ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

An Narrative Methodologies in ATPG For Low Power BIST in Digital VLSI Circuits

G.Sathishkumar, Dr.V.Saminadan

Research Scholar, Department of ECE, Pondicherry Engineering College, Puducherry.

Professor, Department of ECE, Pondicherry Engineering College, Puducherry.

ABSTRACT: Test vectors are generated for post manufacturing test of a digital system. Because of the complexity of digital systems, the size of necessary tests, and test quality factors, automatic methods are used for generation of test patterns. This process is referred to as automatic test pattern generation (ATPG). For a circuit under test (CUT), test pattern generation must be due to the testing of the circuit as thoroughly as possible, and in the shortest possible time.ATPG is done by the utilization of programs, methods, and algorithms; all of which use some forms of circuit and fault models. Often, ATPG refers to test generation from a net list model of CUT using the stuck-at fault model. This paper shows the basics of ATPG methods and shows how test generation programs fit in the overall test cycle. We focus on random test generation (RTG) methods for digital VLSI circuits (On-chip UART).

KEYWORDS: Random test pattern generator, ATPG, BIST, CUT, RTG, UART.

I. INTRODUCTION

In this section, basic methods of test generation using several small examples are presented. The purpose is to familiarize the reader with the basics of test generation procedures, the terminology, categorization of ATPG algorithms, and the role of various utilities facilitating test generation.

Test vectors are generated to detect faults. A test vector is an input vector that creates different outputs for faulty and good circuits. Test generation is finding such input vectors. Functionally, we can find such inputs by finding the Boolean difference of a good circuit model and its faulty model, and then finding input vectors that satisfy this difference .

For Boolean difference, we start with good circuit model, and modify it according to the fault that we are seeking the test for. The XOR of the good circuit and faulty circuit is the equation for the Boolean difference. Test(s) for the given fault are the input vectors that make the Boolean difference 1.

Defects of all types can be processed similarly, as long as they can be modeled by Boolean functions. Faults that make Boolean differences of all outputs equal to 0 are undetectable. The Boolean difference calculation can be regarded as the ultimate solution for generating tests for given circuit faults. This method is exact, covers all fault types, and is complete in that it finds all tests for fault being considered. However, this problem cannot be solved in no polynomial time complexity.

II.Test Generation Process

We present test generation techniques that try to simplify exhaustive solutions discussed above. For this, we use the same example we used above. A deterministic and a random method is presented to optimize the search of test vectors.

Deterministic Search: For generating a test for l_8:*SA*0, we started at the site of the fault line l_8. Suppose this fault exists in an actual circuit and we are trying to make it show itself at a primary output. For this purpose, first we have to use values of the inputs at the circuit to drive a value into l_8 that is different than the faulty value. Since the fault we are trying to detect is a stuck-at-0 fault, we have to drive a 1 into this line. This requires *s* and *b* to be 1. Next, we need to

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

provide input conditions to opagate the effect of this fault to one of the circuit outputs. For this purpose, l_{11} and l_{12} must both be 0 to propagate the effect of $l_8{:}SA0$ to l_9 and then to w. Line l_{11} is already 0 since requiring $s = 1$ causes l_4 and thus l_{11} to become 0. The other condition for propagating $l_8{:}SA0$ is $l_{12} = 0$. This can be accomplished by $c = 0$. This analysis results in generation of test $asbc$ = X110 for $l_8{:}SA0$. The value of a is X, which means that either 0 or 1 is acceptable.
Random Search: An alternative to the deterministic search discussed above, is to use random test vectors and check for faults they can detect. Such a solution can result in the detection of a good number of faults with very few random tests.
Methods and Algorithms: In the above discussions, we casually discussed the ways of reaching test vectors for detecting faults, faster than xhaustive or complete solutions. These casual methods are the basis of more formal algorithms that are used for test generation. Other factors than just detecting faults that must be considered in test generation include the reduction of test vectors, reducing test time for testing the physical device, and the detection of multiple faults by the same test vectors. We deal these issues and the problem of fault detection in the test generation solutions that we present in this chapter.

