International Journal of VLSI System Design and Communication Systems

ISSN 2322-0929
Vol.04, Issue.13,
December-2016,
Pages:1461-1465

www.ijvdcs.org

VLSI Design for PRPG Circuit in Built In Self Test for Compression

M. AMRUTHA[1], S. T. MRUDULA[2]

[1]PG Scholar, Dept of ECE, Sri Krishnadevaraya Engineering College, Gooty, AP, India,
E-mail: mamrutha76@gmail.com.

[2]Associate Professor, Dept of ECE, Sri Krishnadevaraya Engineering College, Gooty, AP, India,
E-mail: mrudulasunkara@gmail.com.

Abstract: The main challenging areas in VLSI are performance, cost, and power dissipation. Due to switching i.e. the power consumed testing, due to short circuit current flow and charging of load area, reliability and power. This project describes a low-power (LP) programmable generator capable of producing pseudorandom test patterns with desired toggling levels and enhanced fault coverage gradient compared with the best-to-date built-in self-test (BIST)- based pseudorandom test pattern generators. It is comprised of a linear finite state machine (a linear feedback shift register or a ring generator) driving an appropriate phase shifter, and it comes with a number of features allowing this device to produce binary sequences with preselected toggling (PRESTO) activity. We introduce a method to automatically select several controls of the generator offering easy and precise tuning. The same technique is subsequently employed to deterministically guide the generator toward test sequences with improved fault-coverage-to pattern-count ratios. Furthermore, this proposes an LP LFSR test compression method that allows shaping the test power envelope in a fully predictable, accurate, and flexible fashion by adapting the PRESTO-based logic BIST (LBIST) infrastructure. The proposed architecture is extended in such that the patterns generated from PRPG is gone through CUT and then to TRA to perform ATE.

Keywords: PRESTO, LFSR, BIST and Verilog HDL.

I. INTRODUCTION

Although over the next years, the primary objective of manufacturing test will remain essentially the same to ensure reliable and high quality semiconductor products—conditions and consequently also test solutions may undergo a significant evolution. The semiconductor technology, design characteristics, and the design process are among the key factors that will impact this evolution. With new types of defects that one will have to consider to provide the desired test quality for the next technology nodes such as 3-D, it is appropriate to pose the question of what matching design-for-test (DFT) methods will need to be deployed. Test compression, introduced a decade ago, has quickly become the main stream DFT methodology. However, it is unclear whether test compression will be capable of coping with the rapid rate of technological changes over the next decade. Interestingly, logic built-in self-test (LBIST), originally developed for board, system, and in-field test, is now gaining acceptance for production test as it provides very robust DFT and is used increasingly often with test compression. This hybrid approach seems to be the next logical evolutionary step in DFT. It has potential for improved test quality; it may augment the abilities to run at-speed power aware tests, and it can reduce the cost of manufacturing test while preserving all LBIST and scan compression advantages. For example, employs an LFSR to feed scan chains through biasing logic and T-type flip-flop. Since this flip-flop holds the previous value until its input is asserted, the same value is repeatedly scanned into scan chains until the value at the output of biasing logic (e.g., a k-input AND gate) becomes 1.

II. PROJECT DESCRIPTION

A. Existing System

The main challenging areas in VLSI are performance, cost, power dissipation is due to switching i.e. the power consumed testing, due to short circuit current flow and charging of load area, reliability and power. The demand for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation VLSI circuits. The power dissipation during test mode is 200% P more than in normal mode. Hence it is important aspect to optimize power during testing. Power optimization is one of the main challenges. The test pattern generator produces test vectors that are applied to the tested circuit during pseudo-random testing of combinational circuits. The nature of the generator thus directly influences the fault coverage achieved. The influence of the type of pseudo-random pattern generator on stuck-at fault coverage. Linear feedback shift registers (LFSRs) are mostly used as test pattern generators, and the generating polynomial is primitive to ensure the maximum period. We have shown that it is not necessary to use primitive polynomials, and moreover that their using is even undesirable in most cases. This fact is documented by statistical graphs. The necessity of the proper choice of a generating polynomial and an LFSR seed is shown here, by designing a mixed-mode BIST for the

ISCAS benchmarks As the complexity of VLSI circuits constantly increases, there is a need of a built-in self-test (BIST) to be used. A typical BIST architecture consists of

- TPG - Test Pattern Generator
- TRA – Test Response Analyzer
- Control Unit

As shown in fig.1 below.

