Scan Data Volume Reduction Using Periodically Alterable MUXs Decompressor

Yinhe Han[†], Shivakumar Swaminathan [††], Yu Hu[†], Anshuman Chandra[†††], Xiaowei Li[†]

[†] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, 100080*
Graduate School of Chinese Academy of Sciences, Beijing, China, 100039
[††] *IBM Microelectronics, Research Triangle Park, NC.*
[†††]*Synopsys, Inc. 700 E. Middlefield Rd. , Mountain View, CA 94043, USA*

ABSTRACT

This paper presents a decompression architecture using a periodically alterable MUXs decompressor for scan data volume reduction. Compared to static XOR network, the periodically alterable MUXs decompressor has multiple configurations to decode the input information more efficiently. Three different DFT techniques are proposed to handle hard, firm and soft cores, respectively. With the proposed pattern decompression algorithms and scan decompression architecture, smaller test data volume and test application time can be achieved as compared to previous techniques.

1. INTRODUCTION

With the rapid development of IC process and wide reuse of pre-designed intellectual property (IP) cores, it is possible to put a large number of transistors on a single chip, which pose serious test challenges. These challenges include test application time, test data volume, test power, use of expensive testers, etc. [1]. Among these, test time and test data volume are dominant problems in high-volume testing of ICs. For a full-scan circuit, both test data volume and test application time are proportional to the number of test patterns (*P*) and the length of the longest scan chain (*L*). Thus, new techniques need to be developed to reduce *P*, *L* or both to reduce test data volume and test application time.

L-Reduction can be obtained through many new scan architectures proposed recently. Using one scan-in pin to drive multiple scan chains can shorten the scan chain and scan shift time. Lee [2] proposed a methodology to use one scan-in signal to feed multiple circuits. The Illinois scan architectures presented in [3], [4] show a broadcast architecture in which one scan-in signal feeds multiple scan chains. [5] proposes a reconfigurable shared scan-in architecture. To achieve *P*-Reduction, new test compaction algorithms have been proposed recently. A test compaction algorithm in [17] is proposed to identify the redundant patterns and then remove them. In [11], [20] and [23], unified methods based on test compaction and compression are presented.

Test data compression is a another way to reduce the product of *P* and *L*. Golomb [6], VIHC [7], dictionary-based [21, 26] and nine coded [22] codes have been presented to compress test data. [8] describes a method based on statistical encoding by converting some specified input values to don't care values first. Reda [9] presents a mutation decoder to compress test patterns into bit stream that indicates which bits need to be flipped in current test slice to obtain the subsequent one.

"Scan Chains Hiding" or "Scan Chains Concealment" is another *L*-Reduction scheme. LFSRs are applied in [10] to hide scan chains. A large LFSR driven by few external pins is presented. It is composed of a large number of smaller LFSRs, each of which feeds a scan chain. The mutation version of LFSR: ring generator is presented in [12]. Few inputs information is expanded by the ring generator. Three distinct ATPG schemes are used to achieve a higher degree of compression ratio. Some other LFSR-based techniques are presented in [24] and [25]. The XOR network is another class of "hiding" technique. They are presented in [11, 14, 20, 23]. These methods use a network composed of XOR gates to expand the input information. It is a very efficient method when combined with the custom test compaction algorithm.

The MUXs decomporessor proposed in this paper can be also regarded as a good *L*-Reduction technique. Previous work based on MUXs decompressor is presented in [13] and [15]. In this paper, a periodically alterable MUXs decompressor is proposed. The proposed scheme is different from the previous techniques in the following ways:

(1) Only a single input is required to change the configurations, which saves the number of scan inputs required. Further, multiple configurations can guarantee to decode the input information flexibly and provide the high fault coverage.

(2) The MUXs decompressor scheme is based on two-pass patterns compaction, and achieves better results in most cases.

The rest of this paper is organized in six sections: the proposed decompression architecture and how to design it are described in Section 2. In Section 3 we present DFT flows for three types of cores. The experimental results are presented in the Section 4 followed by conclusions in Section 5.

