Low Power TPC using BSLFSR

K Jamal #1 , Dr. P.Srihari #2
ECE Department
#1GRIET, Hyderabad, India
#2GCET, Hyderabad, India
1 kjamal24@gmail.com
2 mail2pshari@yahoo.com

***Abstract*— This paper propounds the role of test vectors with minimal power for Built-In-Self-Test (BIST) applications. This method signifies Test-Per-Clock (TPC) based test vectors using Multiple Single Input Change (MSIC). MSIC patterns are generated by using EX-OR operation of counter and test pattern algorithms like Linear Feedback Shift Register (LFSR), Bit-Swapping LFSR (BSLFSR), Hybrid Cellular Automata (HCA). These patterns are used to reduce number of transitions in the test patterns that are generated. The preferred method uses Test-Per Clock scheme for generating MSIC patterns. TPC reduces the power consumption during test mode. The seed generator used in TPC is modified LFSR's i.e., BS-LFSR and Cellular Automata (CA). Using CA we also present a variation on a Built-In-Self-Test (BIST) technique, which is based on a pseudo random number generator inferred from a one-dimensional cellular automata array. We proposed Hybrid Cellular Automata (HCA) using the rules 90 and 150 to generate the pseudo random patterns. In addition, it is noted that CA implementations exhibit data compression properties similar to the LFSRs and that they display locally and with topological regularity significant attributes for a VLSI implementation. In this proposed method, LFSR is replaced with BS-LFSR and CA. BS-LFSR is composed of an LFSR with a multiplexer. This reduces the number of transitions by 33.3% using MSIC whereas CA has more randomness compared with the LFSR. BSLFSR and CA are used in TPC.**

Keywords*— *BIST, MSIC, BS-LFSR, LFSR, TPC, Single Input Change (SIC), Circuit Under Test (CUT), CA, HCA, Output Response Analyzer (ORA)

I. INTRODUCTION

BIST is a DFT technique to solve highly complex VLSI testing problems. The Basic idea behind BIST is to test the VLSI chip itself. Typical BIST architecture is composed of three hardware modules in addition to the Circuit Under Test (CUT), Test Pattern Generator (TPG) gives the test patterns for the CUT. Output Response Analyzer (ORA) compares and studies to the test responses for determining correctness of the CUT. BIST controller is the central unit to ensure all the BIST functions including initialization and length of the BIST sequence.

A. Prior Work

There are number of schemes that are used to generate patterns necessary for testing CUT. It has been found that power consumption is more during test mode than in normal mode [11]. The main idea behind low power techniques is to reduce the power consumption during test mode than in normal mode. A variety of test generation scenarios are required to implement reliable Built-In Self-Test (BIST) schemes. The most common hardware generation is based on pseudorandom test pattern generators (PRPGs). Popularity of pseudorandom tests stems from very simple hardware required to implement on-chip test generation. Consequently, two principal forms of PRPGs which evolved over time and are now commonly in use comprise linear feedback shift registers (LFSRs) and 1- Dimensional (1-D) Linear Hybrid Cellular Automata (LHCA).

Despite some similarities, the sequencing of states is usually different between the LHCA and the LFSR, with the LHCA producing much better randomized test patterns [20]. These CA-based generators may be an alternative to conventional LFSR generators. In addition to better randomness properties these new pseudorandom test pattern generators also have implementation advantages in that they can be designed to require only adjacent neighbour communication and the physical length of the generator can be increased or decreased by only adding or subtracting cells. However, the analysis of aliasing performance is a much less tractable problem for the CA than for the LFSR. The architecture in [6], introduced Seeded Autonomous Circular Shift register (SACSR) generating Single-Input-Change (SIC) sequences of more unique vectors. One of the ways to reduce power consumption is by reducing the transitions between the consequent patterns. Many techniques are introduced to reduce the transitions. The architecture in [2] introduced Bit Swapping LFSR which is different from conventional LFSR reduces 33% of the transitions. BIST technique should generate test sequences with low power, low area overhead and high fault coverage.

