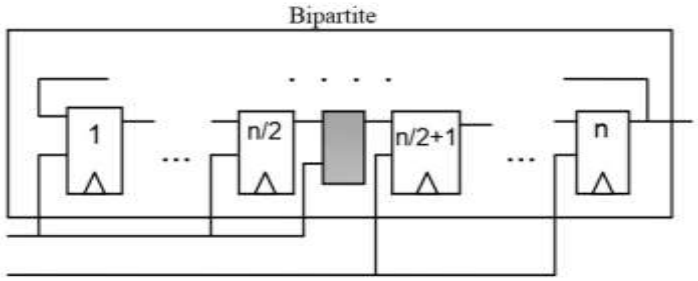



Fig3 :The Bipartite LFSR design

The Bipartite LFSR design stores the n/2th bit using a fake flip-flop when clk1clk2 = 10. The (n/2+1) flip-flop receives this value when the second half begins (clk1clk2 = 01). The effective partitioning of LFSR into two equal halves yields the sheltered flipflop.

An n-bit LFSR is split into two n/2-bit ones to save testing and clock tree power. By dividing the LFSR into two, it loses its randomisation capacity and creates and distributes two nonoverlapping clock signals (at half frequency), increasing area overhead.

Table 1. Pattern-Generator Seed Values

| CLOCK PULSE | FF1_OUT | FF2_OUT | FF3_OUT | COMMENTS |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | Seed value |
| 2 | 0 | 1 | 1 | |
| 3 | 0 | 0 | 1 | |
| 4 | 1 | 0 | 0 | |
| 5 | 0 | 1 | 0 | |
| 6 | 1 | 0 | 1 | |
| 7 | 1 | 1 | 0 | |
| 8 | 1 | 1 | 1 | Starts repeat |

**Implementation Low-Transition Linear Feedback Shift Register Architecture:**

Our approaches develop two test pattern generating schemes (RI and Bipartite LFSR) using the least amount of power. We merge these two strategies with LFSR structure to generate LT-LFSR, which decreases average power compared to Bipartite LFSR alternatives. Due to the unpredictability of injected patterns, most intermediate patterns may be employed for error detection in addition to typical LFSR figures.A 32-bit LT-LFSR structure with bipartite LFSR and random injection is shown in Fig. An external-XOR LFSR is employed in LTLFSR. Figure shows an injector circuit that gets its lower input from the

bottom bipartite lfsr bits and its upper input from the top output. Non-overlapping clock signals clk1 and clk2 form patterns by selecting half of the Bipartite LFSR (Fig.). The significance of Mux circuitry is selecting a bit from the injector bit or bipartite lfsr. Using overlapping clock signals, the FSA generates the control signals clk1 and clk2. The RI injection circuit's upper or lower bit is selected by sel1 and sel2.
A Finite-State Automation (FSA) generates the test pattern generation controls signal:
1.During the first clock cycle, sel1sel2 = 11 and clk1clk2 = 10. The top of the bipartite LFSR is active, while the bottom is dormant.Sel1sel2 = 11 transmits both sides of the bipartite LFSR to Out. Assume Ti is generated.
2. Second clock cycle: sel1sel2 = 10 and clk1clk2 = 00. Bipartite LFSR halves are inactive. Out [31:16] receives the top half of the bipartite LFSR, whereas Out [15:0] receives the RI injector circuit outputs. This stage produces T i1.
3. Sel1sel2 = 11 and clk1clk2 = 01 in the third clock cycle. While the top of the bipartite LFSR is inactive, the bottom is active. Out [31:0] connects to Bipartite LFSR output.This produces Ti2.
4. The fourth clock cycle has sel1sel2 = 01 and clk1clk2 = 00. Each half of the bipartite LFSR is idle. When sel1sel2 =01, the injector outputs and the bottom half of the bipartite LFSR are delivered to the outputs (Out [15:0] and Out [31:16]). This produces Ti3.
5. Step 1 during the fifth clock cycle generates T i+1.