2. PROPOSED DECOMPRESSION ARCHITECTURE

The proposed deompression architecture based on MUXs decompressor is shown in Figure 1. There are *N* external scan-in pins that feed *M* internal scan chains through the MUXs decompressor, where *M*>>*N*. The proposed scan architecture is similar to the Illinois scan architecture [3] where multiple scan chains are fed through a single scan-in pin. However, the decompression architecture presented in this paper is more efficient than [3] as the connection relations between the external scan-in pins and the internal scan chains can be altered

1081-7735/05 $20.00 © 2005 IEEE

periodically, which results in generating small number of more effective patterns. The periodic reconfiguration of the mapping between the scan-in pins and the internal scan chains is done by changing the control signals of the MUXs decompressor. The control signals are generated by the MUX Control Register (*MCR*) and the *T*-counter that runs for *T* cycles. The mapping relation between scan-in pins and the internal scan chains is loaded into the Mapping Register (*MR*) through a separate external pin as shown in the Figure 1. The contents of *MR* are loaded into the *MCR* after every *T* cycles. For example, if *T* is equal to 3, the *MCR* will be updated from the *MR* and the mapping would change after every 3 cycles. The reconfiguration period is equal to the size of *T*-counter, which is also equal to the number of select lines of MUXs decompressor.

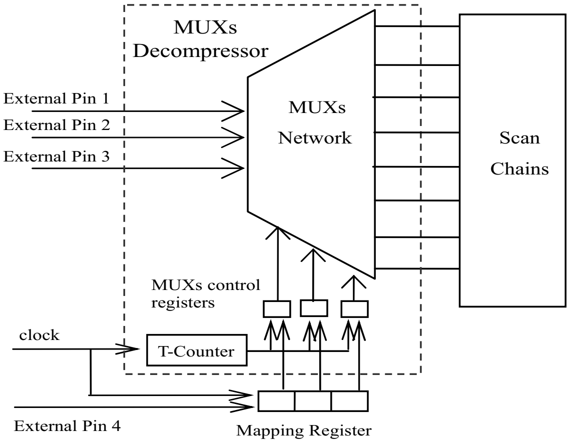

Figure 1. The Proposed Decompression Hardware

If the length of the longest scan chain is L and the number of patterns is P, then the entire test pattern will contain $L \times P$ scan slices. These slices can be partitioned into $B=\lceil L \times P / T \rceil$blocks.

In a block, each scan chain is a segment which contains T scan cells. A block can be represented by a scan chain incompatible graph CI-Graph: $G(V,E)$. In this graph, each node in V represents a scan chain segment. If the values of two nodes: V_i and V_j are incompatible, there is an edge between them. The following definition describes the "incompatible" property: For a scan chain segment S_i, $S_i[q]$ is the value of q^{th} scan cell in S_i. Given two scan chain segments S_i, S_j , they are incompatible, if $\exists\, q(1 \leq q \leq T)$, then ($S_i[q]$=0, $S_j[q]$=1) or ($S_i[q]$=1, $S_j[q]=0$).

In the CI-Graph, if we consider compatibility, the nodes can be partitioned into *independent sets*. In each *independent set*, any pair of nodes is compatible. Thus, the scan chains whose corresponding nodes are included in an independent set, can be assigned to one external input. Hence, the following building property can be obtained: *If a CI-Graph is partitioned into independent sets, these independent sets will correspond to a configuration. The scan chains within an independent set can be connected to one external input pin.*

In order to get the least number of external inputs, the least number of independent sets have to be determined. This problem of partitioning into independent sets equivalent to the graph coloring problem. The least number of independent sets is equivalent to obtaining the least number of distinct colors to color all nodes in a CI-Graph.

Corresponding to $\lceil L{\times}P/T \rceil$ blocks, there are $\lceil L{\times}P/T \rceil$ original CI-Graphs. However, this does not imply that the MUXs decompressor requires these many configurations. In fact, the CI-Graphs in a *N-Mergeable* set can be merged into one CI-Graph. As we computed the least number of external inputs, the least number of configurations can be reduced to a graph coloring problem. It is also NP-Complete. Many heuristic algorithms can be applied to determine the minimum number of configurations. However, in this paper, a simple heuristic approach based on CI-graph mergence was used to merge CI-Graphs.