The architecture in [7] introduced method has to decrease transitions that occur at scan inputs during scan shift operations. The architectures in [8], [10], [12] introduced various new techniques for reducing switching activities and also area overhead. The architecture in [1] introduces a new technique of generating the test patterns with only single bit change compared with the previous patterns and generated using the XOR of the counter output with LFSR. The architecture in [4] power is reduced by increasing the correlation between consecutive test patterns by introducing the intermediate patterns between the consecutive patterns called Random Injection (RI) method.

The architecture in [3] introduces a design, called low transition random test pattern generator (LT-RTPG), is composed from an LFSR, a k-input AND gate, and a toggle flip-flop T-FF. The techniques used in this paper are to reduce the transitions at the input of the CUT. The approach used in [12] called DS-LFSR, uses two LFSRs (a normal speed LFSR and slow LFSR) working at different speeds.

The applicability of this test is confined only for test-per-clock BIST. It requires a long sequence of test vectors to get adequate fault coverage. The method presented in [17] uses weighted random TPG to reduce power consumption while duly withstanding the fault coverage. In this, an extra logic is introduced between CUT and TPG. The approach used in [18] is reordering of the test vectors before applying them to the CUT. This reduces the switching activity between the consecutive sequences. Another category of reducing the power in Scan based BIST is by changing the order of the scan cells in the scan chains [19].

II. MODIFIED PSEUDO RANDOM GENERATORS

A. Cellular Automata

CA is a *c*omputational system, to compute functions and solve algorithmic problems. The one-dimensional cellular automaton exists on an infinite horizontal array of cells. For the purposes of this section we will look at the one-dimensional cellular automata (CA) with square cells that are limited to only two possible states per cell: white and black. The CA's rules determine the sequencing of the infinite arrangement of black and white cells which will be updated from time step to time step. A cellular automaton develops in discrete steps with the next value of one site determined by its previous value and that set of sites called the neighbour sites. The extent of the neighbourhood can vary depending, on other factors, upon the dimensionality of the CA under consideration. Fig. 1[23] illustrates a simple one-dimensional CA, where the next value at a site depends on its present value and the values of the left and right neighbours.

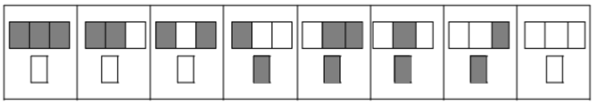

Fig.1.One Dimensional CA

Cellular computing is touted as one of the new prototypes for future computational systems due to three key properties: massive parallelism, simple, and local interconnect [21] to determine the state of a cell in position p at time step t+1, we expect at the states of cells in position p - 1, p, and p + 1 all in time step t. For each eight possible patterns of white and black cells, the state of cell p at time step t+ 1 is chosen as either black or white. In Fig. 2[23], for the eight possible input patterns, as well as one possible output. In all there are 256 different possible outputs [23].

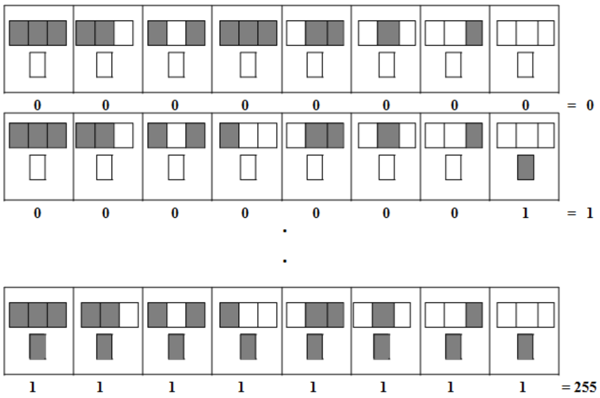

Fig.2. 256 different possible outcomes for 8 bit input

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

Analysis of Boundary conditions: To recognize next state of the cell, consider extreme left and right cells. To extreme left cell, right neighbour 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 analyze the next state of the cell.

There are two types of boundary conditions [22].