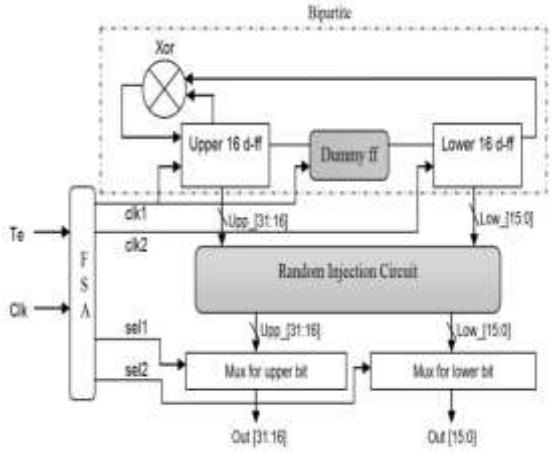



Fig3 :design includes the 16-bit upper and lower D-type flip-flops

The Figure 3 design includes the 16-bit upper and lower D-type flip-flops. The false flip-flop stores the last bit generated by the upper flip-flop throughout the clock cycle. A unique pattern is created via XORing. The non-overlapping enabling signal and Mux select line need four clock cycles. choice signal 1 selects the higher bit Mux, whereas choice signal 2 selects the lower bit Mux.

## 5. RESULTS

### SIMULATION RESULTS

While the schematic is the verification of the verification of the connections and blocks, the simulation is the procedure that is referred to as the ultimate verification with regard to its operation. The simulation window, which limits the output in the form of waveforms, is opened by switching from implantation to the simulation on the tool's home screen.

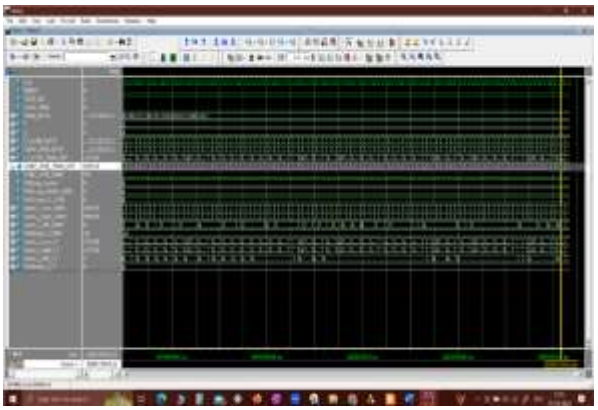


Fig4: simulated wave form1

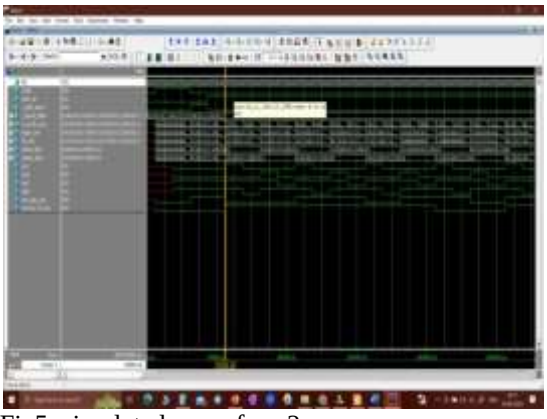


Fig5: simulated wave form2

**Schematic:**

The register transfer level is shortened to schematic. It indicates the architecture's blueprint and is used to compare the planned architecture to the ideal architecture that still has to be developed.

