LOW POWER TEST PATTERN GENERATION USING TEST-PER-SCAN TECHNIQUE FOR BIST IMPLEMENTATION

K. Jamal[1], P. Srihari[2], K. Manjunatha Chari[3] and B. Sabitha[1]
[1]Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India
[2]VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
[3]Gandhi Institute of Technology and Management University, Hyderabad, India
E-Mail: kjamal24@gmail.com

ABSTRACT

This paper introduces the function of test cases with minimal power for Built-In-Self-Test (BIST) implementation. This method intends Test-Per-Scan (TPS) based test cases using Multiple Single Input Change (MSIC) architecture. Multiple SIC patterns are developed by using EX-OR operation of twisted ring counter and test design algorithms like Linear Feedback Shift Register (LFSR),Bit-Swapping LFSR (BSLFSR), and Cellular Automata (CA). These patterns are used to a diminish number of transitions in the test patterns that are generated. The preferred method uses Test-Per-Scan technique for generating Multiple SIC test patterns. TPS diminished the power consumption during test mode. The seed generator used in TPS is modified LFSR's i.e., BS-LFSR, Cellular Automata (CA). BS-LFSR is composed of with an LFSR with a multiplexer. In CA, it also presents a variation on a BIST technique, which is from a one-dimensional cellular automaton; the pseudo random bit generator is generated. The proposed Hybrid Cellular Automata (HCA) using the rules 90 and 150 to generate the pseudo random designs. Moreover, the CA implementations illustrates properties of data compression like LFSRs and that they exhibit locally and with topological consistency significant attributes for a VLSI design. In this proposed method, LFSR is replaced with BS-LFSR, and HCA. Simulation and synthesis outcome with ISCAS c432 benchmark determine that Multiple SIC can reduce the power consumption.

Index Terms: BIST, MSIC, LFSR, BSLFSR, CA, HCA, circuit under test, test-per-scan, single input change, test response analyzer, pseudo random generator.

1. INTRODUCTION

Built-In-Self-Test scheme can adequately minimize the more complex VLSI analysis problems, by generating test hardware into the Circuit-Under-Test (CUT). The Linear Feedback Shift Register (LFSR) is generally exploited as Test Pattern Generators (TPGs) and Test Response Analyzers (TRAs) in traditional BIST technique. Amainsnag of these techniques is that the pseudorandom test cases produced by the LFSR causes a notably huge switching activity in the CUT, which can lead to enormous power dissipation and also blow the circuit and reduce the product yield. The LFSR generally requires very lengthy pseudorandom patterns in order to attain the required fault coverage in BIST implementation.

A. History work on BIST

There are a number of contrives that are used to generate design necessary for testing CUT. It has been founded that power consumption is more in test mode comparatively with normal mode [12]. The main idea behind low power techniques is to minimize the power consumption in test mode. Different kinds of test generation methods are required to develops table Built-In Self-Test (BIST) techniques. The utmost familiar test pattern design generation is based on pseudorandom pattern generators (PRPGs). The simple hardware on-chip test generation can be developed by pseudorandom tests patterns. Therefore, there are two major forms of PRPGs which is derived. Generally, the linear feedback shift registers and 1- Dimensional (1-D) Linear Hybrid Cellular Automata (LHCA) are major forms of PRPGs.

In spite of few coincidences, the series of states is consistently distinct between the LHCA and the LFSR, the LHCA can generates far good randomized test patterns [21]. The CA-based test generators will be an option to traditional LFSR algorithms. Further to meliorated randomization attributes, novel pseudorandom test design algorithms also have benefit in that they can be implemented for only contiguous neighbor communication and the physical length of the pattern generator. These can be elevated or diminished by only summate or deducting the cells. However, the investigation of aliasing function is a secondary controllable job for the CA than LFSR. The architecture in [7], presents Seeded Autonomous Circular Shift Register (SACSR) producing Single-Input-Change (SIC) patterns of maximum unique vectors. One of the ways to minimize power consumption is by reducing the transitions between the consequent patterns. Many techniques are introduced to minimize the transitions. The architecture in [3] presents Bit Swapping LFSR which is unlike from conventional LFSR reduces 33% of the transitions. BIST technique should generate test sequences with shallow power and area overhead and high fault coverage.