Null boundary condition: In this boundary condition extreme right and left neighbours 0’s are included to recognize the next state of the cell, it is as illustrated in Fig.3.[22]

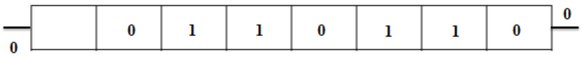

Fig.3.Null Boundary condition

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

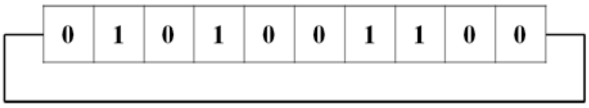

Fig.4.Cyclic Boundary condition

The CA register may possess null boundary conditions, as in Fig.3[22] (i.e., the first and last sites consider their missing neighbour site to constantly have a null value) or cyclically connected as in Fig. 4(i.e., the CA forms a ring thereby forming the first and last sites neighbours). Only binary one-dimensional CA’s with two neighbour sites (left and right) will be considered, however any desired modulus would be used. For a binary CA of this type each site must determine its next value on the basis of the eight possible combinations of the present values itself, and its left and right neighbours. The next-state values corresponding to each possible input form a number which is concerned to as the “rule number” under the classification scheme for Wolfram [23].

B. Hybrid Cellular Automata

In HCA, The next state of the cell is determined by using the combination of 256 rules. For example, Rule 90 and Rule 150 produce better pseudo random patterns. The logical relations which relate a node to its neighbours are known as rules and they define the features of a CA. Rule 90 and Rule 150 are elementary cellular automaton based on the EXOR function. It consists of a one-dimensional array of cells, each of which can hold either 0 or 1 value. In each time step all values are simultaneously interchanged by the EXOR of the two neighbouring values. Rule 90and Rule 150 falls under the Class III from random initial conditions. The CA’s in this class have shapes that repeat themselves, but their location and frequency is random [21]. This class contains about 4% of the basic CAs.

Rule 90[24] states that if right or left neighbour are black in the old step, then the new colour of the cell to be black otherwise, the new colour would be white[1] as shown in Fig.5.[21]

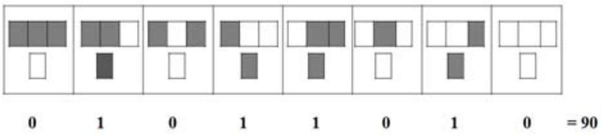

Fig.5 Rule 90

Logic for Rule 90 $X_j = X_{i+1} \oplus X_{i-1}$

Rule 150 states that if there are odd number of blacks along with the cell and its neighbour, then the new state is black otherwise, the opposite colour as shown in Fig.6.[21]

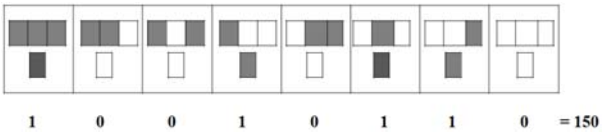

Fig.6 Rule 150

Logic for Rule 150 $X_j = X_{i+1} \oplus X_i \oplus X_{i-1}$

The next state x(t+1) of node x_i is determined by the current state *x(t)* of neighbouring nodes x_{i-1} and x_{i+1} for rule 90 and nodes x_i, x_{i-1} and x_{i+1} for rule 150. The same rule cannot be implemented for all the nodes of a CA register. Whereas, rules vary in accordance with the difference in nodes and its rules. The first and the last nodes of a CA register have only one neighbour unlike all other nodes will have two; hence general *rules* cannot be applied here. One solution is to assume that the missing neighbour is fixed at logic '0' (null boundary condition). The other solution assumes the last and first nodes to be neighbours and is connected using normal *rules* (cyclic condition). Connection between the end nodes (first and last nodes) precedes a feedback loop in the cyclical boundary status; this makes null boundary condition a better choice. Fig.7 [21] shows the construction of a 4-bit CA register using rules 90 & 150 and null boundary condition [21].

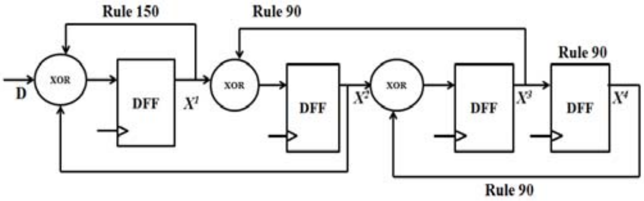

Fig.7. 4-bit HCA using rule 90 and 150

C. Bit Swapping LFSR

Bit swapping LFSR is modified LFSR. Addition of Multiplexer along with conventional LFSR constitutes BS-LFSR. The numbers of transitions are reduced when compared with the conventional LFSR.