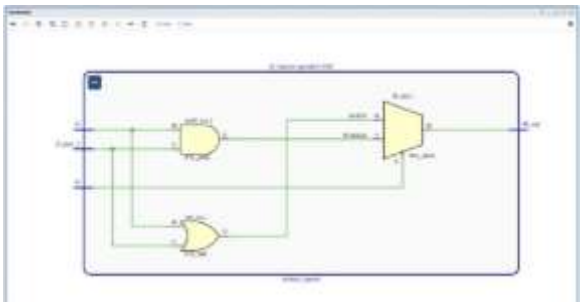


**Fig 6: Schematic_Random_Injection**

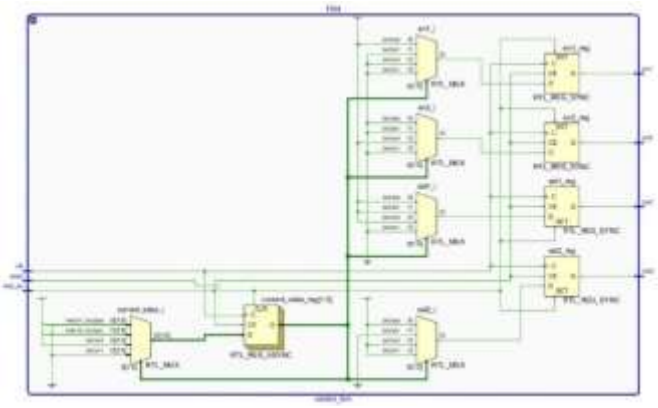


**Fig7: Schematic_Control_FSM**

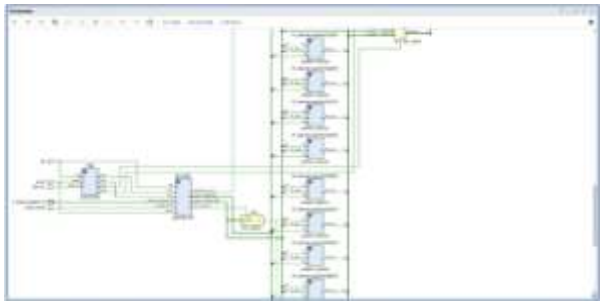


**Fig 8: Schematic_LP_LFSR**

**AREA REPORT**

| Site Type | Used | Fixed | Available | Util% |
|---|---|---|---|---|
| Slice LUTs* | 132 | 0 | 303600 | 0.04 |
| LUT as Logic | 132 | 0 | 303600 | 0.04 |
| LUT as Memory | 0 | 0 | 130800 | 0.00 |
| Slice Registers | 103 | 0 | 607200 | 0.02 |
| Register as Flip Flop | 71 | 0 | 607200 | 0.01 |
| Register as Latch | 32 | 0 | 607200 | <0.01 |
| F7 Muxes | 0 | 0 | 151800 | 0.00 |
| F8 Muxes | 0 | 0 | 75900 | 0.00 |

**TABLE 3. UTILIZATION_REPORT_LP_LFSR**

| Ref Name | Used | Functional Category |
|---|---|---|
| LUT3 | 96 | LUT |
| IBUF | 36 | IO |
| FDCE | 35 | Flop & Latch |
| OBUF | 32 | IO |
| LUT6 | 32 | LUT |
| LDCE | 32 | Flop & Latch |
| FDPE | 32 | Flop & Latch |
| LUT2 | 6 | LUT |
| BUFG | 3 | Clock |
| FDSE | 2 | Flop & Latch |
| FDRE | 2 | Flop & Latch |
| LUT1 | 1 | LUT |

**TABLE 4.PRIMITIVES_USED_LP_LFSR**

**POWER REPORT**

| | |
|---|---|
| LT_LFSR_DATA = 6346efdf | LT_LFSR_TRAN_CNT 127560 |
| LT_LFSR_DATA = 6346a5cb | LT_LFSR_TRAN_CNT 127565 |

**TABLE 6. TRANS_COUNT_PROPOSED_LFSR**

| | |
|---|---|
| CONV_LFSR_DATA = 7022a797 | CONV_LFSR_TRAN_CNT 525432 |
| CONV_LFSR_DATA = e0454f2f | CONV_LFSR_TRAN_CNT 525440 |

