

Linear Feedback Shift Register-Based TestPattern Generators: A Comparative Study

Shubham Mishra¹, Satyarth Tiwari²

M. Tech. Scholar, Dept. of Electronics and Communication, RKDFCE, Bhabha University¹

Assistant Professor, Dept. of Electronics and Communication, RKDFCE, Bhabha University²

Abstract— Pseudo Random Number Generators are largely used in VLSI Architecture as Test Pattern Generators for testing of digital circuits in a BIST system. The test pattern sequence generated also finds applications in cryptography. It is thus require designing an effective test pattern generator which utilizes least hardware, dissipates lowest power and generates most random sequence. This paper attempts to compare a very few recent proposals where Linear Feedback Shift Register (LFSR) has been suitably modified to obtain an efficient architecture of a test pattern generator.

Index Terms—Test Pattern Generator (TPG), Linear Feedback Shift Register (LFSR), Testing of Digital Circuits, Switching Activity, Random Sequence

I. INTRODUCTION

In the field of VLSI Design, Digital Circuits offer a number of advantages in terms of speed, area and power consumption over Analog Circuits. A lot of design methodologies and architectures are proposed in literature to improve these parameters. While improving circuit design parameters, it is also essential to increase the testability, i.e. the ability of fault detection and rectification of the circuit. Built-In Self-Test (BIST) is a such Design for Testability Technique aimed to improve testability of the Circuit Under Test (CUT), while ensuring low power dissipation. A typical BIST^[1] architecture has four elements - Test Pattern Generator, Circuit Under Test, Output Response Analyzer and BIST Controller.

The Test Pattern Generator (TPG) produces test patterns that are applied as inputs to the CUT. Generally, Pseudo Random Number Generators (PRNGs) based on Linear Feedback Shift Register (LFSR) architecture are used for this purpose. Power dissipation becomes a major concern while applying these test patterns due to the high switching activity between consecutive patterns. As the power dissipation increases, the current drawn increases which might lead to damage of the digital circuit.

Previous works have suggested a number of architectures for TPGs. While some^[2] have modified the existing LFSR with a combinational element, while some^{[3][4]} have implemented the physical design with a different logic family. This paper attempts to do a comparative study on the former proposals,

where LFSR has been modified or updated to get an improved Test Pattern Generator.

The parameters considered while comparing the architectures are hardware utilization, i.e. no. of Flip Flops (FFs) and Look-Up Tables (LUTs), and measure of randomness of the test patterns.

II. LFSR ARCHITECTURE

A Linear Feedback Shift Register is a sequential circuit consisting of D Flip Flops and exclusive OR gates, arranged linearly. For a N-bit LFSR, N is referred to as the degree of the characteristic polynomial that defines its construction. The degree of the polynomial gives the number of Flip Flops, and the number of non-zero coefficients (excluding x_n and x_0) gives the number of XOR gates. The characteristic equation is of the form:

$$\varphi(x) = \varphi_n x^n + \varphi_{n-1} x^{n-1} + \dots + \varphi_1 x + \varphi_0$$

A. Type -1 / Modular LFSR

In an internal-XOR LFSR, the output of the last flip-flop in the shift register, D_{n-1} , is fed back to the inputs of a selected subset of flip-flops. In the characteristic equation, for $i = 0, 1, \dots, n-1$, $\varphi_i = 1$ indicates that feedback occurs at the input of D_i , and $\varphi_i = 0$ indicates that no feedback occurs there.

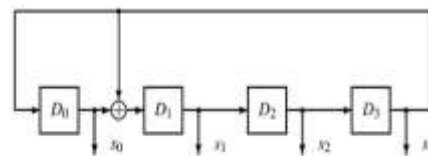


Figure 1 - Type - 1 LFSR

B. Type - 2 / External LFSR

In an external-XOR LFSR, the outputs of a selected subset of flip-flops are combined into a single signal, which is fed back to the input of the first flip-flop in the shift register, D_{n-1} . The fact that the signal is fed back to the input of D_{n-1} (which can be viewed as being driven by non-existent D_n) is indicated by always assigning $\varphi_n = 1$. For $i = 0, 1, \dots, n-1$, $\varphi_i = 1$ indicates that the output of D_i is selected for feedback. Once again, $\varphi_0 = \varphi_n = 1$ for a non-trivial n-stage external-XOR LFSR.

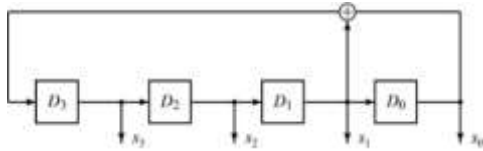


Figure 2 - Type - 2 LFSR

III. COMPARATIVE STUDY

A. Low Power Test Pattern Generator [5],[6]

This algorithm of test pattern generation aims to increase the correlation between test patterns to reduce the switching activity between primary inputs (PIs), which eventually scale down the power utilisation. This architecture consists of an LFSR with appended combinational logic to produce intermediary patterns with fewer transitions compared to the next one.