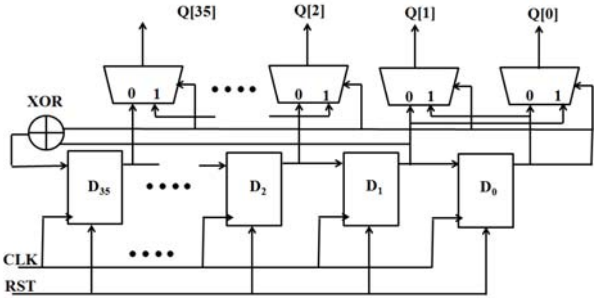

Fig.8. 36-bit BS-LFSR

The Fig.8. [2] shows the 36-bit BS-LFSR.A common clock signal is given to a series of Flip Flops as control signal. Multiplexer is used for swapping the outputs of the D-FFs. The output of the last FF is taken as the selection line for all the FFs. If the output of the last FF is '0',then the output Q[35:0] is same as the output of all the FFs. If the output of the last FF is '1', then swapping of the outputs of the adjacent FFs is done. In conventional LFSR, the number of transitions would be $2^{n-1.}$In BSLFSR, the number of transitions are 2^{n-2}.

III. PROPOSED APPROACH TEST OF GENERATING PATTERN SEQUENCES

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, control circuit, clock signals and XOR network. MSIC sequences are generated by XORing the output from the seed generator and the reconfigurable twisted ring counter. The application of modified LFSRs for generating test patterns for the TPC BIST can reduce the number of transitions to 25% when compared with TPC using conventional LFSR.

A. TPC Test Pattern Generation:

There are *n primary* inputs (PIs) and *N* scan chains in a full scan design, and each scan chain has *l* cells. The vector generated by an *n*-bit LFSR with the primitive polynomial can be expressed as $C(t) = C_0(t)C_1(t)C_2(t)$, . . . , $Cn-1(t)$ (here in after referred to as the seed), and the vector generated by an *l*-bit Twisted Ring counter can be expressed as $K(t) = K_0(t)K_1(t)K_2(t)$, . . . , $K_{l-1}(t)$.In the first clock cycle, $K = K_0K_1K_2$, . . . , K_{l-1} will bit-XOR with $C = C_0C_1C_2$, . . . C_{N-1}, and the results Y_1Y_{1+l},Y_{2+l} . . , $Y_{(N-1)l+1}$ will be shifted into *N*scan chains, respectively. In the second clock cycle, $K= K_0K_1K_2$, . . . , K_{l-1} will be circularly shifted as $K = K_{l-1}K_0K_1$, . . . , K_{l-2}, which will also bit-XOR with the seed $C = C_0C_1C_2$, . . . , C_{N-1}. The resulting$Y_2Y_{l+2}Y_{2l+2}$, . . . , $Y_{(N-1)l+2}$ will be shifted into *N* scan chains, respectively. After *l* clocks, each scan chain will be fully loaded with a unique Twisted ring codeword, and seed $C_0C_1C_2$, . . . , $Cn-1$ will be applied to *n* PIs[11] .

B. MSIC TPC method

The MSIC-TPG for test-per-clock scheme is shown in Fig. 9.[1] The output of the LFSR is XORed with the output of the twisted ring counter. The operation is as follows:

- The LFSR generates a new seed by clocking Clock1one time.
- The Twisted ring counter generates a new vector by clocking Clock2 one time.
- Repeat 2 until 2*n* Twisted ring vectors are generated.
- Repeat the above steps till fault target is reached.