III. Fault and Tests

As mentioned above, there are several ways that test vectors can be generated for a circuit. Some Search for a test using circuit topology, some use a functional model of the circuit, and some use a mix of both. This section categorizes various ways that tests can be generated.
Fault-oriented Test Generation
Considering a fault in a circuit, and then looking for a test that can detect the fault is fault-oriented test generation. This method of test generation is most appropriate when there are few faults remaining in the circuit to be detected.
Fault Independent Test Generation
In fault independent test generation, tests are generated independent of faults. In one case, a test is generated and then evaluated for faults it can detect, and in another scenario, tracing a circuit results in applying test vectors and faults that they can detect. In either case, specific faults cannot be targeted. Using fault independent test generation is most appropriate when there are still many faults left in the circuit to be detected. In this case, a random test or a deterministic test has a good chance of detecting a good number of faults.
Random Test Generation
RTG selects test vectors in random [4–8]. This is most efficient at the beginning of a test generation session when there are many undetected faults in the circuit. Often RTG programs are complemented with evaluation procedures for a better selection of test vectors. Some RTG programs target specific areas of a CUT that has faults that are hard to detect. Decisions about the number of random tests that can be useful in detecting faults, and expected the number of faults to detect can be made based on how many hard-to-detect faults are in the circuit, and how hard it is to detect them. Section 5.2 discusses controllability and observability that will be helpful in making some of these decisions.
Unspecified Inputs
In fault-oriented test generation, certain inputs do not play a role in detecting a fault and their values can be either 0 or 1. This was the case in generating a test for $l8{:}SA0$ in Fig. 5.1. We use ‘X’ for indicating values for such inputs. X’s are also possible in some RTG programs that use a subset of the inputs to target specific areas of a circuit.

Another solution for test generation that has the same problem as the Boolean difference is to apply all possible input combinations to the faulty circuit model and search for those that produce a different output than the good circuit. This has to be repeated for every fault. This solution for the test generation problem is an exhaustive search one, and like the Boolean difference, is not practical for large circuits.

Test generation techniques and algorithms that we discuss in this chapter try solving this problem using heuristics to limit the search space. The rest of this section covers some basics for these algorithms and definitions.

IV.Terminologies and Definitions

In the presentations that follow, we use terminologies that are brief and concise, and help understanding of the materials. Unambiguous definition of such terms is important that is described in this section. *Circuit Under Test*. As before, CUT, MUT, GUT, and FUT are used for Circuit, Model, Good circuit, and Fault able circuit that are being

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

tested. We continue using these labels for models for which test is being generated. *Stuck-at Models*. Unless otherwise specified, methods and algorithms in this chapter apply to the single stuck-at fault model. *Control Value*. A 0 input of an AND gate makes the output 0 regardless of values of all other AND inputs. This value is called control value. For an OR gate, a 1 is its control value. shows control values for four basic gates. A control value on an input of a gate blocks propagation of faults from other inputs. *Inversion Value*. Inversion value of a gate is 0 if no inversion is done, and it is 1 otherwise. For an AND gate the inversion value is 0, while the inversion value of a NAND is 1. Inversion in a path is 0 if there are even number of inversions, and is 1 if there are odd number of gates with 1 inversion values. A control value on an input of a gate generates the same value on the output if the gate has inversion 0, and the complement of the control value if the inversion is 1. For example, the control value on an input of a NOR gate (1) generates the complement of this value (0) on its output.*Test Efficiency*. Efficiency of a test vector is measured with the number of faults it detects. The required number of faults to detect for a test vector to be regarded as efficient depends on many factors. Some of these factors are which test generation method we are using, the remaining undetected faults, difficulty of detecting remaining faults, and where we are in the test generation process, i.e., just starting or near the end.

V.Controllability and Observability

Controllability is defined as a measure of difficulty of setting a circuit line to a certain value. Primary circuit inputs are the most controllable. Consider, for example, circuit shown in For finding a test for l7:SA1, we first need to have a 0 on this line. This being the AND gate control value, requires either input of the gate to be 0. We choose input a since it is more directly controlled and achieves the designated l7 value.

