Comparative Power Analysis of LFSR Test Pattern Generators

H. Srikanth Kamath
Assistant Professor
ECE Department, MIT
Manipal University, India

Aakash Nath, Shobhit Kumar Srivastava, Saket Garg
Students
ECE Department, MIT
Manipal University, India

ABSTRACT

Automatic Test Pattern Generation Automatic Test Pattern Generator is an electronic de-sign automation method used to find an input sequence that, when applied to a digital circuit, enables automatic test equipment to distinguish between the correct circuit behavior and the faulty circuit behavior caused by defects. The designs selected for comparison use LFSR as its core central component around which the entire algorithms are based which contribute in construction of Automatic test pattern generator. The LFSR is considered as it gives excellent random characteristics and has a low area overhead which allows it to be placed in the integrated circuit.

General Terms

Random Pattern Generators for testing.

Keywords

LFSR, random, pattern, low power, four bits,

1. INTRODUCTION

If a product is designed, fabricated, tested, and it fails the test, then there must be a cause for the failure, either test was wrong or the fabrication process was faulty or the design was in-correct, etc. The role of testing is to detect whether something went wrong and the role of diagnosis is to deter-mine exactly what went wrong, Correctness and effectiveness of testing is most important for quality products. Quality and economy are two major benefits of testing. ATPG (Automatic Test Pattern Generation or Automatic Test Pattern Generator) is an electronic design automation meth-od/technology used to find an input (or test) sequence that, when applied to a digital circuit, enables automatic test equipment to distinguish between the correct circuit behavior and the faulty circuit behavior caused by defects. A defect is an error caused in a device during the manufacturing process. A fault model is a mathematical description of how a defect alters de-sign behavior. The logic values observed at the device's primary outputs, while applying a test pattern to some device under test (DUT), are called the output of that test pattern. The output of a test pattern, when testing a fault-free device that works exactly as designed, is called the expected out-put of that test pattern [2]. A fault is said to be detected by a test pattern if the output of that test pattern, when testing a device that has only that one fault, is different than the expected output.

2. BIST and LFSR

Built In Self-Test which is one of the most efficient ways of testing present. BIST is a design technique in which parts of a circuit are used to test the circuit itself. BIST techniques are classified as on-line BIST which includes concurrent and non-concurrent techniques and off-line BIST which includes functional and structural approaches. The main purpose of BIST is to re-duce the complexity, and thereby decrease the cost and reduce reliance upon external (pattern-programmed) test equipment. BIST reduces cost in two ways which are reducing test-cycle duration and reducing the complexity of the test/probe setup, by reducing the number of I/O signals that must be driven/examined under tester control. One of the methods to generate test patterns in BIST is Linear Feedback Shift Register (LFSR) which acts as the seed generator which is then fed into the Circuit under Test (CUT).

Linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus, an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle.

3. ALGORITHMS

3.1 Simple LFSR

Here a four bit LFSR is taken into consideration which is made by using four D flip flops connected in a serial in parallel out form and a XOR gate [5]. This is expected to give all the sixteen patterns but it gets stuck at ‘0000’ and hence only fifteen patterns are generated at a particular moment. Hence an initial seed needs to be set. The connections are decided by the characteristic equation of the LFSR which in the case of four bits is $x^4 + x^3 + 1$[4].

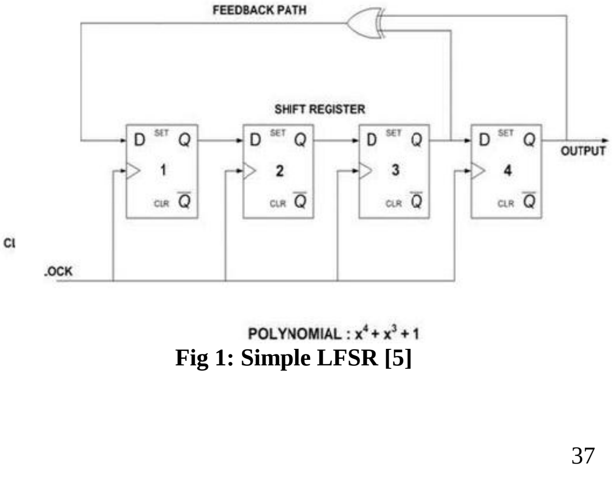

Fig 1: Simple LFSR [5]

3.2 Modified LFSR

To overcome this stuck at fault a modified LFSR was designed .In this to get all the patterns we add a couple of logic gates and it is assumed that the addition of these logic gates have not increased the area overhead and power consumption by a great amount so it is justified to add them to get all the patterns. The gates used are three input NOR gate and two XOR gates each with two inputs rather than the single XOR gate. The rest of the part is similar to that of Simple LFSR which is made by four D flip flops connected in Serial in Parallel Out form.