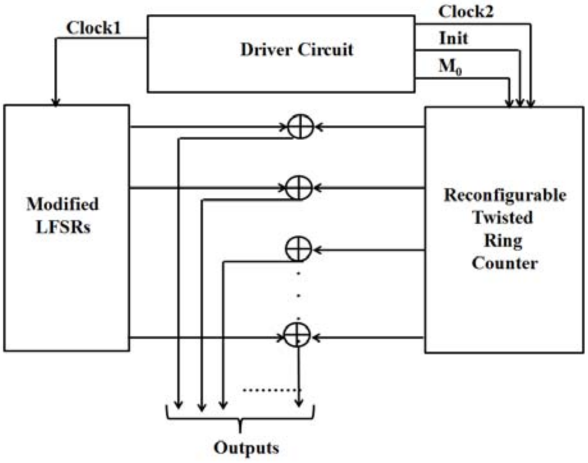

Fig.9 Test–per–clock configuration

C. Reconfigurable Twisted Ring Counter

Multiple patterns are generated by giving complement to the last bit of patterns 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 Fig.10.[1] In this counter, we have three modes of operations are shown below in Table I.

TABLE-I. Modes of RTRC

Mode	M_0	Start	Operation
Start	1	0	Counter is initialized to all 0's by clocking more than n times
Circular shift	1	1	Gives output code by clocking n times
Normal	0	1	2n unique SIC vectors by clocking 2n times

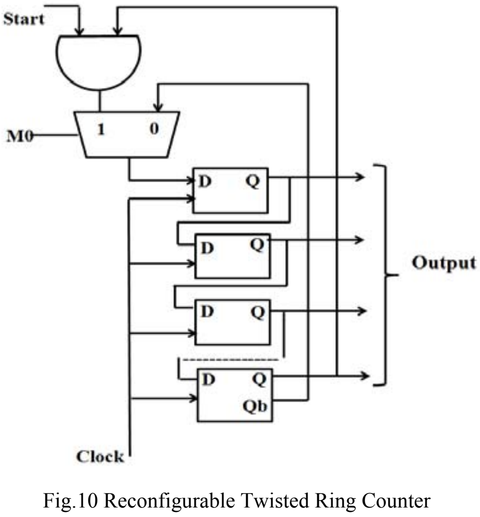

Fig.10 Reconfigurable Twisted Ring Counter

IV. IMPLEMENTATION OF THE PROPOSED DESIGN

The implementation of test patterns to CUT steps is shown in Fig.11. Design flow clearly depicts step by step procedure of further implementation applying pseudo random patterns to the CUT to test and judge whether CUT is faulty or fault free.

The used CUT is the standard benchmark circuits C432 and C3540. C432 is the Interrupt controller and has 36 inputs and 7 outputs. The generated Patterns are applied to the CUT. The output from the CUT is compressed in MISR. 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. C3540 is the Arithmetic Logic Unit having 50 inputs and 22 outputs.

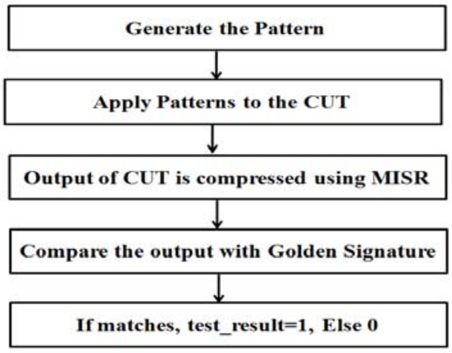
Generate the Pattern
Apply Patterns to the CUT
Output of CUT is compressed using MISR
Compare the output with Golden Signature
If matches, test_result=1, Else 0

Fig.11 Design Flow representation of the testing process

Fig.12 gives the Flowchart representing the control signals that are used in the testing process. The below flow chart depicts the states (idle, running, compare) and signals that are used for the implementation. When Start=0, the process is in idle state else it is in the running state. The BIST Controller is in running state until the counter counts the clock pulses reaches 1000.If the count =1000 clock pulses ,then the controller goes to the Compare state. After the comparison, the signal done is high. If the output of the MISR matches with the golden signature then the result is high and signifies that the CUT is fault free.