The architecture in [8] the introduced method has to decrease scan input bit transitions along operations of scan shifting. The architectures in [9], [11], [13] introduced various new techniques for reducing switching activities and also area overhead. The architecture in [2] introduces a new technique for generating the test designs with only single bit change compared with the previous patterns and generated using the XOR of the counter output with LFSR. The architecture in [5] power is reduced by rising the correlation between successive test patterns by introducing the intermediate designs between the consecutive patterns called Random Injection (RI) method.

The architecture LTRTPG in [4] introduces a design, which composed from a LFSR, k-input AND gate and a toggle flip-flop T-FF. The techniques used in this paper are to reduce the move at the input of the CUT. The approach used in [13] called DS-LFSR, uses two LFSRs (a normal speed LFSR and slow LFSR) working at different speeds.

The architecture in [1] introduces Test-Per-clock BIST using LFSR, BSLFSR, CA. The applicability of this test is confined only for test-per-clock BIST. It requires a lengthy sequence to get adequate fault coverage. This method is introduced in [18] used WRTPG to minimize power by withstanding the fault coverage. In this, an extra logic is introduced between CUT and TPG. The approach used in [19] is reordering of the test sets before applying them to the CUT. This reduces the switching activity between the consecutive sequences. Another category of reducing the power in Scan cell BIST is by changing the scan cells order in the scan chains [20].

2. MODIFIED PSEUDO RANDOM PATTERN GENERATORS

A. Cellular automata

CA is a computational design, to compute functions and solve algorithmic problems. The one-dimensional cellular automata existon a boundless horizontal array of cells. The one-dimensional cellular automata having square cells are only two probable states per cell: one is white and other is black. The CA's rules resolve the sequencing of the boundless arrangement of black and white cells which will bring up to date from time step to time step. A cellular automation generates in individual steps with the succeed value of one site determined by its preceded value and that set of sites termed as neighbor sites. The size of the neighborhood can vary, depending on the dimensionality of the CA. Figure-1 [24] depicts an example of one-dimensional CA, where the future value at a site depends on its present value and the values of the left and right neighbors.

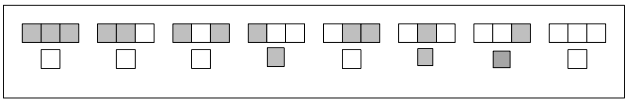

Figure-1. Example of one dimensional CA.

Cellular computing is promoted as one of the new prototypes for future computational systems because of three key properties: massive parallelism, simple, and local interconnect [22] to resolve the state of a cell in position z at time step t+1, expecting the states of cells in position z - 1, z, and z + 1 all in time step t. For each eight probable patterns of black and white cells, the state of cell y at time step t+ 1 is selected as blackelse white. In Figure-2 [24], for the probable eight input vectors, and also one probable output. In over all the total different probable outputs are 256[24].

In CA, n-dimensional Cellular Automata can be represented. In 1-D Cellular automata, next state of the cell is described on the preceding cell and its next

Analysis of Boundary conditions: To admit next state of the cell, consider extreme left and right cells. To extreme left cell, right neighbor is not present and vice versa with the extreme right cell. The right neighbor is not present to the extreme left cell, vice versa with the extreme right cell.

Boundary conditions are followed to analyze the next state of the cell. To overcome boundary conditions are followed to result the next state of the cell.

There are two types of boundary conditions [23].

Null boundary condition: In this boundary condition extreme right and left neighbors 0’s are included to recognize the next state of the cell; it is as illustrated in Figure-3. [23]