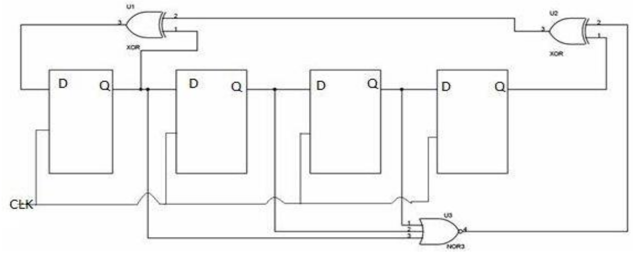

Fig 2: Modified LFSR

3.3 Unit Switching LFSR

In this algorithm the main aim is to minimize the switching power as it can be minimized using various techniques and in a well-designed circuit switching power plays a pivotal role in the total power consumption. The switching power is reduced by reducing the switching activity. To achieve this a seed generator is used, a n bit counter, gray code converter, XOR gates and nor gates. In this the counter will produce the sequence and nor operation will be performed with the previous term of the gray encoder which then will be fed to and gate which will supply the clock for seed generator. Then there will be XOR operation between the seed generator and the gray converter output. If the seeds are selected carefully then we have a single change code hence reducing the switching considerably which will result in reduction of the switching power [1].

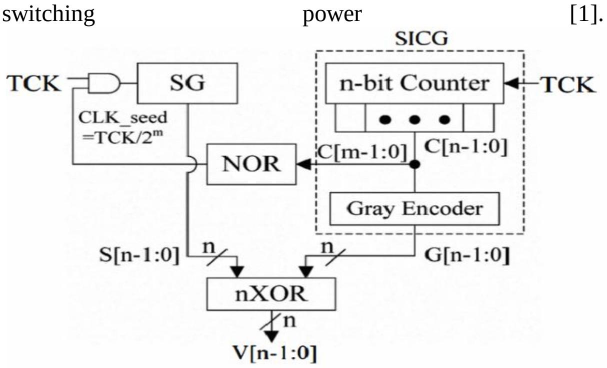

Fig 3: Unit Switching LFSR [1]

3.4 Bit Swapping LFSR

After the following algorithms then a new technique called the Bit Swapping LFSR was implemented. This is a technique in which multiplexer act as the means to reduce the switching activity. Here the bits are swapped if the switching exceeds a threshold value as compared to the previous pattern. In this algorithm the reduction of switching activity is quite significant when the pattern length is grater as compared to that smaller length of same algorithm. It was simulated and it showed reduction of switching activity [5].

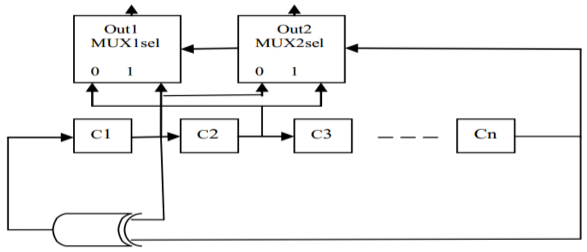

Fig 4: Bit Swapping LFSR [5]

4. SIMULATION AND RESULTS

To check the working and observe the various patterns the algorithms were initially constructed using VHDL coding and after satisfactory results the power analysis of the same was done[6]. After this these algorithms were simulated in Cadence Virtuoso by using schematics and their power analysis was done. A 180 nm technology was used for the schematics and the test conditions for the analysis were set as follows:

- Voltage: 1.8Volts
- Frequency: 800Mhz
- Rise Time: 100ps
- Fall Time:100ps
- Duty Cycle:50%

The power of the individual algorithms were first found and then their test power was found by putting the patterns in a test circuit (4:1 mux in this case) and he values were tabulated.