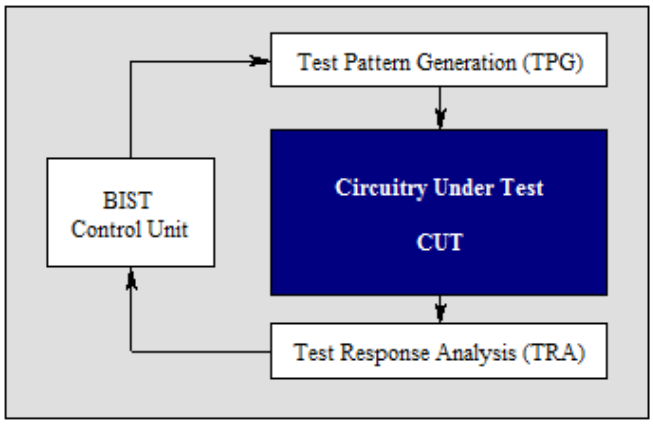

Fig.1. Test Pattern Generator.

It generates test pattern for CUT. It will be dedicated circuit or a micro processor. Pattern generated may be pseudo random numbers or deterministic sequence. Here we are using a Linear Feedback Shift Register for generating random number.

Test Response Analyzer (TRA): TRA will check the output of MISR & verify with the input of LFSR & give the result as error or not.

BIST Control Unit: Control unit is used to control all the operations. Mainly control unit will do configuration of CUT in test mode/Normal mode, feed seed value to LFSR, Control MISR & TRA. It will generate interrupt if an error occurs. You can clear interrupt by interrupt_clear_i signal.

Circuit under Test (CUT): CUT is the circuit or chip in which we are going to apply BIST for testing stuck at zero or stuck at one error.

Need for using BIST Technique: Today's highly integrated multi-layer boards with fine-pitch ICs are virtually impossible to be accessed physically for testing. Traditional board test methods which include functional test, only accesses the board's primary I/Os, providing limited coverage and poor diagnostics for board-network fault. In circuit testing, another traditional test method works by physically accessing each wire on the board via costly "bed of nails" probes and testers. To identify reliable testing methods which will reduce the cost of test equipment, a research to verify each VLSI testing problems has been conducted. The major problems detected so far are as follows:

- Test generation problems
- Gate to I/O pin ratio

B. Test Generation Problems

The large number of gates in VLSI circuits has pushed computer automatic-test-generation times to weeks or months of computation. The numbers of test patterns are becoming too large to be handled by an external tester and this has resulted in high computation costs and has outstripped reasonable available time for production testing.

C. The Gate to I/O Pin Ratio Problem

As ICs grow in gate counts, it is no longer true that most gate nodes are directly accessible by one of the pins on the package. This makes testing of internal nodes more difficult as they could neither no longer be easily controlled by signal from an input pin (controllability) nor easily observed at an output pin (observe ability). Pin counts go at a much slower rate than gate counts, which worsens the controllability and observe ability of internal gate nodes.

Basic Architecture: An n-bit PRPG connected with a phase shifter feeding scan chains forms a kernel of the generator producing the actual pseudorandom test patterns. A linear feedback shift register or a ring generator can implement a PRPG. More importantly, however, n hold latches are placed between the PRPG and the phase shifter. Each hold latch is individually controlled via a corresponding stage of an n-bit toggle control register. As long as its enable input is asserted. Data moving from the PRPG to the scan chains. Two additional parameters kept in 4-bit Hold and Toggle registers determine how long the entire generator remains either in the hold mode or in the toggle mode, respectively. To terminate either mode, a 1 must occur on the T flip-flop input. This weighted pseudorandom signal is produced in a manner similar to that of weighted logic used to feed the shift register as shown in Fig.2. The T flip-flop controls also four 2-input multiplexers routing data from the Toggle and Hold registers. It allows selecting a source of control data that will be used in the next cycle to possibly change the operational mode of the generator .For example, when in the toggle mode, the input multiplexers observe the Toggle register. Once the weighted logic outputs 1,the flip-flop toggles, and as a result all hold latches freeze in the last recorded state. They will remain in this state until another 1 occurs on the weighted logic output. The random occurrence of this event is now related to the content of the Hold register, which determines when to terminate the hold mode.