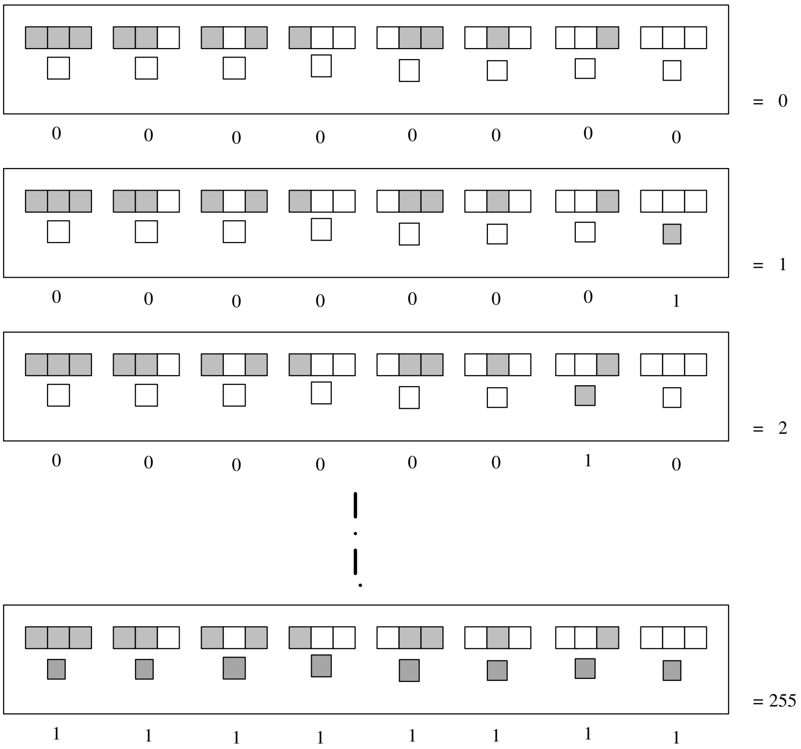

Figure-2. 8-bit input with 256 different possible outcomes.

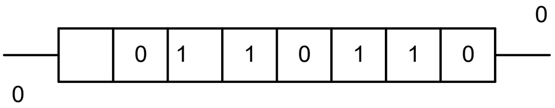

Figure-3. Null Boundary condition.

Cyclic boundary conditions: In these boundary conditions, the extreme right cell acts as neighbor to the extreme left and vice versa as illustrated in the diagram.

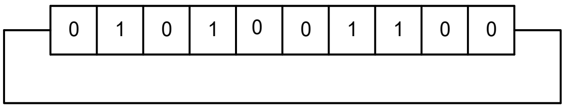

Figure-4. Cyclic boundary condition.

There may be two types of conditions mainly exists in the CA register, they are, null boundary Figure-3 [23] and cyclically connected Figure-4. One-dimensional CA's which contain two neighbor sites are only accepted; likewise, another modulus can be used. Each site will find out its neighbor site depending on the eight probable possibilities of the current site itself. The next-state sites related to each probable input site, and which is generates a number termed as "rule number" as per the Wolfram [24] classification.

B. Hybrid cellular automata

In HCA, by using the combination of 256 rules, the next state of the cell will be determined. For example, Rule 90 and Rule 150 generate better pseudo random test patterns. The features of a CA determined by coherent relations can relate a node to its neighbors. Rule 90 and Rule 150 are primary cellular automaton based on the EXOR function. It comprises of a 1-dimensional array of cells, for each one of node can hold either 0 or 1 value. In each time interval, all values are at the same instant interchanged by the EXOR of the two next values. Rule 90and Rule 150 falls under the Class III from random initial conditions. In this type of CA have shapes that rerun themselves, but their position and frequency is stochastic [22].

The Rule 90[25] states that if right or left next are black in the old step, then the new color of the cell to be black otherwise, the new color would be white[2] as shown in Figure-5. [22]

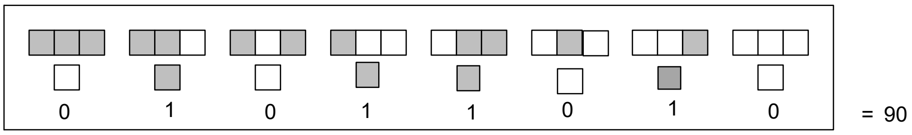

Figure-5. Rule 90.

Logic for Rule 90 is$Z_j = Z_{i+1} \oplus Z_{i-1}$

Rule 150 states that if there are odd number of blacks along with the cell and its neighbor, then the new state is black otherwise, the opposite color as shown in Figure-6. [22]

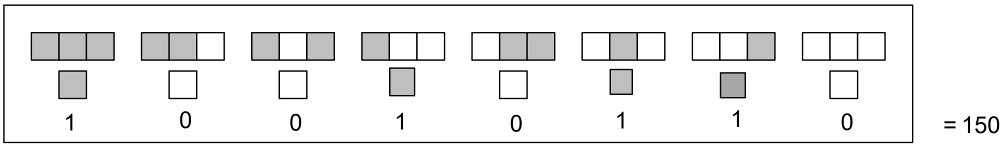

Figure-6. Rule 150.

