International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7537-7551

http://www.ripublication.com

Deterministic Seed Selection and Pattern Reduction in Logic BIST

Ramesh Bhakthavatchalu[1] and Nirmala Devi M.[2]

Electronics and Communication Engineering,
Amrita Vishwa Vidyapeetham, Coimbatore, India.
[1]rameshb@am.amrita.edu, [2]m_nirmala@cb.amrita.edu

Abstract

A technique to select the proper seed and to reduce the number of test patterns generated in Logic Built in Self Test (BIST) is proposed. This paper explains the algorithm that can be used offline of BIST flow to search and classify the random patterns based on the deterministic test patterns generated by the Automatic Test Pattern Generator (ATPG). The seed activated Linear Feedback Shift Register (LFSR) generates exhaustive test patterns which are applied on any Circuit Under Test (CUT). The responses are received at the output of the scan chains in the CUT and they are compressed to produce a signature. This signature is compared with the expected golden signature to indicate the BIST status. It is shown that this scheme produces the same fault coverage with lesser number of random test patterns compared to an arbitrary seed. This technique is well suited for any scan based sequential design. It is applied on ISCAS-89 designs with the help of Cadence Encounter Test Architect 13.1 tool. The results show that this method is comparable with similar methods. Possible limitations of this technique when employed in large designs and solutions are also suggested as future work.

Keywords LBIST, random patterns, LFSR, seed selection, scan chain, ATPG, MISR, DFT, Deterministic Test.

1. Introduction

Logic built in self test (BIST) [1, 2, 3] is considered as the suitable testing technique for current day VLSI (Very Large Scale Integration) designs. It is a design technique where any electronic hardware unit, chip or circuit can be made to test itself by embedding a small extra circuitry in to it. With process technologies shrinking in to the nanometer regions, the defect levels of the manufactured devices are constantly growing. Conventional Automatic Test Equipment (ATE) based testing method is no

longer able to handle the ever-growing test challenges. Logic BIST is widely getting adopted as the testing technique for most current day scan based designs. Logic BIST does not alter the scan structure of the designs permitting them to have both ATE based testing and also Logic BIST.

In a typical Logic BIST architecture [4], an on-chip pattern generator usually a LFSR generates the random patterns. These patterns are applied in a serial fashion in to the scan chains of the *design under test* (DUT). A response compactor, usually a *multiple input signature register* (MISR) compacts the responses from the design under test. A ROM memory unit stores the expected golden response. A comparator is present to compare the golden response with the circuit's output response. After a pre-determined number of clock cycles, after comparison of the signatures the comparator indicates whether the DUT has passed the BIST test or not [5]. A controller circuit co-ordinate the overall operations [6].

Logic BIST test structures are embedded in to the integrated circuit (IC) with the design unit, thus, adding a small permissible overhead to the area of the IC[1]. The nature of vectors in Logic BIST are usually pseudo random and so even for a moderately sized design, several thousands of patterns are to be generated in the Logic BIST compared to a few hundreds of deterministic test patterns in ATPG to achieve adequate fault coverage. So, methods to improve the fault coverage of Logic BIST by increasing the pattern efficiency were constantly explored.

Alternate methods to provide high fault coverage with less area and pin overhead and without degrading performance were also explored.

Touba, and McCluskey[7] explored the random pattern generation using mapping techniques. Higher fault coverage can be achieved by the combination of Logic BIST with top-up ATPG [8]. ATPG and fault simulation at the Hardware Description Language (HDL) level in the design phase was proposed by Chen and Noh. In [10], a hierarchical test scheme with heterogeneous core test and test access methods has been devised. In [11], allocating the timing faults for testing in different intra/inter clock domains when the maximum number of applied BIST test patterns is specified. Reconfigurable Logic BIST up to 32 bits was explored in [12]. In [13] a programmable BIST with reseeding and weighted pattern generation were discussed. A system level Logic BIST implementation is discussed in [14]. Andal et al. [15] gives a methodology for Logic BIST diagnosis. Accumulator based weighted test pattern generation is discussed in [16]. [17] described a technique to compress the test vector volume using bit fixing methods. A full scan BIST solution by unscanning some scan flip-flops is proposed in [18]. In [19] and [20], presented a pattern reduction method based on seed encoding and reseeding using ATPG patterns in between the random patterns. The objective of all the above research is to increase the efficiency of the BIST process. In common, the efficiency of a BIST process is assumed as the test coverage achieved with smallest test set possible without area or power overhead. All the previously proposed techniques involve either design or BIST circuitry modifications [21]. In order to achieve the specified objective, this paper discusses a technique for the judicial selection of the seed (starting state) of the on-chip pattern generator unit based on the ATPG patterns.

The proposed method uses STUMPS (Self-Test Using a MISR and Parallel Shift register) architecture which is widely used in practice[4]. In STUMPS architecture, it is proposed to have the primary inputs and scan inputs of the circuit are fed by separate LFSR's (*linear feedback shift register)* to have better pattern controllability of the shift and capture cycles. A parallel LFSR structure is used as the Pseudo Random Pattern Generator (PRPG) to feed the scan inputs and primary inputs of the design which in turn reducing the BIST area compared to a serial LFSR structure.