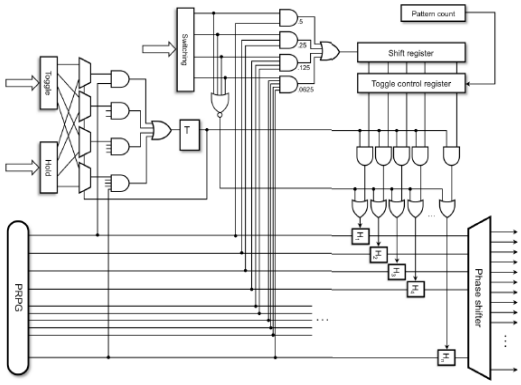

Fig. 2. Fully operational version of PRESTO.

III. PROPOSED ARCHITECTURE

The proposed low power LFSR technique uses bit swapping technique to reduce the peak power. By connecting multiplexers on the LFSR register. the number of transitions are decreased for that cell which are under bit swapping. The number of transitions in each register in LFSR without applying bit swapping technique here two cells in an nbit LFSR are considered to be adjacent if the output of one cell feeds the input of the second directly (i.e., without an intervening XOR gate). Each cell in a maximal-length n-stage LFSR (internal or external) will produce a number of transitions equal to 2n-1 after going through a sequence of 2n clock cycles. The sequence of 1s and 0s that is followed by one bit position of a maximal-length LFSR is commonly referred to as an m sequence. Each bit within the LFSR will follow the same m sequence with a one-time-step delay. The m-sequence generated by an LFSR of length n has a periodicity of 2n-1. It is a well-known standard property of an m-sequence of length n that the total number of runs of consecutive occurrences of the same binary digit is 2n-1.The beginning of each run is marked by a transition between 0 and 1.Therefore, the total number of transitions for each stage of the LFSR is 2n-1.

One way to improve the correlation between the bits of the successive vectors is to avoid frequent transitioning of the logic levels of the primary inputs. The new approach entails inserting 3 intermediate vectors between every two successive vectors. The total number of signal transitions between these 5 vectors is equal to the total number of signal transitions between the 2 successive vectors generated using the conventional approach. This reduction of signal transition activity in the primary inputs reduces the switching activity inside the design under test and therefore results in reduced power Consumption by the device under test. The additional circuitry used to accomplish the generation of the 3 intermediate vectors is minimal at best consisting of few logic gates.

IV. INTERNAL STRUCTURE OF 8-BIT LP-LFSR

The number of LFSR outputs required is driven by the number of test inputs required for circuit under test. The technique of inserting 3 intermediate vectors is achieved by modifying the conventional LFSR circuit with two additional levels of logic between the conventional flip-flop outputs and the low power outputs as shown in Fig.3. The first level of hierarchy from the top down includes logic circuit design for propagating either the present or the next state of the flip-flops to the second level of hierarchy. The second level of hierarchy is a multiplexer function that provides for selecting between the two states (present or next) to be propagated to the outputs as low power output. inimal at best consisting of few logic gates.

Step1: if en1 en2=10, sel1,sel2=11.

The first half of LFSR is in active mode and second half of lfsr is in idle mode.by selecting selection inputs both halves of lfsr are sent to the outputs..hence generating test vector is t1.

Step 2: if en1,en2=00, sel1 sel2=10

Both halves are in idle mode. first half is sent to the outputs but the Injector flip flop outputs are sent to outputs .hence generating test vector is Ta.

Step3: if en1 en2=01,sel1 sel2=11

The first half of lfsr is in idle mode. second half of LFSR is in active mode. Hence generating test vector is Tb.

Step4: if en1 en2=00,sel1 sel2= 01

Both halves are in idle mode .from first half the injector outputs ar sent to outputs of lfsr and second half sends exact bits in lfsr. hence generating test vector is Tc.

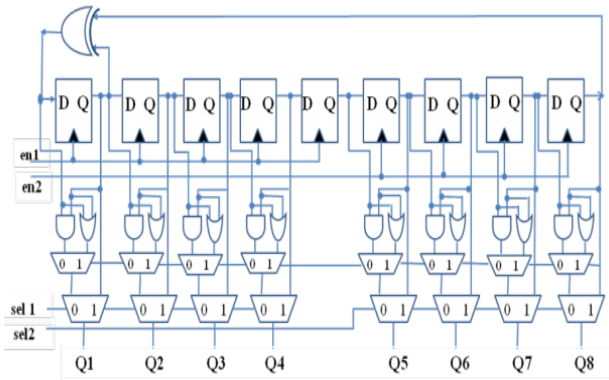

Fig 3. Internal structure of 8-bit LP-LFSR.