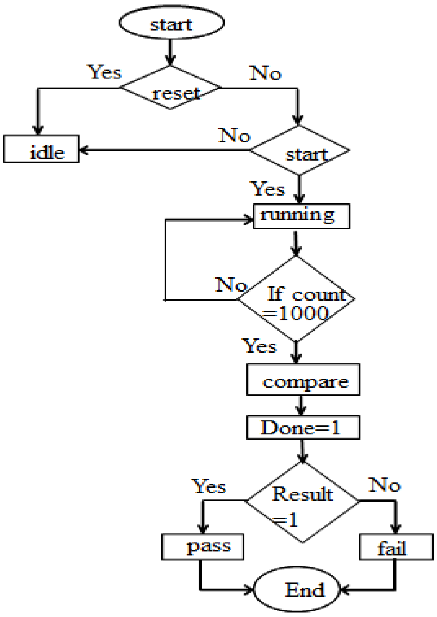

Fig.12 Flowchart representing the implementation of the signals used

V. RESULT ANALYSIS

The proposed TPGs are conducted on ISCAS benchmark to analyze the power performance. The performance simulations are carried out with Cadence NCSim and RTL Compiler. Synthesis is carried out with 180-nm technology. The test application method is TPC, for C432 and C3540 benchmarks. A table summarizes about standard TPGs like standard LFSR, BSLFSR, HCA and also TPC based. The first row of the Table II, III, IV and V, bench mark circuit name is given. Constraints like area and power described in the second row. In the consecutive rows explained about TPGs of each and every modules area and power description.

TABLE-II. Power Analysis of standard TPGs (C432)

Benchmark	**C432**					
Constraints	#	**Area**	**Power**			
#	**Modules**	**Cell Area**	**Leakage (nW)**	**Internal (nW)**	**Net (nW)**	**Switching (nW)**
LFSR	TOP	6759.24	195.68	862385.18	139056.98	1001442.16
	BC	1736.38	47.37	169069.96	9449.21	178519.17
	CUT	1054.47	27.01	28045.67	26045.12	54090.79
	LFSR	3299.79	104.34	550425.97	27694.94	578120.91
	MISR	615.38	15.61	114804.03	4419.11	119223.14
	TRA	53.22	1.36	39.55	32.12	71.67
BSLFSR	TOP	10548.01	296.16	1233313.31	203091.65	1436404.97
	BC	1736.38	47.37	174555.30	13087.14	187642.45
	CUT	1051.14	28.60	21844.28	21314.37	43158.65
	BSLFSR	7095.21	203.16	928927.19	51987.72	980914.91
	MISR	615.38	15.61	107604.02	3648.19	111252.21
	TRA	49.90	1.42	382.52	87.48	470
HCA	TOP	7028.68	192.52	901526.37	145868.12	1047394.49
	BC	1736.38	47.37	179186.34	13477.39	192663.73
	CUT	1054.47	27.01	20167.59	19857.28	40024.86
	HCA	3572.55	101.01	593664.12	41237.25	634901.37
	MISR	615.38	15.61	108211.22	4231.25	112443.07
	TRA	49.90	1.52	297.10	98.42	395.51

We find that dynamic power is less for BSLFSR compare with other TPGs using ISCAS C432 CUT. While using with ordinary BSLFSR the switching power is 43158.65 nW where as 54090.79 nW for LFSR and 40024.86 nW for HCA. Table II summarizes the power comparison C432 using TPC shown in Table II.

TABLE-III. Power Analysis of TPC TPGs (C432)

Benchmark	**C432**					
Constraints	#	**Area**	**Power**			
#	**Modules**	**Cell Area**	**Leakage (nW)**	**Internal (nW)**	**Net (nW)**	**Switching (nW)**
TPC (LFSR)	TOP	10251.96	340.67	577025.38	111530.85	688556.23
	BC	1679.83	50.95	66954.13	4013.83	70967.96
	CUT	1027.86	26.52	26587.90	25861.28	52449.17
	TPC	6805.81	244.27	399141.95	37976.57	437118.52
	MISR	615.38	15.61	79362.32	5216.00	84578.31
	TRA	46.57	1.32	213.99	100.47	314.45
TPC (BSLFSR)	TOP	13901.03	429.33	696688.99	141834.47	838523.46
	BC	1673.18	51.22	68518.76	4812.08	73330.84
	CUT	1054.47	27.01	20167.59	19857.28	40024.86
	TPC	10461.53	333.05	524558.39	67011.73	591570.12
	MISR	615.38	191.32	229876.15	39547.79	269423.94
	TRA	46.57	1.51	201.65	26.65	94.41
TPC (HCA)	TOP	7028.68	192.52	901526.37	145868.12	1047394.49
	BC	1736.38	47.37	179186.34	13477.39	192663.73
	CUT	1027.86	26.52	22757.06	24042.65	46799.71
	TPG	3572.55	101.01	593664.12	41237.25	634901.37
	MISR	615.38	15.61	108211.22	4231.84	112443.07
	TRA	49.90	1.52	297.10	98.42	395.51