Observability is defined as a measure of difficulty of observing the value change of a line on a primary output. Primary circuit outputs are the most observable. SA1 can most easily be seen from the path shown to output .

Controllability and observability examples presented above were very simple and could be decided by inspection. However, for larger circuits and for lines and gates deep inside a circuit, calculation of controllability and observability are not as simple and more systematic methods are needed.

VI.Lawer SCOAP Controllability and Observability

In deterministic or random test generation, controllability and observability measures are used for simplifying the related algorithms. However, complexity in calculation of these parameters defeats the main purpose for which they are used. The fan-out problem, and the method of calculating probability based on controllability and observability parameters for circuits with reconvergent fan outs is too complex for the methods to be useful for test generation.
Sandia Controllability/Observability Analysis Program (SCOAP) [16] is a testability measure, t he complexity of which grows only linearly with the size of the circuit. SCOAP is based on the topology of the circuit, is a static analysis, and does have some inaccuracies due to reconvergent fan-out's. Nevertheless, it is easy to calculate and provide a good estimate for test generation programs, as well as design for test techniques. SCOAP defines a set of parameters for combinational and sequential controllability and observability measures. The combinational parameters have to do with the number of lines that need to be set for controlling and observing a line. The sequential parameters are related to the number of clocks it takes for controlling and observing a line. In SCOAP parameters, lower values mean more controllable and observable, and lines that are more difficult to control and observe have higher SCOAP parameter values.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