V. WORKING

In the simulation environment, the outputs of the flip-flops are loaded with the seed vector. The feedback taps are selected pertinent to the characteristic polynomial $x^8 + x + 1$. Only 2 inputs pins, namely test enable and clock are required to activate the generation of the pattern as well as simulation of the design circuit. It is also noteworthy here that the intermediate vectors in addition to aiding in reducing the number of transitions can also empirically assist in detecting faults just as good as the conventional LFSR patterns. Description of the technique to produce low power pattern for BIST The following is a description of a low power test pattern generation technique as depicted in the 9-bit LFSR based schematic in Fig.4. Verilog based test bench is used in assigning the initial output states (0100 1011) of the 9-bit LFSR. The feedback taps are designed for maximal length LFSR generating all zeros and all one's as well. The first step is to generate T1, the first vector by enabling (clocking) the first 4-bits of the LFSR and disabling (not clocking) the last 4 bits. This Shifts the first 4 bits to the right by one bit. The feedback bits of the LFSR are the outputs of the 8th and the first flip-flop. The output of the 8^{th} flip-flop is 1 and the output of the first flip-flop is 0. The exclusive-or of the 8^{th} -flip-flop (logic 1 in this case) and the first flip-flop(logic 0 in this case) is input (1 EXOR 0 = 1 into the first D flip-flop. The new pattern in the first four bits of the LFSR is 1010.

Note that the shaded register is clocked along with the first 4 bits of the LFSR. So the input of the shaded flip-flop is the output of the 4^{th} flip-flop which in this case is 0. Also note

that prior to the first clock, the input of the shaded register was the seed value of the 4th flip-flop at the output of the 4th flip-flop which in this case is 0. So after the first clock this value of 0 will now appear at the output of the shaded flip-flop. In other words the value of the 4th output is stored in this shaded register and is used in the next few steps. The first 4 shifted bits of the LFSR and the last 4 un-shifted bits (i.e. the seed value) are propagated as T1 (1010 1011) to the final outputs. Next few steps involve generating the 3 intermediate patterns from T1. These patterns are defined as Ta, Tb and Tc shown in below Ta is generated by maintaining (disabling the clock to the first 4 bits) the first four bits of the LFSR outputs (as is from T1) as the final first four low power outputs 1010. Note that the clock to the last four bits of the LFSR is also disabled.

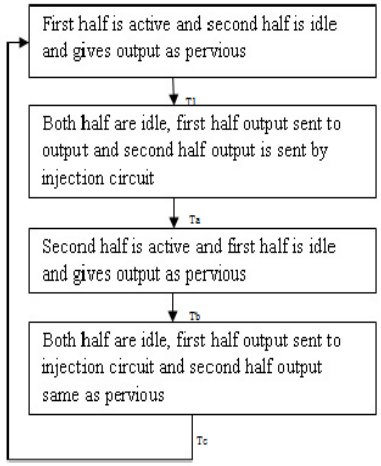

Fig.4. Proposed algorithm for low power LFSR.

The last four bits however are the outputs from the Injector flipflops. The Injector flipflop compares the next value (the input of the D-flip-flop) with the current value (the output of the D-flip-flop). According to T1, the outputs (current values) of the last 4 bits of the LFSR are 1011. The next values are the values at the inputs of the D-flip-flops which in this case are 0101**. C**ompare the current values (1011) bit by bit with the next values (0101). If the values bit by bit are not the same then use the random generator feedback R (in this case is logic 1) as the bit value as shown in the schematic above. If however both values bit by bit are the same then propagate that bit value to output as opposed to the R bit. This bit by bit comparison gives us the last four bits of Ta to be 1111. Therefore Ta = 1010 1111. Next step is to generate Tb. Shift the last 4 flip-flops to the right one bit but do not shift the first 4 flip-flops to the right. The clock to the first 4 bits plus the shaded flipflop is disabled. The clock to the last 4 bits is enabled. Propagate the outputs of the flip-flops of the entire LFSR as opposed to the outputs of the injection circuit to the outputs (low power).