1: Observed Patterns

Simple LFSR	Modified LFSR	Unit Switching (n=3)	Bit Swapping
0001	1010	-	0010
1000	1011	-	0100
1100	0110	-	1100
1110	0011	-	1110
1111	1001	-	1111
0111	0100	-	0111
1011	0010	-	1011
0101	0001	1101	0110
1010	0000	1100	1010
1101	1000	0100	1101
0110	1100	0000	0101
0011	1110	0010	0011
1001	1111	0011	1001
0100	0111	1011	1000
0010	1011	0101	0001
Not Possible	0101	1101	0000
Total Number Of Switching per pattern			
2	2	1	1.625

Table 2: Average Power for Pattern Generator

Generating Power	
Algorithm	**Power Consumed**
Simple LFSR	0.960mW
Modified LFSR	0.978mW
Bit Swapping LFSR	1.137mW
Unit Switching LFSR	1.250mW

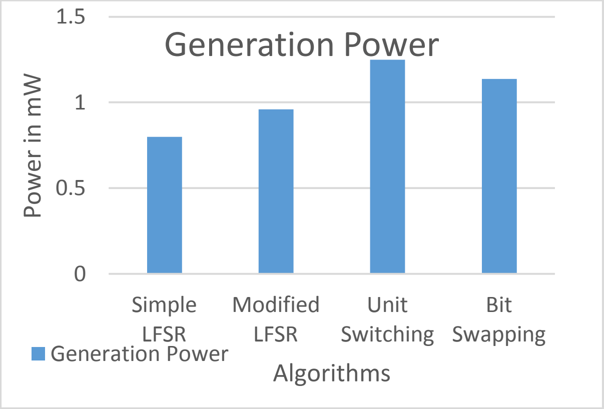

Fig 5: Graphical representation of Generation Power

Table 3: Average power during Testing

Power Consumption		
Algorithm	**Power Consumed**	**Power Saving**
Modified LFSR	0.814mW	NA
Bit Swapping	0.306mW	62.20%
Unit Switching	0.434mW	46.60%

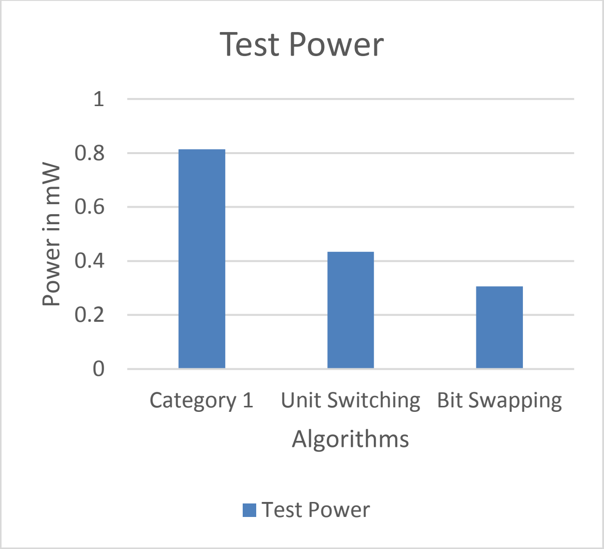

Fig 6: Graphical Representation of Test Power

5. CONCLUSION

The power saving in testing is quite evident from the data above. The overhead generating power is compensated if the number of bits is increased and where testing take the chunk of the power consumed in the total setup. During test power simple LFSR was not considered as it did not give all the pat-terns. The results are quite significant if there is great amount of testing required and where flexibility is required for the number of pat-terns generated.

6. ACKNOWLEDGMENTS

We take this opportunity to express a deep sense of gratitude to Prof. S N Bhat, Associate Professor, MIT Manipal, for his cordial support, valuable information and guidance, which helped us in completing this task through various stages. The constant inputs were life savers and were the driving force behind this project. We are obliged to staff members, Department of Electronics and Communication, for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of this assignment. A very special mention is due to the HOD Dr K. Prabhakar Nayak for his unwavering support.

7. REFERENCES

[1] BO YE and Tian-Wang Li," A novel BIST scheme for low power testing," 2010 IEEE.

[2] P. Girard," survey of low-power testing of VLSI circuits," IEEE design and test of computers, Vol. 19,no.3,PP 80-90,May-June 2002.

[3] S. Wang and S.K. Gupta," DS-LFSR: a BIST TPG for low switching activity," IEEE Trans computer-aided design of Integrated circuits and systems, Vol. 21, No.7,pp.842-851, July 2002.

[4] Low power, Low Transition random pattern generator.2012

[5] Mark Goresky and Andrew M Klapper, "Fibonacci and Galois Representations of Feedback with Carry Shift Registers", IEEE, Vol.48, No.11, November 2002.

[6] J Bhasker,"A VHDL Primer", Prentice Hall,3rd Edition,8120323661

[7] H Roth, Lizzy Kurian John, Digital System Design Using VHDL 2nd Edition.