2. LFSR based Pseudo Random Pattern Generation :

Consider a Multi-Input Multi-Output (MIMO) combinatorial circuit with '*n*' inputs and '*m*' outputs. There exists a number '*w*' such that "*w* < *n*" where '*w*' is the maximum number of inputs, the outputs of the circuit depend upon. The MIMO circuit is called as *(n,w)* CUT (circuit under test), where *n>w>0*. This MIMO cicuit can be tested exhaustively by giving all possible combinations of the *'w'* inputs rather than the *'n'* inputs. As shown below an LFSR can be designed in such a way to produce the '2^w' combinations of the inputs.

Design a *(p,y)* LFSR in such a way that it is an '*p*' stage autonomous LFSR of period 2^y-1 such that (*y* < *p*). The problem of exhaustive testing of the (*n,w*) CUT is to obtain the smallest integer *y* such that any *w* combinations of the *(p,y)* LFSR outputs contain all 2^w distinct test patterns (states) where $w \leq y < n$.

Theorem 1 in [22] shows that the *(p, y)* LFSR, with the smallest integer *y* is capable of exhaustively testing the CUT provided the following two conditions are met.

1) The characteristic polynomial of the LFSR is of the form:

$$f(x) = g(x)\, p(x) = (1 + x + x^2 + \ldots + x^{n-y})\, p(x) \qquad \ldots(1)$$

where p(x) is a primitive polynomial of degree 'k' and g(x) is any monic polynomial of degree 'n – k'.

2) The initial state $S_o(x)$ of the LFSR is divisible by g(x).

In the on-chip testing scenario, it is comparatively easy to generate pseudo random pattern generator circuits than deterministic pattern generator circuits. It is because generation of deterministic patterns requires design of highly complex digital circuits adding a tremendous overhead on area, size and complexity.

Extending the application of the above theorem the following concept for SOC testing can be formulated.

Let A be the exhaustive set of pseudo random patterns generated by a Pseudo Random Pattern Generator (PRPG). Let B be a subset of A generated deterministically. Further, for a given CUT, let A_i be a set of pseudo random patterns generated by some PRPG. Note that in general generating the patterns of set B involves complex circuitry. However, the generation of A_i's with modified PRPGs (of manageable complexity) has been demonstrated.

What was needed is to breakdown the enormous complexity of the probelm of selecting the useful random patterns in to simpler functions. *The main problem addressed in this article is:* Given a set A of 2^n binary strings with n bits and B a subsequence of A with cardinality k where $'0< k < 2^n$. Let $'A_1, A_2, A_3, \ldots'$, be unique subsets of A, where each A_i is a superset of B. We are interested in selecting A_i with least cardinality. Note that each A_i is a set of pseudo random patterns generated by some PRPG.

This paper proposes an algorithm that attempts to identify the smallest subset of patterns (i.e. A_i for some i) that can be generated by PRPG that does not require circuit modification. Further, the generated patterns cover B entirely (except when A is $>>2^{32}$), thus, achieving the same fault coverage of the corresponding ATPG. In practice, the A_i's that were produced using different pattern compression techniques proposed to cover B do not cover it 100%.

3. Proposed Approach

A technique to reduce the number of random test patterns by properly selecting the starting seed is explained below. This approach breaks down the process into three phases: in the first phase, the test vectors for the design is generated using ATPG. In the second phase an exhaustive set of pseudo random patterns is generated with a non-zero value as the initial seed. In the third phase, a heuristic approach is made by comparing the small number of ATPG patterns and the large set of random patterns. A new optimized subset of the random patterns is generated and the starting pattern is selected as the 'seed' for this new optimized subset. This subset comprises all the ATPG test patterns.

Unlike ATPG, the number of primary inputs and outputs do not limit the number of scan chains that can be implemented in logic BIST. Since the proposed technique uses the data from the ATPG to compress the size of BIST patterns, it is considered that the BIST and ATPG are having the same number of scan chains.

3.1. Significance of Seed Selection

The non-zero initial state of the LFSR flip-flops is called the seed value. An LFSR produces the (Pseudo) random patterns starting from the seed value and continue to generate all the possible 2^N-1combinations. The detection of faults by the random patterns is by probability. The ordering of the test vectors may or may not make an impact on the fault coverage in random pattern testing and it varies with different designs. A method of reversing the test vector order from end of the list and analyzing the trend of fault coverage is explained in [23]. The number of faults detected by a random pattern is usually high for the first few or many patterns and then reduces with further patterns. The detected faults are removed from the list after the application of each test pattern. In general, after a particular number of patterns, the fault coverage of random patterns saturates and even hundreds of patterns may detect a few faults only. These faults are the most difficult to detect. Further generated many random patterns may not detect even a single fault in the CUT. These undetected faults may also be referred as random pattern resistant faults. Fig.1 shows the general

trend in fault coverage with the size of the random patterns.