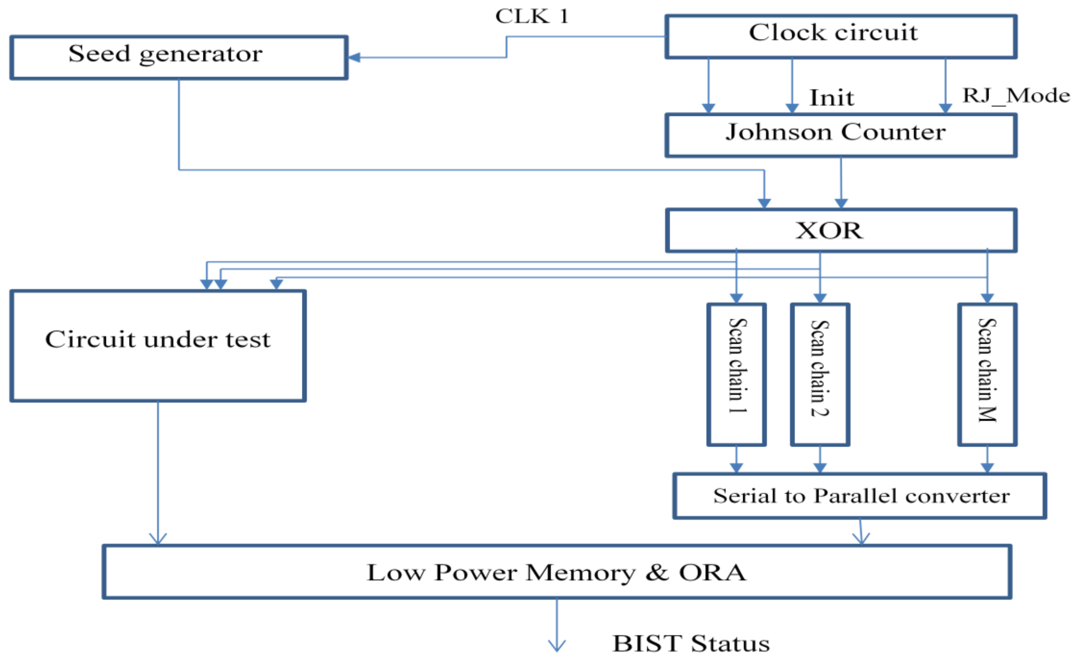

Fig.1 BS BIST LFSR

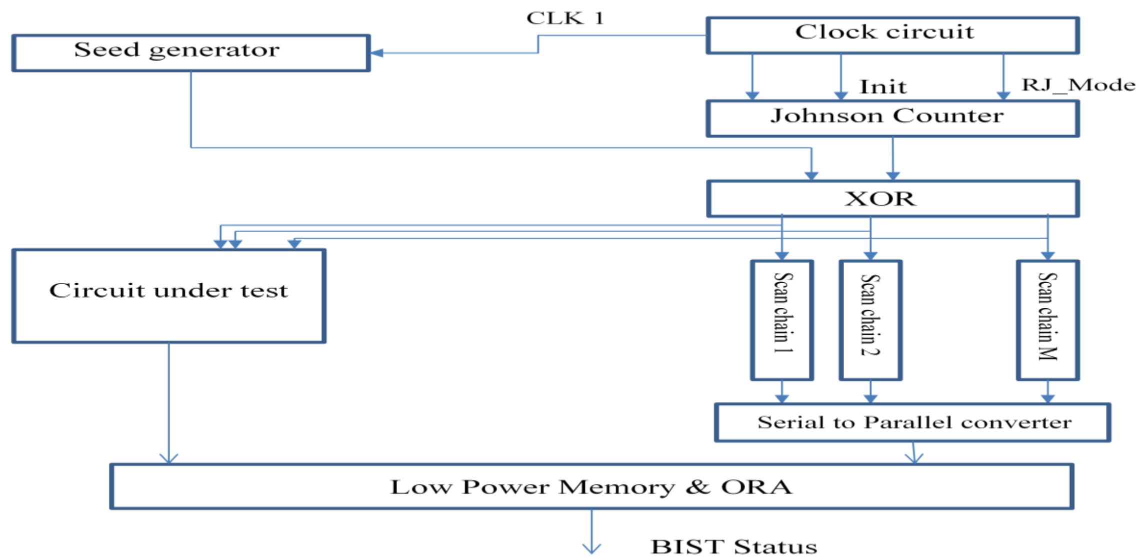

Fig.2 Bit Swapping LFSR

VII .BIT SWAPPING LFSR

In recent years, the design for low power has become one of the greatest challenges in high performance very large scale integration (VLSI) design. As a consequence, many techniques have been introduced to minimize the power consumption of new VLSI systems. However, most of these methods focus on the power consumption during normal mode operation, while

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

test mode operation has not normally been a predominant concern. However, it has been found that the power consumed during test mode operation is often much higher than during normal mode operation. This is because most of the consumed power results from the switching activity in the nodes of the circuit under test (CUT), which is much higher during test mode than during normal mode operation. Several techniques that have been developed to reduce the peak and average power dissipated during scanbased tests.[2] A direct technique to reduce power consumption is by running the test at a slower frequency than that in normal mode. This technique of reducing power consumption, while easy to implement, significantly increases the test application time.

VIII.UART Implementation

The universal asynchronous receiver/transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion.[1] At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is less costly than parallel transmission through multiple wires.

The UART usually does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signalling levels. External signals may be of many different forms. Examples of standards for voltage signaling are RS-232, RS-422 and RS-485 from the EIA. Historically, current (in current loops) was used in telegraph circuits. Some signaling schemes do not use electrical wires. Examples of such are optical fiber, IrDA (infrared), and (wireless) Bluetooth in its Serial Port Profile (SPP). Some signaling schemes use modulation of a carrier signal (with or without wires). Examples are modulation of audio signals with phone line modems, RF modulation with data radios, and the DC-LIN for communication. Communication may be *simplex* (in one direction only, with no provision for the receiving device to send information back to the transmitting device), *full duplex* (both devices send and receive at the same time) or *half duplex* (devices take turns transmitting and receiving).