The algorithm consists of a Finite State Machine with 4 states to drive the test pattern generator. The LFSR is divided into two halves with an intermediary state-capturing Flip Flop in between. The half-LFSRs and output MUXs are enabled or disabled using control signals.

To begin with, the first half of the LFSR performs the shift operation, while the second half is disabled. The output is given by the LFSR output. Next, both the halves are idle, and the MUX performs the selective bit complement operation based on previous value of the output. The output comprises DFF outputs of the first half and bit-complemented output of the second.

Further, the first half of LFSR is disabled, while the second half performs the shift operation giving the output. At last, again both the halves are idle, and the MUX performs selective bit-complement based on previous value of the output. The output comprises bit-complemented results of the first half and DFF outputs of the second half.

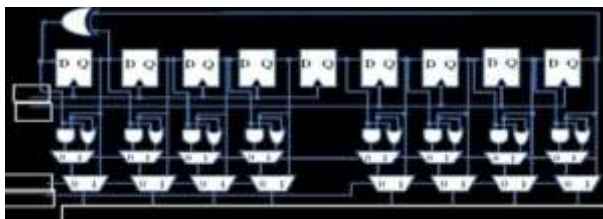


Figure 3 -Low Power LFSR [6]

B. Bit Swapping LFSR [7],[8]

This architecture aims to reduce the switching activity between consecutive outputs of the LFSR that serve as input to the CUT. The number of bit transitions in the subsequent outputs when minimized results in reduction of power as well. The LFSR is initiated with a non-zero seed polynomial. The modification to a conventional LFSR is that one of the output lines (say N^{th} , for an N-bit LFSR) serves as a selection line for swapping the remaining bits. If that bit is 1 the bits are swapped, otherwise not. The bit swapping takes place in

such a fashion that the 1st bit is swapped with 2nd bit, 3rd bit with 4th bit, and goes on till (n-2)th and (n-1)th bit (if n is odd), and till (n-3)th and (n-2)th bit (if n is even).

This bit swapping technique reduces the number of transitions between the subsequent patterns generated, and hence would result in lower power consumption during the testing phase of the CUT.

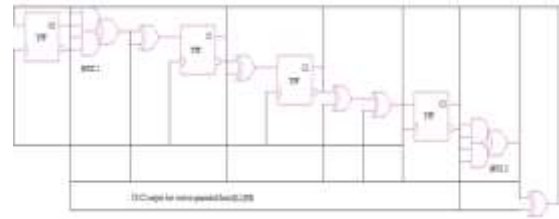


Figure 4 - Bit Swapping LFSR [7]

C. A-periodic LFSR [9]

This architecture aims to introduce randomness among the patterns generated, and drift away from a fixed period after which the output of the test pattern generator starts repeating. It introduces control units for Feedback configuration Algorithm, Seed Configuration Algorithm and Range Configuration Algorithm to a traditional LFSR.

Feedback Configuration Algorithm:

The fixed feedback paths are broken by a control input, which is ANDed with every feedback path. If the value of this input is 0, feedback is disabled while on 1, it is enabled.

Seed Configuration Algorithm:

A random seed value is inserted at random intervals in the circuit. The seed comes from the samples of meta-stable statesignals, which are almost unpredictable. For simulation purposes, the random function of Verilog was employed.

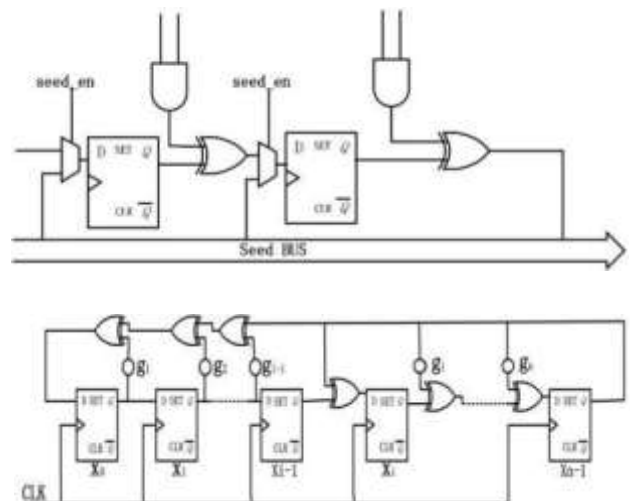


Figure 5 - Aperiodic LFSR [9]