The injection circuits are disabled. As in Ta, maintain the first four LFSR outputs (1010) as the low power outputs. Again from Ta, the inputs of the last four D flip-flops from the previous step (generating Ta) are 0101. Also note that the output of the shaded register is 0 from the previous step (generating Ta). Therefore the input of the 5th flip-flop is a 0. The outputs of the last 4 flip-flops are 0101 resulting in Tb = 1010 0101. The 3rd intermediate vector Tc is generated via disabling the clock to the entire LFSR. Propagate the first 4 outputs from the injection circuit as the first 4 low power outputs and maintain the last 4 low power outputs the same as Tb. Generating injection circuit outputs for Tc is conceptually the same as explained above in generating Ta. Current values (the outputs of the flip-flops) of the first four flip-flops are compared with the next values (the inputs of the flip-flops) of the flip-flops. The feedback from the 8th flip-flop is 1 (please see generating Tb). Therefore the logical feed forward value of R is 1. The feedback value from the first flip-flop is also 1 as per the current values above. The exclusive or of two ones is a 0. Therefore the input to the first flip-flop is a 0 which is also the next state of the first flip-flop. Hence the next values are 0 for the first flip-flop and 101 for the 2nd, 3rd and 4th flip-flop respectively.

The next values are 0101. The first four outputs from the injection circuit are 1111. The last 4 outputs are the same as Tb which are 0101 resulting in the 3rd and final intermediate vector Tc = 1111 0101. Generating T2 is quite similar to generating T1. As in Tc the outputs of the last four LFSR flops are 0101. The outputs of the first 4 flip-flops of the LFSR are the current values which are 1010. Therefore the seed vector for generating T2 is 1010 0101. Shift the first four bits of the LFSR plus the shaded flip-flop. Do not clock the last four flip-flops. Propagate the outputs of the entire LFSR to the final low power outputs. The output of the 8th flip-flop from the previous step (generating Tc) is a 1 and the output of the first flip-flop from the previous step (generating Tc) is also a 1. The exclusive or of the output of the 8th flip-flop and the first flip-flop is 0. Therefore the input to the first flip-flop will be a 0. The inputs to the 2nd, 3rd, 4th and the shaded flip-flops are 1010. These are also the current values from the previous step (generating Tc). Shifting the first four flip-flops of the LFSR to the right by one bit results in 0101 as the outputs of the first four flip-flops. Therefore T2 generated is 0101 0101.

VI. RESULTS

Results of this paper is as shown in bellow Figs.5 to 8**.**

A. RTL Schematic Diagram

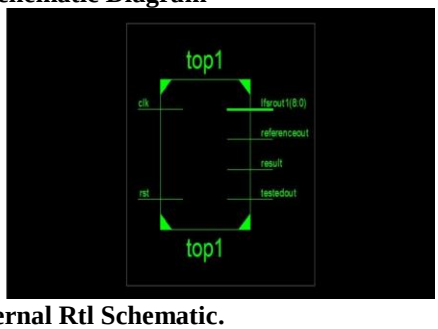

Fig.5. Internal Rtl Schematic.

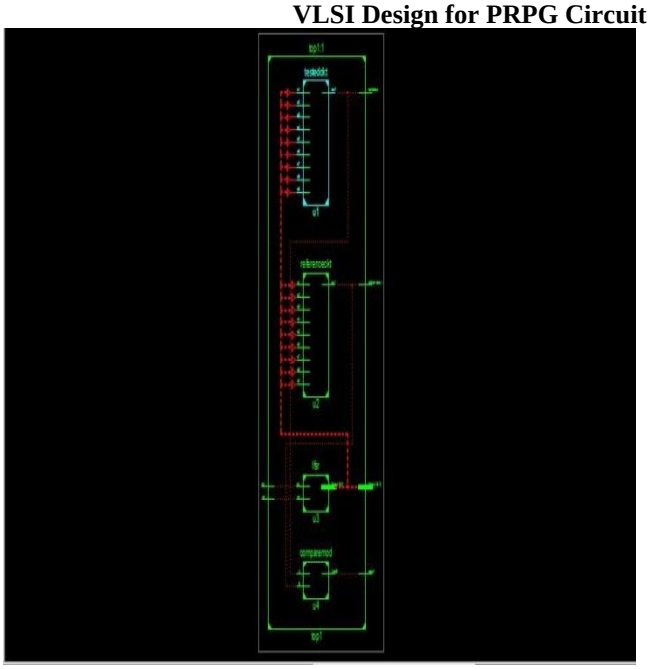

Fig.6. Technology Schematic.