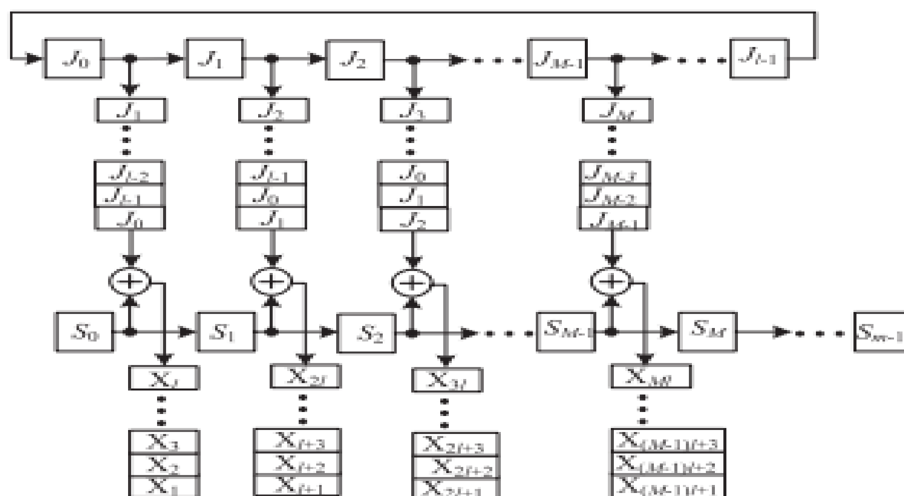

Fig.3 Test pattern Generation

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

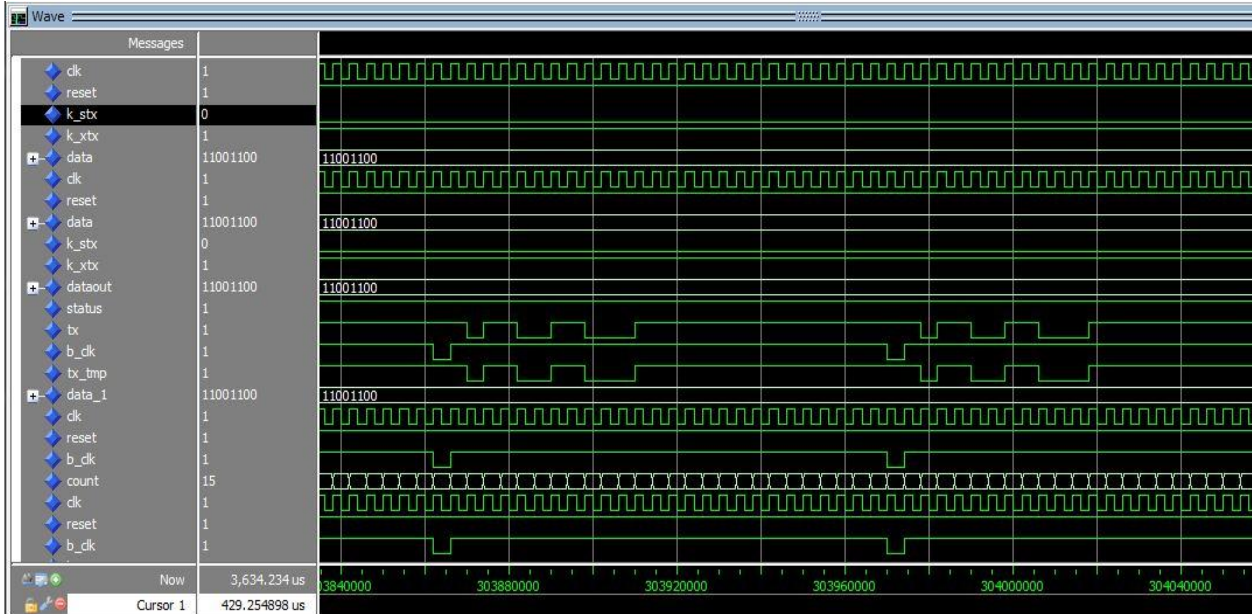

Fig.4 Simulation Result of UART transmitter

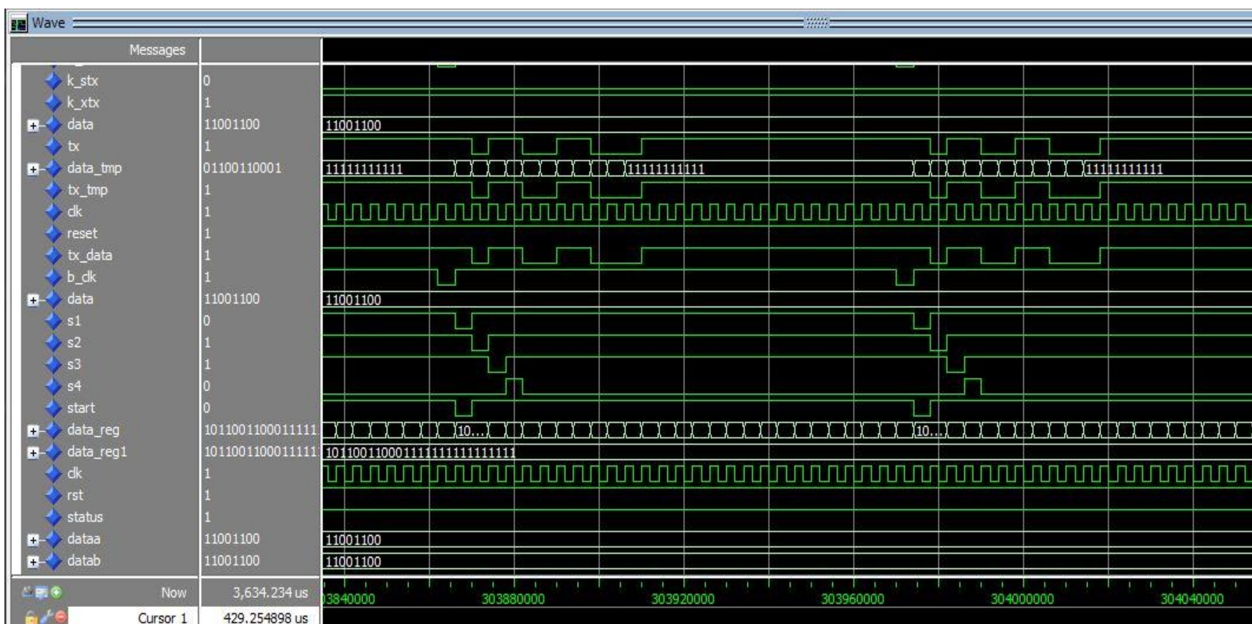

Fig.5 Simulation Result of UART Receiver

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

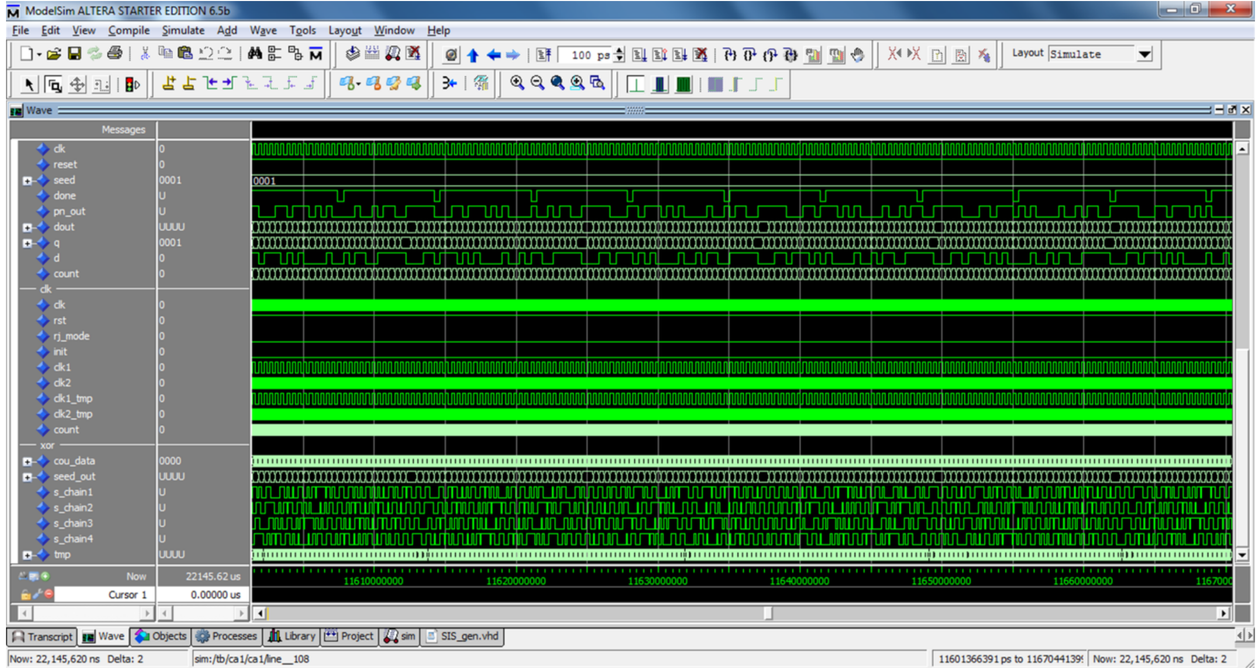

Fig.6 Simulation Result of Test Pattern Generation

Device	
Family	Spartan6
Part	xc6slx100
Package	fgg484
Temp Grade	C-Grade
Process	Typical
Speed Grade	-3

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	2	---	---
Logic	0.000	38	63288	0
Signals	0.000	78	---	---
IOs	0.000	22	326	7
Leakage	0.081			
Total	0.081			

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.038	0.000	0.038
Vccaux	2.500	0.009	0.000	0.009
Vcco25	2.500	0.005	0.000	0.005

	Total	Dynamic	Quiescent
Supply Power (W)	0.081	0.000	0.081

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Heat Sink	None
Custom TSA (C/W)	NA

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	16.4	83.7	26.3

Characterization	
Production	v1.3,2011-05-04

Fig.7 Power Analysis of LP-LFSR in UART

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

IX.CONCLUSION