Logic for Rule 150 is $Z_j = Z_{i+1} \oplus Z_i \oplus Z_{i-1}$

The succeeding stateZ$(t+1)$ of node $Z_J$is found by the present stateZ(t) of neighboring nodes$Z_{i+1}$and $Z_{i-1}$for rule 90. And nodesZ_i, $Z_{i+1}$and $Z_{i-1}$for rule 150. The similar rules cannot be carried out for all the nodes of a CA cell. Whereas, rules vary in accordance with the difference in nodes and its rules. The initial and the final nodes of a CA cell have only one neighbor node than all other nodes, which contains two; so general *rules* cannot be implemented here. Fixing missing next node at logic “0” is one solution (null boundary condition). Or final and initial nodes to be the next and it is associated by normal *rules* (called cyclic condition) is other solution. In cyclic boundary, the connection among the end nodes (the first and last nodes) precedes a feedback which creates a loop. This creates null boundary condition is a better one. The Figure-7 [22] depicts the implementation of a 4-bit CA register using rules 90 & 150 and null boundary condition [22].

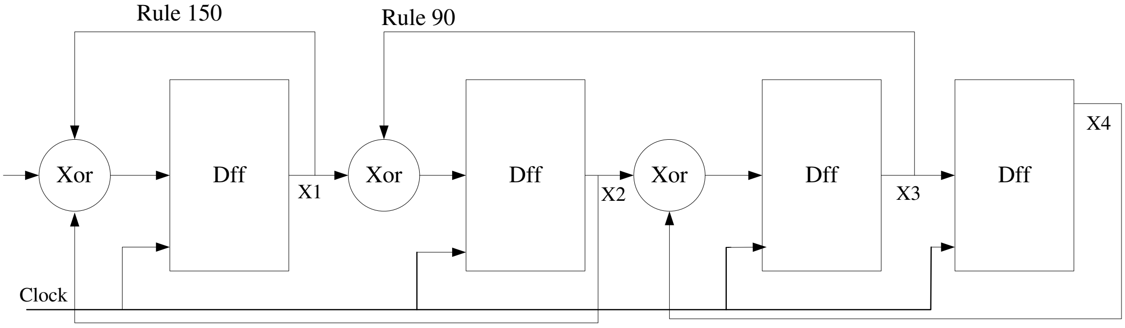

Figure-7. 4-bit HCA using rule 90 and 150,

C. Bit-swapping *LFSR*

Bit swapping LFSR (BSLFSR) is an altered LFSR, which includes Multiplexers with traditional LFSR. The BSLFSR generates test patterns with minimal number of transitions as compared with the traditional LFSR.

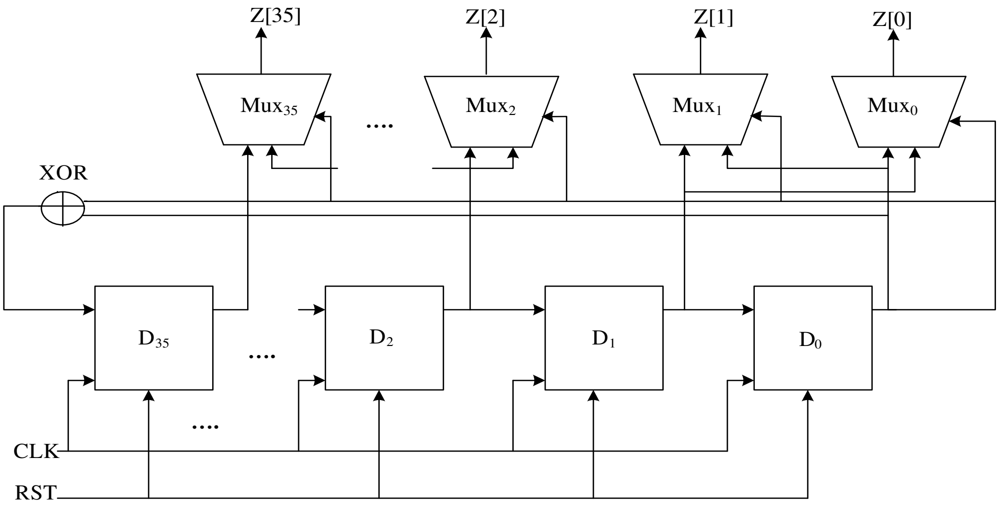

Figure-8. 36-bit BS-LFSR.