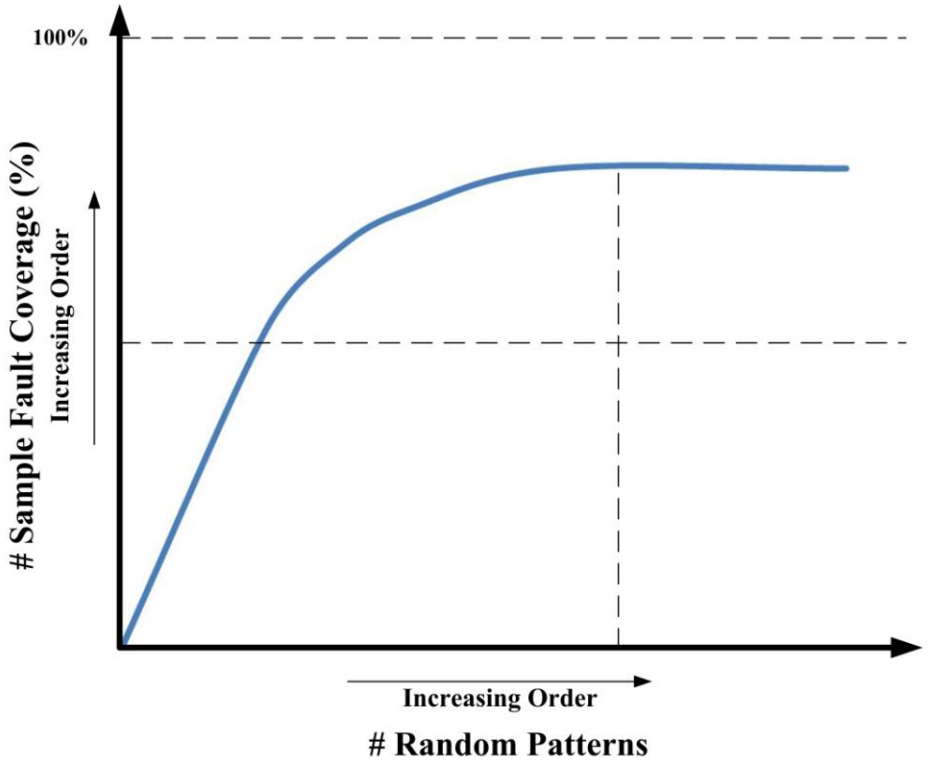

Fig. 1. **Typical Logic BIST fault coverage map**

It is not known precisely when to stop generating these random patterns and is usually done when no more improvement in fault coverage is seen. Common multi-cycle shift and single capture with hold is adapted here. Single cycle access test with peak power reduction is discussed in [24].

3.2. Methodology

Testing using the set of exhaustive patterns need a lot of time and resources, so techniques are needed to reduce this count. Selecting a proper seed can significantly help us in reducing these random test vectors to a lesser number. Usually the ATPG generates the deterministic test patterns for a design. These deterministic patterns are a subset of exhaustive patterns and our aim lies to use lesser number of exhaustive patterns with increased number of these deterministic patterns in it thereby achieving fault coverage as of ATPG. These ATPG patterns are then analyzed in the set of random patterns from which seed is selected and random patterns are limited.

Fig.2 shows a typical Logic BIST architecture. The Fig. 3 shows the pattern matching between the ATPG and random patterns for an example 8-bit LFSR. The first step is to find the location of these deterministic ATPG patterns in the set of huge exhaustive patterns and the distance from the default seed value, ie, the initial value is noted. Thus another subset is obtained which contain the distances. The highest

among it can be deleted as it can be thousands for a SoC design and the end of the highest distance is used as seed. This can be made clearer by an example shown, here the position numbers from the initial value is found to be 4,6,25,27,28,31 and 2. The highest distance or we can say the longest gap is between 6 and 25 which is almost 19. The seed value is selected 24 (patterns 00011) because the number of random patterns generated keeping 24^{th} position as seed is reduced to 14 and includes all the ATPG patterns in it, thus helping in achieving our aim of reduce pattern count and fault coverage as of ATPG. This reduction in number of random test patterns can help in saving significant test power consumption on each manufactured part. Another advantage of this technique, being BIST patterns and seed selection based on the ATPG deterministic patterns; it avoids the need for reseeding method [25]. In addition to that, it reduces the testing time significantly.