Dynamic power for TPC BSLFSR for ISCAS C432 CUT is 40024.86. While using with TPC BSLFSR the switching power is 40024.86 nW where as 52449.17 nW for LFSR and 46799.71 nW for HCA. Table II summarizes the power comparison C432 using TPC shown in Table III.

We observe that dynamic power is less for BSLFSR compare with other TPGs using ISCAS C3540 CUT. While using with ordinary BSLFSR the switching power is 190556.04 nW where as 245885.84 nW for LFSR and 232344.84 nW for HCA shown in Table IV.

TABLE-IV Power Analysis of standard TPGs (C3540)

Benchmark	**C3540**					
Constraints	#	**Area**	**Power**			
#	**Modules**	**Cell Area**	**Leakage (nW)**	**Internal (nW)**	**Net (nW)**	**Switching (nW)**
LFSR	TOP	22865.67	738.63	488740.85	155630.34	644371.19
	BC	1812.89	50.09	42890.60	10915.18	53805.78
	CUT	14742.60	499.95	158021.06	87864.78	245885.84
	LFSR	4194.59	128.54	199521.55	38634.04	238155.59
	MISR	1962.58	55.67	88097.95	5471.60	93569.55
	TRA	153.01	4.38	209.69	101.15	310.84
BSLFSR	TOP	27722.22	863.32	555893.67	154357.09	710250.77
	BC	1812.89	50.09	36316.89	8051.58	44368.47
	CUT	14742.60	499.95	129160.80	61395.24	190556.04
	BSLFSR	9044.48	252.29	316338.16	63337.57	379675.73
	MISR	1962.58	55.67	73322.56	4591.33	77913.89
	TRA	159.67	5.32	755.27	198.88	954.15
HCA	TOP	23614.11	751.33	510789.94	166452.57	677242.51
	BC	1812.89	50.09	46003.21	11562.40	57565.61
	CUT	14742.60	499.95	151698.53	80646.31	232344.84
	TPG	4946.36	141.58	231534.78	61887.32	293422.09
	MISR	1962.58	55.67	81094.76	5377.29	86472.05
	TRA	149.69	4.04	458.66	232.37	691.03

Dynamic power for TPC BSLFSR for ISCAS C432 CUT is 40024.86. While using with TPC BSLFSR the switching power is 517502.36 nW where as 1195718.61 nW for LFSR and 634590.64 nW for HCA. Table II summarizes the power comparison C432 using TPC shown in Table V.

TABLE-V Power Analysis of TPC TPGs (C3540)

Benchmark	**C3540**					
Constraints	#	**Area**	**Power**			
#	**Modules**	**Cell Area**	**Leakage (nW)**	**Internal (nW)**	**Net (nW)**	**Switching (nW)**
TPC (LFSR)	TOP	19991.66	679.20	1148645.29	543435.67	1692080.96
	BC	1719.75	49.50	15523.93	1842.55	17366.48
	CUT	14742.60	499.95	748865.30	446853.31	1195718.61
	TPC	1330.56	67.19	50057.40	77574.94	127632.34
	MISR	1962.58	55.67	332618.44	14028.58	346647.02
	TRA	159.67	4.55	578.58	235.79	814.36
TPC (BSLFSR)	TOP	32522.21	1045.35	740039.71	207452.67	947492.37
	BC	1693.14	51.30	20660.58	6733.23	27393.81
	CUT	14742.60	499.95	188397.52	95150.22	283547.74
	TPC	13901.03	431.20	433401.96	84100.40	517502.36
	MISR	1962.58	55.67	95837.78	5938.39	101776.17
	TRA	146.36	4.90	740.23	347.19	1087.41
TPC (HCA)	TOP	18621.19	610.07	639162.76	300544.37	939707.14
	BC	1753.01	49.85	42455.50	9604.35	52059.85
	CUT	14742.60	499.95	390201.59	244389.05	634590.64
	TPG	13951.03	451.20	473424.96	37066.92	37066.92
	MISR	1962.58	55.67	206272.01	9232.56	215504.57
	TRA	162.99	4.60	233.66	191.36	425.02

VI. CONCLUSION

A TPG for test-per-clock BIST using HCA and BSLFSR generates test patterns which can reduce switching activity during test application which reduces 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-TPC is taking low power than the other TPGs.