The Figure-8 [3] depicts the 36-bit BSLFSR. Asynchronous clock is given to a chain of D-Flip Flops. Multiplexer is used for swapping the bits return from the D-FFs. The selection line for all multiplexers is taken from the output of the last flip flop. Based on the last FF output, the BSLFSR test pattern Z[35:0] can be determined. The BSLFSR generated test patterns Z[35:0] is as it is the output of FFs, if lastFF output is ‘0’, or test patterns Z[35:0] is generated by swapping of adjacent FFs output. The number of transitions in standard LFSR is 2^{n-1}, where it is 2^{n-2} for BSLFSR.

3. PROPOSED APPROACH TEST OF GENERATING PATTERN SEQUENCES

A. Proposed TPS BIST architecture

Single Input Change (SIC) sequences are generated by Reconfigurable Twisted Ring Counter and Scalable SIC counter. Multiple SIC generator consists of reconfigurable twisted Ring counter, seed generator i.e., LFSR replaced with BS-LFSR and HCA, control circuit, clock signals and XOR network. Multiple SIC sequences are generated by XORing the output from modified seed generator and the reconfigurable twisted ring counter (RTRC). The application of modified LFSRs for generating test designs for the TPS BIST can reduce the number of transitions to 25% when compared with TPS using conventional LFSR.

The Proposed architecture using TPS is shown in Figure-9. TRA is a Test response analyzer which compares the output from MISR with golden response. If it matches, gives good response else faulty response.

BIST Controller controls all the blocks for proper operation. Figure-10 explains the operation of Test-Per-Scan configuration.

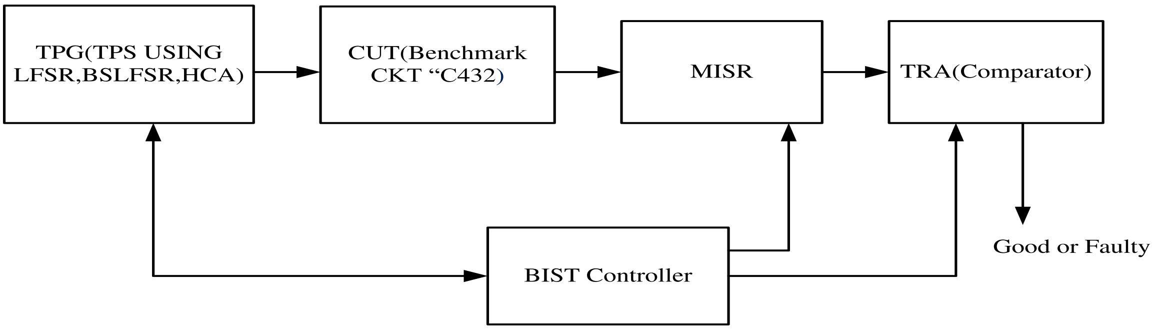

Figure-9. Proposed BIST architecture using TPS.

B. TPS test pattern generation

The Figure-10 depicts the Multiple SIC TPG using test-per-scan technique. The output from the modified seed algorithms and the counter are fed to the XOR gate. Here the counter is reconfigurable twisted ring counter. The return from the XOR gates is fed to the N scan chains and to the PI’s of the DUT.The steps are showing below:

a) The modified seed algorithm gives a seed (S) by triggering CLK1 one time at low frequency.

b) Set Mode M to “0”. For mode M=0, the twisted ring counter is work as twisted counter mode and gives a twisted vector (V) by triggering CLK2 one time.

c) Set Mode and start to “1”. The twisted ring counterworks as a circular shift register, and gets *n* codeword’s by triggering CLK2 *n* times. A appropriate operation is done.

d) Repeat the steps 2 and 3until 2*n* vectors (V) can be generated.

e) Repeat the steps 1 to 4 until the required test length or fault coverage is accomplished.