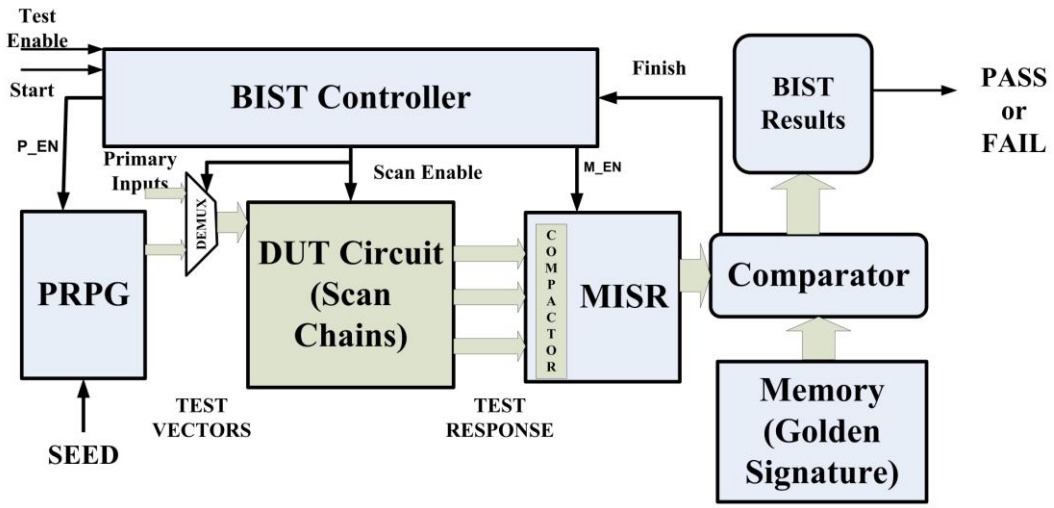

Fig. 2. **Logic BIST Architecture**

So the fault coverage is equal to ATPG with just 14 patterns in Logic BIST instead of the entire 31 patterns. This reduction in the number of random patterns grows tremendously with larger designs where several tens of thousands of random patterns are employed. Minimum test length seed selection based on polynomial analysis [26] does not compare ATPG pattern set. The proposed method employs two LFSR's, LFSR_1 for primary inputs and LFSR_2 for scan inputs (SI), thereby increasing controllability [27]. Fig. 7 shows the proposed algorithm. The steps involved for selecting the seed and subset of the random patterns are specified below through steps 1 through 6. Let 'L' be the count of exhaustive patterns and 'K' be the count of ATPG patterns.

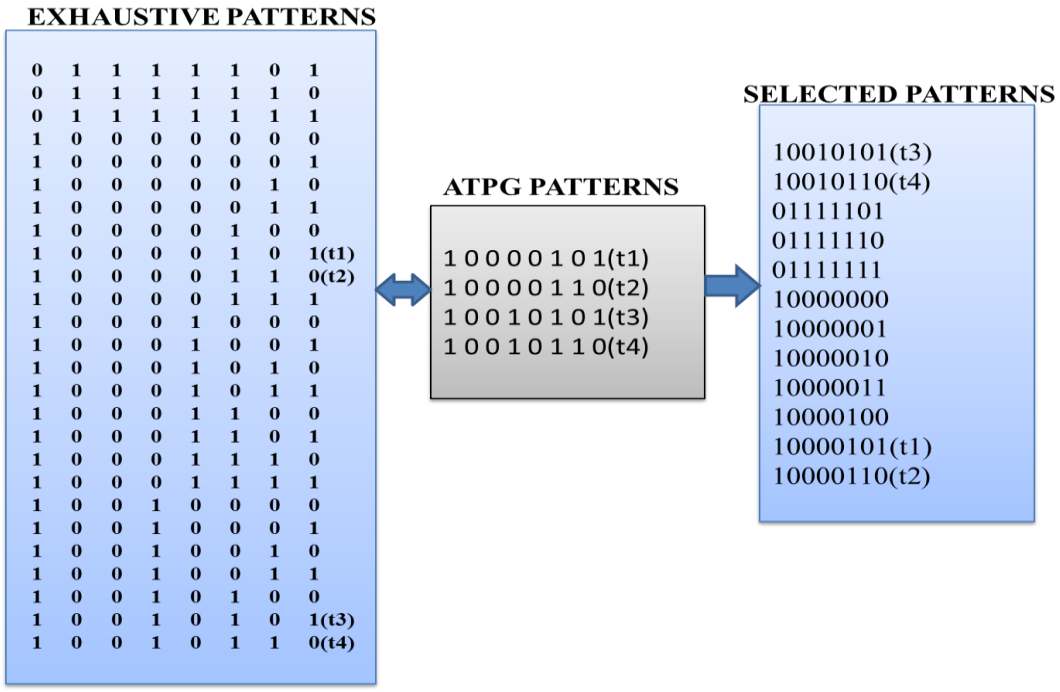

Fig. 3. **Seed selection based on ATPG patterns**

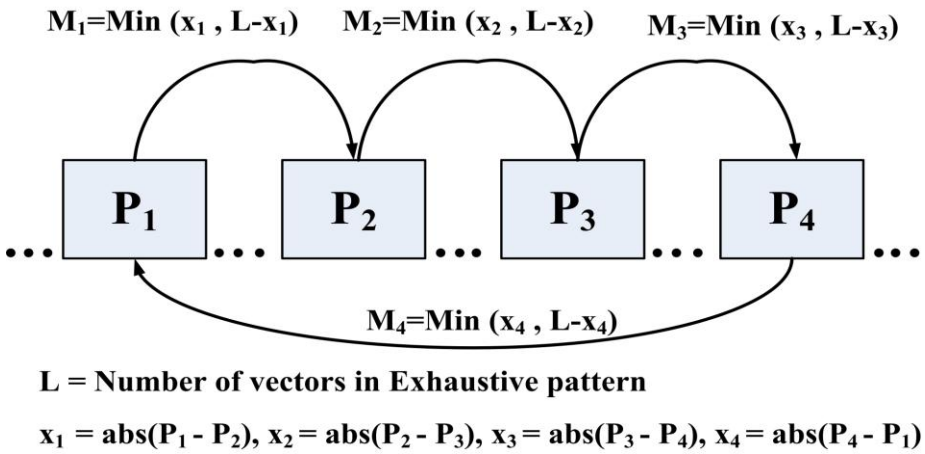

Fig. 4. **ATPG based Seed selection algorithm**

As shown in Fig.4. following steps implement the algorithm:

1. Generate the exhaustive test set and ATPG test set for the CUT.
2. Compare test vectors produced by ATPG and find its position in the exhaustive set.
3. Arrange the positions in ascending order. (P_1, P_2,…)
4. Take the absolute difference between the positions (x_1,x_2,x_3,…) and find largest distance (eg. M_2) elements, pick both of them.(eg. P_2 and P_3).