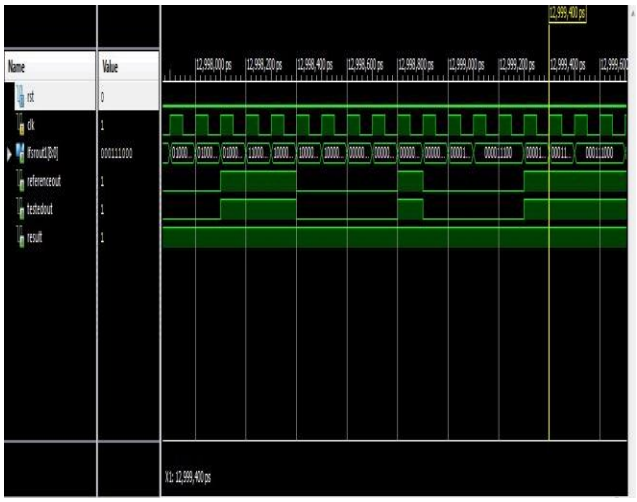

Fig.7. Waveforms.

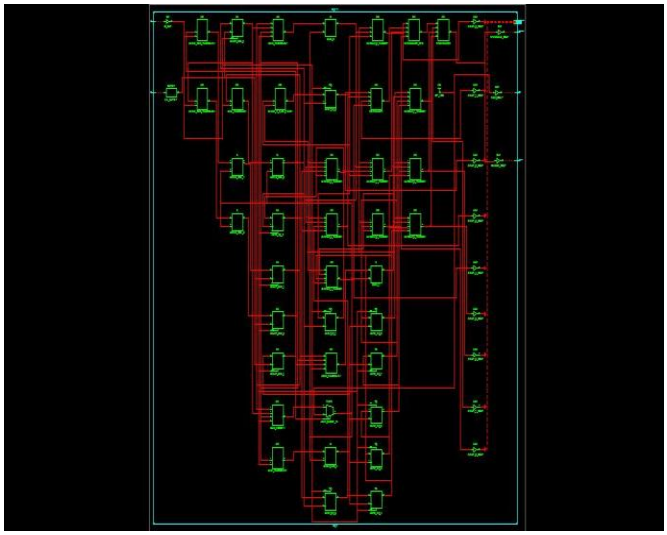

Fig.8. Technological Schematic View.

TABLE I: Design and Summary Reports

top1 Project Status			
Project File:	prpg_proposed.xise	Parser Errors:	No Errors
Module Name:	top1	Implementation State:	Synthesized
Target Device:	xc3s500e-5vq100	• Errors:	No Errors
Product Version:	ISE 12.3	• Warnings:	7 Warnings (7 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	14	4656	
Number of Slice Flip Flops	19	9312	
Number of 4 input LUTs	23	9312	
Number of bonded IOBs	14	66	
Number of GCLKs	1	24	

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sun Jun 5 15:20:12 2016	0	7 Warnings (7 new)	6 Infos (6 new)
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

VII. CONCLUSION AND FUTURESCOPE

The proposed approach shows the concept of reducing the transitions in the test pattern generated. The transition is reduced by increasing the correlation between the successive bits. The simulation results shows that how the patterns are generated for the applied seed vector. This paper presents the implementation with regard to Verilog HDL. Synthesizing and implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx - Project Navigator, ISE 12.3i suite. The power reports shows that the proposed low power LFSR. Future work will include additional circuit optimizations to further reduce the power dissipation by adapting dynamic and analog implementations for the comparator resolution module and a high-speed zero-detector circuit for the decision module.

VIII. REFERENCES

[1]. Michael L.Bushnell, Vishwani D.Agawal," Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits," Kluwer Academic Publishers, 2000.

[2]. Mohammad Tehranipoor, Mehrdad Nourani, Nisar Ahmed," Low- Transition LFSR for BIST-Based Applications," 14th Asian Test Symposium, pp. 138- 143, 18-21 Dec. 2005.

[3]. F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda," A Test Pattern Generation methodology for low power onsumption," pp.1-5, 2008.

[4] P.Glard et al "Survey 0 F Low-Power Testing Of Vlsi Circuits"IEEE Design & Test Of Computers, vol. 19, no. 3. (2002), pp. 80-90.

[5]. Dr.K.Gunavathi, Mr. K. ParamasivaM, Ms.P.Subashini Lavanya, M.Umamageswaran," A novel BIST TPG for testing of VLSI circuits", IEEE International Conference on Industrial and Information Systems, pp. 8 - 11, August 2006.