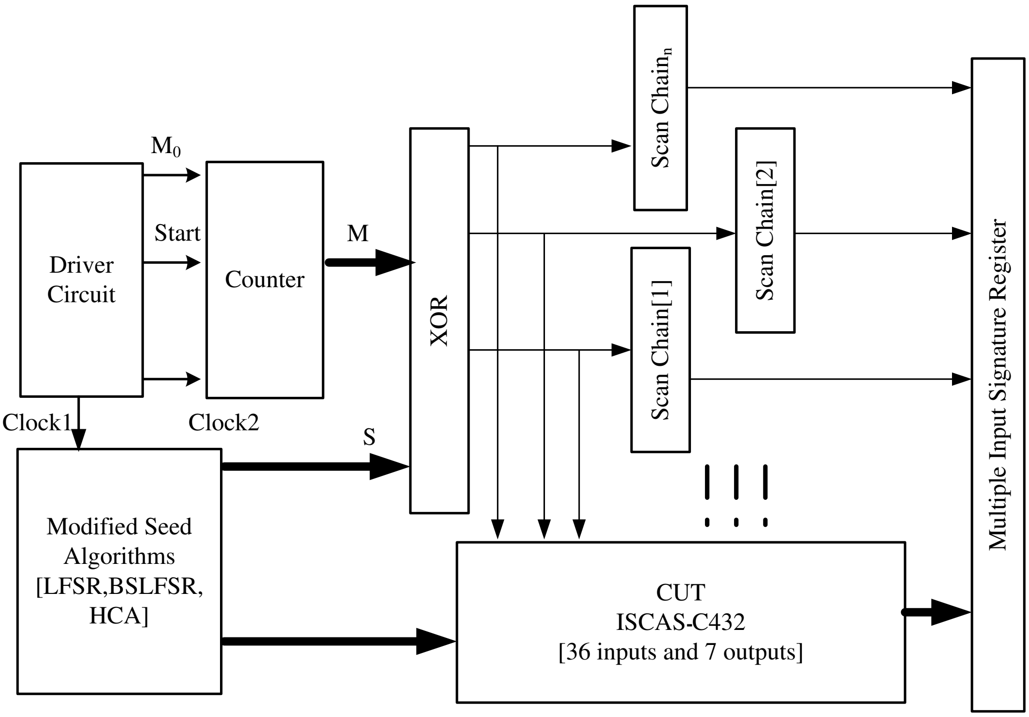
Driver Circuit
M0
Start
Counter
M
Clock1
Clock2
S
XOR
Scan Chainn
Scan Chain[2]
Scan Chain[1]
Modified Seed Algorithms [LFSR,BSLFSR, HCA]
CUT
ISCAS-C432
[36 inputs and 7 outputs]
Multiple Input Signature Register

Figure-10. Test-per-scan configuration.

C. Reconfigurable twisted ring counter

Multiple designs are generated by giving complement to the last bit of designs placed in first bit and every generated pattern can be XOR with seed generate. This process can be continued up to 2n cycles. Reconfigurable Twisted Ring Counter block diagram shown in Figure-10 [2]. In this counter, we have three modes of operations are shown below in Table-1.

Table-1. Modes of reconfigurable twisted ring counter.

Function	M	Start	Operation
Start	1	0	Counter is set to zero's by triggering clock more than l time
Operates as Circular shift	1	1	Imparts output by triggering clock *n* times
Operates as Normal	0	1	Generates 2*n* unique SIC test vectors by triggering clock 2*n* times

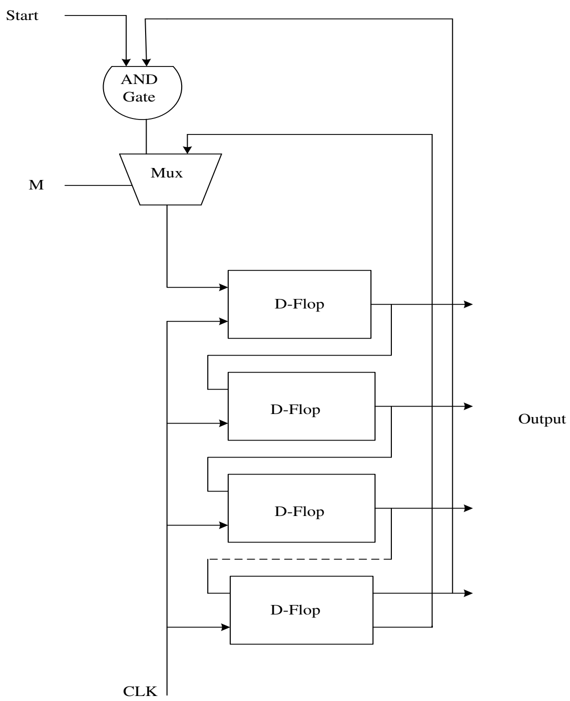

Figure-11. Reconfigurable twisted ring counter.