5. Find distance from picked elements to other position elements.
6. From this the pattern with minimum distance is the seed P_3 such that when P_3 is kept as seed the exhaustive set is reduced to minimum with all the deterministic patterns in the new set.

Two common important factors to measure the performance of the algorithm output is evaluated below.

Efficiency, η, is the measure of pattern reduction with respect to the exhaustive test set.

$$\eta = M/L \quad \ldots \quad (2)$$

In other way, the Gain, β, is the measure of pattern compression and is calculated as

$$\beta = L/M \quad 3)$$

M is the number of compressed patterns after seed selection technique and L is the total number of exhaustive patterns.

4. Results on ISCAS Benchmark

Seed selection and Classification of Random patterns technique was developed and simulated in Matlab and experiments were conducted on ISCAS'89 benchmark designs on a computer with 3.1 GHz CPU and 8GB memory. All the hardware designs were in Verilog HDL and simulations are performed in Modelsim RTL simulator. These designs were synthesized using 90nm standard cells library in Cadence Encounter Tool. The ATPG test simulation is performed in Cadence Encounter Test Architect and pattern classification techniques are implemented using Matlab. A deterministic test pattern set is generated from the Standard Test Interface Language (STIL) file output of the ATPG tool. All the test vectors are created for the stuck at fault model.

TABLE I. **SCAN STRUCTURE DETAILS, FAULT SUMMARY AND ATPG DETAILS ON ISCAS'89 DESIGNS**

	# PI	**# PO**	**Scan chains #**	**Scan FF / chain**	**LFSR #1 size (PI)**	**LFSR #2 size (SI)**	**Speed (GHz)**	**Static**		**# Nets**	**Test cycles**	**Scan cycles**	**Test seq**	**Test cvge**
								collapsed	uncollapsed					
s27	4	1	1	3	4	5	0.5	90	102	125	19	27	8	98
s298	3	6	2	7	3	8	0.86	468	538	651	53	182	25	100
s344	9	11	3	5	9	7	0.824	553	651	767	57	140	27	100
s349	9	11	3	5	9	8	0.91	557	659	767	55	135	26	100
s382	3	6	3	7	3	8	0.821	668	746	937	69	238	33	100
s386	7	7	1	6	7	9	0.966	414	568	582	105	312	51	100
s400	3	6	3	7	3	8	0.88	679	767	936	71	245	34	100
s420	18	1	1	16	18	10	0.274	754	790	1016	10	64	57	100
s444	3	6	3	7	3	8	0.88	683	759	943	69	238	33	100
s510	19	7	1	6	15	9	0.87	840	1230	1249	161	480	79	100
s526	3	6	3	7	3	8	0.63	704	844	975	69	238	33	100
s641	35	24	1	19	15	10	0.62	785	977	1048	89	836	43	100

s713	35	23	1	19	15	10	0.62	785	975	1050	97	912	47	100
s820	18	19	1	5	18	9	0.89	880	1262	1243	207	515	102	100
s832	18	19	1	5	18	10	0.951	915	1333	1310	221	550	109	100
s838	34	1	1	32	15	12	0.78	986	1058	1402	207	3296	102	100
s953	16	23	1	29	16	12	0.82	1543	1993	2162	201	2900	99	99.9
s1196	14	14	6	3	14	10	0.75	1531	2133	2001	367	549	182	100
s1238	14	14	6	3	14	10	0.744	1526	2136	1993	363	543	180	100
s1423	17	5	2	37	18	12	0.232	2473	2935	3439	157	2886	77	100
s1488	8	19	1	6	8	10	0.93	1734	2552	2526	275	822	136	100
s5378	35	49	7	25	15	9	0.674	5561	6511	7660	371	4810	184	99.9
s9234	36	39	10	15	15	13	0.47	4316	5124	6206	167	1245	82	100
s13207	62	152	30	20	15	15	0.298	14035	15598	20772	255	2540	126	99.9
s15850	77	150	28	15	15	14	0.21	14812	16875	21231	255	2413	126	99
s38417	28	106	28	56	28	15	0.30	42000	47679	62017	385	10752	191	99
s38584	38	304	38	34	15	15	0.29	43946	54398	61541	451	7650	224	98

5. Scan Insertion and Pattern Reduction Analysis:

Different exhaustive random test vector patterns are produced from the LFSR #1 and LFSR #2 combined for every ISCAS benchmark. Since the seeds of the LFSR and MISR can be programmed using external logic and pins of the design, this technique does not need any extra logic to add in to the original design. This technique being offline and one time operation, Matlab is used to implement the vector seed selection and classification purposes. The seeds used for every design before compression was "000…001" for all the LFSR's used in the designs. This seed is called as **β0.**

The tables show results of the experiments when run on ISCAS'89 benchmark sequential circuits. Table I shows the scan and design details on the ISCAS'89 designs. The first column shows the name of the design. The second and third columns show the number of input and output ports of the designs respectively. Third and fourth columns show the number of scan chains and the number of scan able registers in each scan chain respectively. Fifth and sixth columns show the LFSR sizes employed for the (PI) primary inputs and (SI) scan inputs respectively. Eighth column shows the speed. Ninth and tenth columns show the number of static stuck at faults in collapsed and un collapsed form. Eleventh column shows the number of nets in the CUT. Twelth and thirteenth columns show the number of test cycles and scan cycles employed. Fourteenth and fifteenth columns show the number of test vectors and test coverage respectively.