The question of how well the TPG portion of the BIST circuitry tests itself is addressed in the graph in Figure 4, Fig.5, and Fig.6 in terms of single stuck-at gate-level fault coverage as a function of the number of clock cycles in the BIST sequence. As can be seen from the graph, 100% fault coverage was obtained for all TPG implementations.
However, the number of clock cycles required for 100% fault coverage gives an indication of how easy the TPG is to test. An easier to test TPG approaches 100% fault coverage more rapidly than a harder to test TPG. The time to obtain 100% fault coverage correlates with the area overhead of the TPG implementations. Therefore, the LFSR is not only the most area efficient design, but is also easiest to test. There is very little difference in the fault coverage curves for the internal feedback LFSR and CA implementations. However, the external feedback LFSR implementing the same polynomial takes a few more clock cycles to achieve 100% fault coverage than the internal feedback LFSR or the CA. As a result, the internal feedback LFSR with prim-itive characteristic polynomial is the best TPG in terms of area overhead, performance penalty, and the ability to test itself. It is easy to implement as long as one has access to a list of primitive polynomials, such as the ones given in Fig.7. The LFSR is also a key component in the design of output response analysis circuits. However, the pseudo-random patterns produced by the LFSR are not good for testing all CUTs. Therefore, it is important to keep the other TPGs in one's BIST tool kit.