4. IMPLEMENTATION OF THE PROPOSED DESIGN

The implementation of test designs to CUT steps is shown in Figure-12. Design flow clearly depicts step by step procedure of further implementation applying pseudo random designs to the CUT to test and judge whether CUT is faulty or fault free. The used CUT is the standard benchmark circuits C432. C432 is the Interrupt controller and has 36 inputs and 7 outputs. The generated patterns are put on to the CUT. Return from the CUT is compressed in MISR and the signature generated from MISR is compared with the golden signature that is stored in the Test Response Analyzer. TRA is simply the comparator. The matching of signature with the golden signature determines whether the CUT that is tested fault or fault free.

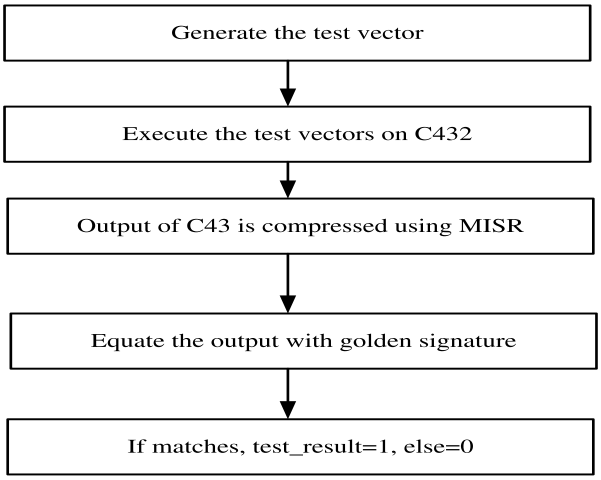
Generate the test vector
Execute the test vectors on C432
Output of C43 is compressed using MISR
Equate the output with golden signature
If matches, test_result=1, else=0

Figure-12. Design flow representation of the testing process.

5. RESULT ANALYSIS

The proposed TPGs are conducted on ISCAS c432 benchmark to analyze the power performance. The performance analysis is carried out with Cadence RTL Compiler. The test application method is TPS, for C432 benchmarks. A Table-2 summarizes about standard TPGs like standard LFSR, BSLFSR, HCA and also TPS based. The first row of the Table-2, bench mark circuit name is given. Constraints like area and power are described in the second row. In the consecutive rows explained about TPGs of top module area and power description. It is observed that dynamic power and Leakage Power is lower for HCA in TPS than conventional and other TPS TPG modules.

It is observed that the Leakage power and Dynamic power less for HCA compare with other TPGs using ISACS c432 CUT. While using with ordinary LFSR the dynamic power is 20553.614 µW, where as 21445.253 µW for BSLFSR and 754.843 µW for HCA shown in Table-2.

Table-2. Power analysis of standard TPGs (C432).

Benchmark		**C432**			
Constraints	**#**	**Area**	**Power**		
Seed algorithm	Modules	Area (Cell)	Power in µW (Leakage)	Power in µW (Dynamic)	Total Power in µW
TPS_LFSR	top	7002	4.001	20553.614	20557.616
TPS_BSLFSR	top	7108	4.097	21445.253	21449.351
TPS_ HCA	top	318	0.323	754.843	755.167

6. CONCLUSIONS

A Test Pattern Generator for MSIC Test-Per-Scan BIST using HCA and BSLFSR gives test cases which can diminish the switching activity in test application with minimal power consumption. This paper has proposed a low power TPG that could be low switching activity from one transition to the next. Experiment results demonstrate that the MSIC-TPS is consuming lesser power than the other TPGs.

REFERENCES

[1] K. Jamal, Dr. P. Srihari. 2016. Low Power TPC using BSLFSR. International Journal of Engineering and Technology. 8(2).

[2] Feng Liang, Luwen Zhang, Shaochong Lei, Guohe Zhang, Kaile Gao and Bin Lian. 2013. Test Patterns of Multiple SIC Vectors: Theory and Application in BIST Schemes. IEEE Transactions on Very large Scale Integration (VLSI) Systems. 21(4): 614-625.

[3] Abdallatif S. Abu-Issa and Steven F. Quigley. 2009. Bit-Swapping LFSR and Scan-Chain Ordering: A Novel Technique for Peak- and Average-Power Reduction in Scan-Based BIST. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 28(5): 755-759.