All the designs were taken through the Cadence Encounter Test Architect ATPG flow for checking the maximum fault coverage and test coverage. The number of untestable faults was zero and the total test coverage for all the designs was 100% except for the few designs whose circuit style has some redundant faults. Also for some designs with large number of inputs, the subset of the patterns generated in the simulations were used.

The results of the pattern size compression with respect to exhaustive pattern sizes are shown in Table II based on the proposed technique. In Table II, the first column shows the name of the design. The second column shows the number of scan registers. The third and fourth columns show the number of conventional BIST

patterns and ATPG patterns generated to achieve the test coverage shown in column 8. Column 5 shows the optimized number of the random test patterns generated by the proposed approach. Column 6 shows the performance benefit of this approach as a measure of gain. Columns 7 show the efficiency as a measure of the reduction in patterns size between the conventional BIST and proposed approach.

Note here that an average 81% reduction from the huge set of random test patterns is achieved. This will in turn, reduce the time for testing and also the testing power.

TABLE II. **SEED SLECTION BASED PATTERN COMPRESSION ANALYSIS**

CUT Ckt.	Scan FF #	Seed **β0** BIST Vectors	Proposed Seed # BIST Vectors	Gain	Efficiency (%)	Test Cov. (%)
s27	3	50	8	6.25	84	98
s298	14	80	7	11.4	91.25	100
s344	15	1100	435	2.53	60.45	100
s349	15	1100	478	2.3	56.54	100
s382	21	30	6	5	80	100
s386	74	300	123	2.43	59	100
s400	21	80	6	13.3	92.5	100
s420	16	600001	246624	2.43	58.89	100
s444	21	80	6	13.3	92.5	100
s510	6	32766	14007	2.34	57.25	100
s526	21	30	6	5	80	100
s641	19	32766	15385	2.13	53.04	100
s713	19	32766	14435	2.27	55.94	100
s820	5	600001	252401	2.38	57.93	100
s832	5	600001	261190	2.3	56.46	100
s838	32	32792	11984	2.74	63.45	100
s953	29	32792	11439	2.87	65.11	99.9
s1196	18	40000	16061	2.5	59.84	100
s1238	18	40000	15992	2.5	60.02	100
s1423	74	300000	128622	2.33	57.12	100
s1488	6	600	246	2.44	59	100
s5378	175	32766	15606	2.1	52.37	99.9
s9234	150	32766	11956	2.74	63.51	100
s13207	600	32766	9029	3.63	72.45	99.9
s15850	420	32766	15321	2.14	53.26	99
s38417	1568	32766	11903	2.75	63.67	99
s38584	1292	32766	12738	2.57	61.12	98

6. Modifications Proposed for Large Designs

Though the proposed technique works well with small VLSI designs following issues needs to be taken care while applying this technique to large SoC designs.

(i) *Pattern Variability:* While using a separate LFSR for all the scan chains the variable patterns that can be applied in to the scan chains are limited. For example, while using a 4 bit LFSR in a design with 4 (N) scan chains each of length with 10 (M) scan registers in it. When there are N number of scan chains present in a given design this technique suggests the designer to use a N+K bit LFSR. Also a decompressor logic circuit can be used to increase the pattern variability.

$$\mathbf{K = (\log_2 M/2)} \quad \text{.....} \quad (4)$$

where, M = Number of Flip-Flops in the scan chain.

(ii) *Pattern searchability issue:* Since the number of scan chains in the ISCAS benchmarks were small in number it was practically possible to search the subset of patterns from the exhaustive set of patterns. But, in practical designs the number of scan chains may be also in the range of few hundreds. The task of searching the ATPG patterns in that range is almost impossible task.

For large design units where hundreds of scan chains are created, it is important to divide the LFSR #1 in to multiple number of smaller LFSR's so that the size of the exhaustive pattern to be searched is always a finite and can be worked out with known pattern searching methods. Using this technique, while the complexity of the searchability reduces to a large extent the number of iterations may increase to some extent depending on the number of smaller LFSR's utilized in the testing structure. This method can be employed in the SoC designs with the proper techniques suggested above. The typical circuit structure is shown in the fig.5, whereas the individual component size may vary depending the CUT.