REFERENCES

[1] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in 11th Annu. IEEE VLSI Test Symp. Dig. Papers, Apr. 1993,pp. 4–9.

[2] P. Girard, "Survey of low-power testing of VLSI circuits," IEEE Design Test Comput., vol. 19, no. 3, pp. 80–90, May–Jun. 2002.

[3] A. Abu-Issa and S. Quigley, "Bit-swapping LFSR and scan-chain ordering: A novel technique for peak- and average-power reduction in scan-based BIST," IEEE Trans. Comput.-Aided Design Integr. Circuits
Syst., vol. 28, no. 5, pp. 755–759, May 2009.

[4] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, J. Figueras,S. Manich, P. Teixeira, and M. Santos, "Low-energy BIST design:Impact of the LFSR TPG parameters on the weighted switching activity,"in Proc. IEEE Int. Symp. Circuits Syst., vol. 1. Jul. 1999, pp.110–113.

[5] S. Wang and S. Gupta, "DS-LFSR: A BIST TPG for low switching activity," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,vol. 21, no. 7, pp. 842–851, Jul. 2002.

[6] F. Corno, M. Rebaudengo, M. Reorda, G. Squillero, and M. Violante, "Low power BIST via non-linear hybrid cellular automata,"in Proc. 18th IEEE VLSI Test Symp., Apr.–May 2000, pp.29–34.