**TABLE 7. TRANS_COUNT_CONVENTIONAL_LFSR**

## CONCLUSION

Using the Low Power LFSR technique, this project shows an effective HDL implementation of low power utilisation for the test pattern generator. It also presents a hypothesis that explains how a test pattern is generated using the Low Transition Linear Feedback Shift Register design. This approach may use less electricity than typical LFSR methodologies. It shows that the low transition linear feedback shift register consumes 50.06% less power than the regular LFSR. The findings demonstrate how beneficial Low Power LFSR is for power reduction strategies in testing mode.In electrical hardware design, the BIST (Built-In Self-Test) approach is used to test integrated circuits (ICs) without the need for external test equipment. Because it provides a practical and affordable means of evaluating integrated circuits (ICs) during production and in-field testing, BIST is growing in popularity. LPTPGs, or low power test pattern generators, are a crucial part of BIST. For power integrity testing, LPTPGs provide test patterns that aid in identifying issues in an IC's power distribution network.

Since low-power design methods are increasingly being used in ICs, the potential applications of LPTPGs in BIST seem bright. Low-power consumption is becoming a crucial component of IC design due to the growing use of mobile and Internet of Things devices. LPTPGs are a crucial tool for BIST as they may assist in identifying power-related issues in low-power integrated circuits. Furthermore, as ICs continue to become more complicated and need more thorough testing, the demand for LPTPGs is anticipated to increase over the next few years. The need for dependable, high-performing, and energy-efficient integrated circuits (ICs) for a variety of applications, such as consumer electronics, healthcare, and automotive, is driving this trend. In conclusion, LPTPGs have a promising future in BIST and will probably play a bigger role in IC testing as the need for high-performance, low-power integrated circuits keeps rising.

***REFERENCES***


1. A. M. A. Rahim, S. M. A. Motakabber, M. M. A. Hashem, M. R. Islam, "A low power and high performance built-in-self-test architecture for embedded memories," in 2016 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2016, pp. 1-5.
2. Y. Li, M. M. Hasan, H. Zhao, L. Chen, and K. Roy, "Energy-efficient test pattern generation using charge sharing in transition fault testing," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 26, no. 7, pp. 1372-1382, July 2018.
3. S. Devadoss, V. Srinivasan, and S. B. Venkatesh, "A novel low-power test pattern generator for embedded memories," Journal of Low Power Electronics, vol. 9, no. 4, pp. 383-392, Dec. 2013.
4. N. E. Zaki and M. H. El-Halawany, "Efficient test pattern generation technique for combinational circuits," in 2015 28th Canadian Conference on Electrical and Computer Engineering (CCECE), 2015, pp. 703-706.
5. S. N. Nashed, N. E. Zaki, and M. H. El-Halawany, "An efficient test pattern generator for sequential circuits based on selective gate manipulation," in 2016 29th Canadian Conference on Electrical and Computer Engineering (CCECE), 2016, pp. 1-4.
6. A. S. Al-Sabbagh and A. R. Al-Ali, "A low-power BIST for sequential circuits based on clock gating and scan chain," Journal of Circuits, Systems and Computers, vol. 24, no. 7, pp. 1550115-1-1550115-16, Jul. 2015.
7. P. Gao, K. Xu, H. Liu, and Q. Luo, "A low-power test pattern generator based on shared random access memory for scan-based BIST," in 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), 2018, pp. 954-957.

8. S. K. Singh and S. M. Khairnar, "Low power BIST design for combinational circuits using 4-phase clock," in 2018 International Conference on Intelligent Computing and Control Systems (ICCS), 2018, pp. 211-214.
9. S. V. Aradhya and B. S. Sathyanarayana, "A low power BIST architecture for combinational circuits using efficient shift register design," in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1390-1395.
10. D. K. Kim, K. S. Kim, S. H. Kim, J. S. Kwak, and K. C. Kim, "A low-power test pattern generator for multiple-port embedded memories," in 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019, pp. 1-5.