REFERENCES

[1] Feng Liang, Luwen Zhang, Shaochong Lei, Guohe Zhang, Kaile Gao, and Bin Lian- “Test Patterns of Multiple SIC Vectors: Theory and Application in BIST Schemes”, IEEE Transactions on Very large Scale Integration (VLSI) Systems, Vol. 21, No. 4, April 2013,PP 614-625.

[2] Abdallatif S. Abu-Issa and Steven F. Quigley “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, Vol. 28, No. 5, May 2009, PP 755-759.

[3] Seongmoon Wang, Sandeep K. Gupta “LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation” ITC International Test Conference, PP 85-94.

[4] MehrdadNourani, Mohammad Tehranipoor, and Nisar Ahmed, “Low-Transition Test Pattern Generation for BIST-Based Applications” IEEE Transactions on Computers, Vol. 57, No. 3, March 2008, PP 303-315

[5] Dong Xiang, Yang Zhao, Krishnendu Chakrabarty, and Hideo Fujiwara “A Reconfigurable Scan Architecture With Weighted Scan-Enable Signals for Deterministic BIST” IEEE Transactions on

[6] Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No. 6, June 2008, PP- 999-1012

[7] Lei Shaochong, HouXueyan, Shao Zhibiao, and Liang Feng “A Class of SIC Circuits: Theory and Application in BIST Design” IEEE Transactions on Circuits and Systems- II: Express Briefs, Vol. 55, No. 2, February 2008, PP- 161-165

[8] Seongmoon Wang “A BIST TPG for Low Power Dissipation and High Fault Coverage”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 7, July 2007, PP-777-789

[9] Nan-Cheng Lai, Sying-Jyan Wang, and Yu-Hsuan Fu “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, Vol. 25, No. 11, November 2006, PP- 2586-2594

[10] Rajendra S. Katti,XiaoyuRuan, and HareeshKhattri, “Multiple-Output Low-Power Linear Feedback Shift Register Design” IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol. 53, No. 7, July 2006, PP- 1487-1497

[11] Seongmoon Wang and Sandeep K. Gupta “LT-RTPG: A New Test-Per-Scan BIST TPG for Low Switching Activity”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 8, August 2006, PP-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 “DS-LFSR: A BIST TPG for Low Switching Activity” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, No. 7, July 2002, PP- 842-851

[14] L.Avra, E.J.McCluskey: "Synthesizing for scan dependence in Built-in Self-Testable Designs", Proc. Int. Test Conf., 1993, PP. 734-743

[15] Rosenberg: "Exhaustive Generation of Bit Patterns with Applications to VLSI Self-Testing", IEEE Transactions on Computers, Vol. C-32, No. 2, Feb. 1983, PP. 190-194

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

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

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

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

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

[21] WOLFRAM,S.“Random Sequence Generation by Cellular Automata, Advances in Applied Mathematics. Volume 7, Issue 2, June 1986, Pages 123-169.

[22] Z. Barzilai, D. Coppersmith, ‘Cellular Automata-based Pseudorandom number generators for Built-in-Self-Test’, IEEE Transactions on Computer Aided Design, Vol 8, No.8, August 1989.

[23] K.Salman, “Analysis of elementary Cellular Automata Boundary Conditions,” International journal of Computer Science & Information Technology (IJCSIT) vol 5, No.4, August 2013.

[24] WOLFRAM, S.: ‘A New Kind of Science’. Champaign, IL: Wolfram Media, 2002 page no.22- 50.

[25] K.Jamal, Dr.P.Sri hari “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, 2015