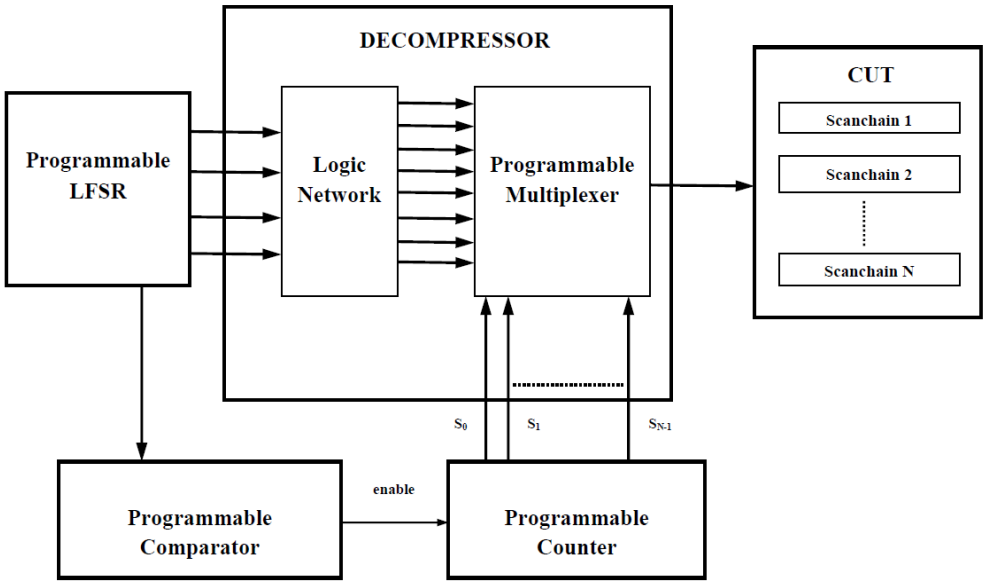

Fig. 5. **Multiple LFSR and Decompressor Logic for Pattern variability in large designs**

7. Conclusion

A technique to reduce the Logic BIST random patterns is suggested. ISCAS'89 designs with reconfigurable Logic BIST implemented and pattern reduction without changes in the ATPG scan structure of the design units is presented. The initial value of the compacted pattern set is selected as the seed for the PRPG. It is shown that the compacted pattern set is much smaller than the exhaustive pattern size even for small sized ISCAS designs. This technique needs modification in the test structure by adding a decompressor logic and dividing the single LFSR in to smaller sized multiple LFSR's to work with large designs. For the ISCAS designs, it is shown that the ATPG pattern set and the reduced random pattern set generated by the proposed technique have the same test coverage. Since this is an offline technique it does not add any overhead to the testing time. This method can be combined with multi-seeding, bit-flipping and bit fixing techniques for further compression.

8. References

[1] Michael L Bushnell, Vishwani D Agrawal, "Essentials of Electronic Testing For Digital, Memory And Mixed Signal VLSI Circuits," Kluwer Academic Publishers, 2002.

[2] L.-T. Wang, C.-W. Wu, and X. Wen, Eds., VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann, San Francisco, 2006.

[3] L.-T. Wang, C. E. Stroud, and N. A. Touba, Eds., "System-on-Chip Test Architectures: Nanometer Design for Testability", Morgan Kaufmann, San Francisco, 2007.

[4] P.H. Bardell and W.H. McAnney, "Self-Testing of Multiple Logic Modules," in *Proc. IEEE Int. Test Conf.*, pp. 200–204, 1982.

[5] J. Qian, X. Wang, Q. Yang, F. Zhuang, J. Jia, X. Li, Y., Zuo, J. Mekkoth, J. Liu, H.-J. Chao, S. Wu, H. Yang, L. Yu, F. Zhao, and L.-T. Wang, "Logic BIST Architecture for System-Level Test and Diagnosis," in Proc. IEEE Asian Test Symp., pp. 21–26, 2009.

[6] Katarzyna Radecka, Janusz Rajski, Jerzy Tyszer, "Arithmetic Built-In Self-Test for DSP Cores", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 16, No. 11, November 1997.

[7] Nur A. Touba and Edward J. McCluskey, "Transformed Pseudo-Random Patterns for BIST", Proceedings of the International Test Conference, pp.674-682, 1995.

[8] Das D. and Touba, N.A., "Reducing test data volume using external/LBIST hybrid test patterns ",Proceedings. of the International Test Conference, pp. 115 - 122, 2000.

[9] Chen, C.-I.H. ; Noh, T.H., "VHDL behavioral ATPG and fault simulation of digital systems", IEEE Transactions on Aerospace and Electronic Systems, Volume: 34 , Issue: 2 , pp.428 – 447, 1998.

[10] Jin-Fu Li, Hsin-Jung Huang, Jeng-Bin Chen, Chih-Ping Su, Cheng-Wen Wu,Chuang Cheng, Shao-I Chen, Chi-Yi Hwang, and Hsiao-Pin Lin, "A Hierarchical Test Scheme for System-on-Chip Designs", Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition, pp: 486 – 490.

[11] Yu Huang, Xijiang Lin, "Programmable Logic BIST for At-speed Test" proceedings in the Asian Test Symposium, ATS '07, IEEE Computer Society, pp: 295-300.

[12] Ramesh Bhakthavatchalu, Deepthy G R, Sreenivasa Mallia S3, HariKrishnan R, "32-bit Reconfigurable Logic-BIST Design Using Verilog for ASIC Chips" proceedings in 2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS).