[7] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and H. Wunderlich,"A modified clock scheme for a low power BIST test pattern generator," in Proc. 19th IEEE VTS VLSI Test Symp., Mar.–Apr. 2001, pp. 306–311.

[8] D. Gizopoulos, N. Krantitis, A. Paschalis, M. Psarakis, and Y. Zorian,"Low power/energy BIST scheme for datapaths," in Proc. 18th IEEE VLSI Test Symp., Apr.–May 2000, pp. 23–28.

[9] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch,"A gated clock scheme for low power scan testing of logic ICs or embedded cores," in Proc. 10th Asian Test Symp., Nov. 2001, pp.253–258.

[10] C. Laoudias and D. Nikolos, "A new test pattern generator for high defect coverage in a BIST environment," in Proc. 14th ACM Great Lakes Symp. VLSI, Apr. 2004, pp. 417–420.

[11] S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy,"Low-power scan design using first-level supply gating," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 3, pp. 384–395, Mar.2005.

[12] X. Kavousianos, D. Bakalis, and D. Nikolos, "Efficient partial scan cell gating for low-power scan-based testing," ACM Trans. Design Autom. Electron. Syst., vol. 14, no. 2, pp. 28-1–28-15, Mar. 2009.

[13] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector inhibiting technique for low energy BIST design," in Proc. 17th IEEE VLSI Test Symp., Apr. 1999, pp. 407–412.

[14] S. Manich, A. Gabarro, M. Lopez, J. Figueras, P. Girard, L. Guiller,C. Landrault, S. Pravossoudovitch, P. Teixeira, and M. Santos, "Low power BIST by filtering non-detecting vectors," J. Electron. Test.-TheoryAppl., vol. 16, no. 3, pp. 193–202, Jun. 2000.

[15] F. Corno, M. Rebaudengo, M. Reorda, and M. Violante, "A new BIST architecture for low power circuits," in Proc. Eur. Test Workshop, May 1999, pp. 160–164.

[16] S. Gerstendorfer and H.-J. Wunderlich, "Minimized power consumption for scan-based BIST," in Proc. Int. Test Conf., Sep. 1999, pp. 77–84.

[17] A. Hertwing and H. Wunderlich, "Low power serial built-in self-test,"in Proc. Eur. Test Workshop, 1998, pp. 49–53.

[18] S. Wang and W. Wei, "A technique to reduce peak current and average power dissipation in scan designs by limited capture," in Proc. Asia South Pacific Design Autom. Conf., Jan. 2007, pp. 810–816.

[19] N. Basturkmen, S. Reddy, and I. Pomeranz, "A low power pseudorandom
BIST technique," in Proc. IEEE Int. Conf. Comput. Design:VLSI Comput. Process., Sep. 2002, pp. 468–473.

[20] S. Wang and S. Gupta, "LT-RTPG: A new test-per-scan BIST TPG for low switching activity," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 25, no. 8, pp. 1565–1574, Aug. 2006.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

[21] M. Nourani, M. Tehranipoor, and N. Ahmed, “Low-transition test pattern generation for BIST-based applications,” IEEE Trans. Comput., vol. 57,no. 3, pp. 303–315, Mar. 2008.

[22] X. Zhang, K. Roy, and S. Bhawmik, “POWERTEST: A tool for energy conscious weighted random pattern testing,” in Proc. 12th Int. Conf. VLSI Design, Jan. 1999, pp. 416–422.

[23] S. F. Q. Abdallatif and S. Abu-Issa, “Multi-degree smoother for low power consumption in single and multiple scan-chains BIST,” in Proc. 11th Int. Symp. Qual. Electron. Design, Apr. 2010, pp. 689–696.

[24] S. Chun, T. Kim, and S. Kang, “A new low energy BIST using a statistical code,” in Proc. Asia South Pacific Design Autom. Conf., Mar. 2008, pp. 647–652.

[25] B. Zhou, Y.-Z. Ye, Z.-L. Li, X.-C. Wu, and R. Ke, “A new low power test pattern generator using a variable-length ring counter,” in Proc. Qual. Electron. Design, Mar. 2009, pp. 248–252