IV. METHODOLOGY

All the LFSR architectures described in the previous two sections were designed in Verilog HDL. The circuit elements like 2-input logic gates, 2:1 Multiplexers, D Flip Flops etc. were implemented from standard Verilog libraries to

maintain consistency. As a reference for uniformity, the following characteristic polynomial was chosen for all the LFSRs:

$$f(x) = 1 + x^4 + x^6 + x^{11} + x^{13} + x^{15}$$

The Verilog modules were simulated in Xilinx Vivado Design Suite 2018.3. The Elaborated Design (RTL) of each architecture has been attached in the above sections, where they were described. The Test Pattern Generators were tested against a common testbench with a 16-bit Ripple Carry Adder as the Circuit Under Test.

The 16-bit outputs of LFSR were tested for pseudo-randomness via NIST Statistical Test Suite^[10]. The LSB's of every test vector were appended to form a single 100-bit sequence. The Frequency Monobit Test was performed in MATLAB, through a script.

V. SIMULATION RESULTS

Table 1: Comparison of LFSR architectures

Architecture	Utilisation		
	FFs	LUTs	I/O
Conventional LFSR	32	23	19
Low Power LFSR	52	74	19
Bit Swapping LFSR	32	47	19
Aperiodic LFSR	32	39	21

Table 1 clearly shows that a conventional LFSR is the most efficient to implement, in terms of hardware utilisation. This is because no combinational circuit has been added to modify its functionality. While a Low Power LFSR having suitable modifications uses the most hardware.

Table 2: NIST Test Suite Results

Architecture	p-value	Type of sequence
Conventional LFSR	0.1615	Random
Low Power LFSR	0.6153	Random
Bit Swapping LFSR	0.5485	Random
Aperiodic LFSR	0.8132	Random

The Frequency Monobit Test of NIST Statistical Test Suite^[10] states that a p value close to 1 indicates a truly random sequence and that close to 0 indicates non-randomness of a sequence. A p-value > 0.01 is generally considered as a random sequence. From Table 2, it can be stated that the Aperiodic LFSR generates the most random test sequence, while a conventional LFSR generates the least random sequence.

VI. CONCLUSION

This paper presented a comparative analysis of three modern LFSR-based Test Pattern Generators. While the design with the least hardware throughput, generates a test sequence which is less random than the one with a higher hardware utilisation. A low power design, as the name suggests, dissipates lowest power but is unable to generate a truly random sequence. Other designs that require fewer hardware

components and generate truly random sequences can be studied and implemented in the future.

REFERENCES

- [1] Babu, S. L., & Sarma, N. S. M. (2018). Automatic test pattern generation in VLSI - A survey. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, 3012-3017.
- [2] Saraswathi, T., Ragini, K., & Ganapathy Reddy, C. (2011). A review on power optimization of linear feedback shift register (LFSR) for Low power built in self test (BIST). ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology, 6, 172-176.
- [3] Oommen, R., George, M. K., & Joseph, S. (2018). Study and Analysis of Various LFSR Architectures. 2018 International Conference on Circuits and Systems in Digital Enterprise Technology, ICCSDET 2018, 1-6.
- [4] Babu, S. L., & Sarma, N. S. M. (2018). Automatic test pattern generation in VLSI - A survey. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, 3012-3017.
- [5] Thubrikar, T., Kakde, S., Gaidhani, S., Kamble, S., & Shah, N. (2018). Design and implementation of low power test pattern generator using low transitions LFSR. Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017, 2018-January, 467-471.
- [6] Saifuddin, K., Reddy, D. & Raviteja, C. (2019) Modified design of test pattern generator for built-in self test applications. International Journal for Innovative Engineering Management and Research, IJIEMR, Vol. 08, Issue 05, May 2019
- [7] Dhanesh, P., & Jayanth Balaji, A. (2015). Dual threshold bit-swapping LFSR for power reduction in BIST. ICACCS 2015 - Proceedings of the 2nd International Conference on Advanced Computing and Communication Systems, 5-7.
- [8] Abu-Issa, A.S., Quigley, S.F., Bit-swapping LFSR for low-power BIST (2008), Electronics Letters Vol. 44, no. 6, pp. 401-402, 13 March 2008.
- [9] Tang, H., Qin, T., Hui, Z., Cheng, P., & Bai, W. (2018). Design and Implementation of a Configurable and Aperiodic Pseudo Random Number Generator in FPGA. 2018 IEEE 2nd International Conference on Circuits, System and Simulation, ICCSS 2018, 47-51.
- [10] Rukhin, A., Soto, J., & Nechvatal, J. (2010). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Nist Special Publication, 22(April), 1/1--G/1.4