[13] Liyang Lai, Wu-Tung Cheng, Thomas Rinderknecht, "Programmable Scan Based Logic Built in self test" 16th IEEE Asian Test Symposium, IEEE computer society, IEEE 2007, pp: 371 - 377

[14] Fei Zhuang, Junbo Jia, Xiangfeng Li, "System Level LBIST Implementation", published in Asian Test Symposium, 2008. ATS '08. 17th, Nov, 2008. Pp 263

[15] Andal Jayalakshmi, Tan Ewe Cheong, "A Methodology for LBIST Logic Diagnosis" proceedings in the 4th Asia Symposium on Quality Electronic Design, 2012 , pp 249 - 253

[16] Antonis Paschalis, Ioannis Voyiatzis, and Dimitris Gizopoulos "Accumulator Based 3-Weight Pattern Generation", IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 20, No. 2, February 2012

[17] Kedarnath J. Balakrishnan, and Nur A. Touba,"Improving Linear Test Data Compression", Ieee Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 14, No. 11, November 2006, pp. 1227-1237.

[18] Huan-Chih Tsai, Kwang-Ting (Tim) Cheng and Sudipta Bhawmik, "On Improving Test Quality of Scan-Based BIST", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 19, No. 8, August 2000, pp. 928-938.

[19] Peter Wohl, John A. Waicukauski, Sanjay Patel and Minesh B. Amin, "Efficient Compression and Application of Deterministic Patterns in a Logic BIST Architecture", DAC 2003, June 2-6, 2003, pp. 566-569.

[20] Ahmad A. Al-Yamani, Subhasish Mitra, and Edward J. McCluskey, "Optimized Reseeding by Seed Ordering and Encoding", Ieee Trnsactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 24, No. 2, February 2005, pp. 264-270.

[21] Michael Cogswell Don Pearl James Sage Alan Troidl, "Test Structure Verification of Logical BIST: Problems and Solutions", proceedings in the ITC International Test Conference, IEEE 2000, pp: 123 – 130.

[22] LAUNG-TERNG WANG AND EDWARD J. McCLUSKEY, " Condensed Linear Feedback Shift Register (LFSR) Testing – A Pseudoexhaustive Test Technique", IEEE Transactions On Computers, Vol. C-35, No. 4, April 1986, pp.367 – 370.

[23] Dorit **S.** Hochbaum, "An Optimal Test Compression Procedure for Combinational Circuits", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 15, No. 10, Pp. 1294 – 1299, October 1996.

[24] Tobias Strauch, "Single Cycle Access Structure for Logic Test", IEEE Transactions On Very Large Scale Integration Systems, Vol. 20, No. 5, May 2012, pp. 878 – 891.

[25] S. Hellebrand, B. Reeb, S. Tarnick, and H.-J.Wunderlich, "Pattern generation for a deterministic BIST scheme," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1995, pp. 88–94.

[26] Kenichi ICHINO, Ko-ichi WATANABE, Masayuki ARAI, Satoshi FUKUMOTO and Kazuhiko IWASAKI, "A Seed Selection Procedure for LFSR-Based Random Pattern Generators", Proceedings of the Design Automation Conference, 2003. pp. 869 – 874.

[27] Ramesh Bhakthavatchalu, Sreeja Krishnan, Vineeth. V., Nirmala Devi M., "Deterministic Seed Selection and Pattern Reduction in Logic BIST" proceedings in 2014, 18th IEEE International VLSI Design and Test Symposium (VDAT-2014).

Authors Biography

Ramesh Bhakthavatchalu received his B.E from Vellore Institute of Technology in 1994. He completed his M.E in Applied Electronics from College of Engineering, Guindy, Anna University in 1998. He served as Senior Design Application Engineer in Cirrus Logic Inc., USA and Syntest Technologies, USA for 8 years. He has taped out 4 designs during his tenure in VLSI Industry. Currently he is working as Assistant professor in ECE department of Amrita Vishwa Vidyapeetham, Amritapuri, India for the past 10 years. His areas of expertise are Design For Testability, FPGA based system design, VLSI signal processing. He has published more than 15 international conference and journal papers in VLSI design and testing area in last 3 years.

Dr. M. Nirmala Devi obtained her B.E. degree in Electronics and Communication Engineering (ECE) in 1990 and M. E. (Applied Electronics) Degree in 1996 from Government College of Technology, Coimbatore, Bharathiar University. She is currently working as the **Professor** in Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore. She has received her **Ph.D** degree in the area of **VLSI Design of Artificial Neural Networks** from Anna University, Chennai.

She is the recipient of "Appreciation Award" from Amrita Institute of Technology during the year 2003. Moreover, Marquis Who's Who in the World 2011 distinguishes her as one of the leading achievers from around the country. Furthermore, International Biographical Centre, Cambridge, England has chosen her for inclusion in the prestigious publication "2000 Outstanding Intellectuals of the 21st Century – 2011". Her areas of interest include **VLSI Design and Testing, Computational Intelligence, Hardware Security and Trust, Evolvable Hardware and RF System Design**. She has published around 55 papers in the International Journals and Conferences in her field of expertise. She has served as the reviewer for a few international conferences and the international journals.