[4] Seongmoon Wang, Sandeep K. Gupta. LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation. ITC International Test Conference. pp. 5-94.

[5] Mehrdad Nourani, Mohammad Tehranipoor and Nisar Ahmed. 2008. Low-Transition Test Pattern Generation for BIST-Based Applications. IEEE Transactions on Computers. 57(3): 303-315.

[6] Dong Xiang, Yang Zhao, Krishnendu Chakrabarty and Hideo Fujiwara. 2008. A Reconfigurable Scan Architecture with Weighted Scan-Enable Signals for Deterministic BIST. IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems. 27(6): 999-1012.

[7] Lei Shaochong, HouXueyan, Shao Zhibiao and Liang Feng. 2008. A Class of SIC Circuits: Theory and Application in BIST Design. IEEE Transactions on Circuits and Systems- II: Express Briefs. 55(2): 161-165.

[8] Seongmoon Wang. 2007. A BIST TPG for Low Power Dissipation and High Fault Coverage. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 15(7): 777-789.

[9] Nan-Cheng Lai, Sying-Jyan Wang and Yu-Hsuan Fu. 2006. Low-Power BIST with a Smoother and Scan-Chain Reorder under Optimal Cluster Size. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 25(11): 2586-2594.

[10] Rajendra S. Katti, XiaoyuRuan and Hareesh Khattri. 2006. Multiple-Output Low-Power Linear Feedback Shift Register Design. IEEE Transactions on Circuits and Systems-I: Regular Papers. 53(7): 1487-1497.

[11] Seongmoon Wang and Sandeep K. Gupta. 2006. LT-RTPG: A New Test-Per-Scan BIST TPG for Low Switching Activity. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 25(8): 1565-1574.

[12] Patrick Girard. Survey of Low-Power Testing of VLSI Circuits. IEEE Design & Test of Computers. pp. 82-92.

[13] Seongmoon Wang, Member, IEEE and Sandeep K. Gupta. 2002. DS-LFSR: A BIST TPG for Low Switching Activity. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 21(7): 842-851.

[14] L. Avra, E.J. Mc Cluskey. 1993. Synthesizing for scan dependence in Built-in Self-Testable Designs. Proc. Int. Test Conf. pp. 734-743.

[15] Rosenberg: 1983. Exhaustive Generation of Bit Patterns with Applications to VLSI Self-Testing. IEEE Transactions on Computers. C-32(2): 190-194.

[16] P. H. Bardell, W. H. McAnney. 1984. Parallel Pseudorandom Sequences for Built-In Test. Proc. Int. Test Conf. pp. 302-308.

[17] P. Bardell, W. H. Mc Anney, J. Savir. 1987. Built-in Test for VLSI. Wiley-Interscience, New York.

[18] X. Zhang, K. Roy and S. Bhawmik. 1999. POWERTEST: A tool for energy conscious weighted random pattern testing. Proceedings of International Conference on VLSI Design. pp. 416-422.

[19] X. Kavousianos, D. Bakalis, M. Bellos and D. Nikolos. 2004. An efficient test vector ordering method for low power testing. Proceedings of IEEE Computer Society Annual Symposium on VLSI. pp. 285-288.

[20] M. Bellos, D. Bakalis, and D. Nikolos. 2004. Scan cell ordering for low power BIST. IEEE Computer Society Annual Symposium on VLSI pp. 281-284.

[21] WOLFRAM, S. 1986. Random Sequence Generation by Cellular Automata Advances in Applied Mathematics. 7(2): 123-169.

[22] Z. Barzilai, D. Coppersmith. 1989. Cellular Automata-based pseudorandom number generators for Built-in-Self-Test. IEEE Transactions on Computer Aided Design. 8(8).

[23] K.Salman. 2013. Analysis of elementary Cellular Automata Boundary Conditions. International journal of Computer Science & Information Technology (IJCSIT). 5(4).

[24] WOLFRAM S. 2002. A New Kind of Science. Champaign, IL: Wolfram Media. pp. 22-50.

[25] K. Jamal, Dr. P. Srihari. 2015. Analysis of Test Sequence Generators for Built-In Self-Test Implementation. 2nd International Conference on Advanced Computing and Communication Systems (ICACCS -2015, ISSN: 978-1-4799-6438-3/15/$31.00 ©2015 IEEE) Jan. 05-07.