3. DFT FLOWS FOR THREE KINDS OF CORES

3.1 DFT Flow for Hard Cores

Hard cores are cores which come with predesigned scan architecture and precomputed test patterns that cannot be modified by the system integrator. The test patterns and the information regarding the scan architecture are provided by the core vendor.

For such cores, the MUXs decompressor is designed as a wrapper surrounding the core. The DFT flow for hard cores is: The CI-Graphs are constructed first according to pattern blocks. Then the graph-coloring program is run to determine the maximum chromatic number among all CI-Graphs. After the number of external inputs have been determined, CI-Graphs are merged and minimize the area overhead. At last, the MUXs decompressor can be built based on the building property.

3.2 DFT Flow for Firm Cores

For firm cores, the scan architecture can't be modified, but the test patterns can be generated by using in-house ATPG tool. This provides additional freedom to customize the ATPG algorithm to compact patterns.

```
DFT Flow for Firm Core

(1) Use the uncompacted patterns to evaluate the least number
of required external pins: $N_{min}$;
(2) $N= N_{min}$;
(3) $ODV$= UnityFrame_Compaction&Compression($N$);
(4) $CDV=((M-N)/M)*ODV$;
(5) If ($N \leq N_{max}$), then $N=N+1$; Jump to (3);
(6) Select the pattern generated in (4) (5) (6) with the minimal
$CDV$, which should be the final pattern.

(7) Build the MUXs decompressor based on Property 1 and
Obtain the input data of MUXs decompressor using the decoding
equation.
```

Algorithm 2. DFT Flow for Firm Core

We first determine the minimum number of external input pins based on the uncompacted patterns as shown in Algorithm 1. There is a tradeoff between compression ratio and compaction efficiency. If the number of external pins is larger, the compression ratio will be lower, but it will be advantageous to compact. A search for the optimal balance between compression ratio and compaction efficiency is used by changing the external inputs number *N*. In order to get the minimum compressed test data volume and test application time, a more efficient algorithm is shown below:

In Algorithm 1, the test patterns are generated by the procedure of UnityFrame_Compaction&Compression(*N*). It is an algorithm that accounts for the compaction and compression at the same time. The algorithm is as follows:

```
UnityFrame_Compaction&Compression(N)

N: is the maximal chromatic number among all CI-Graphs;
UFL: undetected fault list;

(1)  Generate a test pattern T for a selected fault f.
(2)  While the UFL is not empty,
(2.a)        Generate a test pattern T' for a new selected fault f'.
(2.b)        Check two conditions:
         i.   T and T' are compatible
         ii.  The CI-Graphs of T and T' are N-Mergeable.
     If two conditions are both satisfied, then merge T and T' into a new T.
(3)  Fault Dropping: Run fault simulation and Delete all detected faults from UFL.
(4)  If undetected faults remain in UFL, go to step (2);
(5)  Run the Redundant Patterns Reduction Algorithm;
(6)  Return the final test data volume of patterns.
```

Algorithm 2. The Unity Frame of Test Compaction and Compression

Compared to the unity algorithm presented in [20], a Redundant Pattern Remove (RPR) algorithm is also performed. The algorithm in [20] is forward-compacting completely, so some faults detected by earlier patterns may also be accidentally detected by the test patterns later. Thus, some of the patterns generated earlier may become redundant.

3.3 DFT Flow for Soft Cores

The soft cores are cores that offer flexibility to reorganize the scan chains and allow custom ATPG algorithms to generate patterns. These cores provide maximum flexibility to the integrator.

[13] presents a scan chain design scheme based on the compatibility analysis of logic cones in a circuit. However, the compatibility analysis of a circuit is time-consuming as it goes through the entire scan circuitry and identifies the logic cones. In this paper, the scan chains will be organized based on the distribution of specified bits in test patterns. In the proposed scheme, the process on test patterns only needs to be conducted.

A careful study of the test patterns shows that some scan cells have higher probability to have a specified value than others. The "specified probability" of a scan cell is defined as:

$$P_{cell\ i} = \frac{the\ number\ of\ patterns\ with\ specified\ value\ in\ cell\ i}{the\ number\ of\ total\ patterns}$$

In the topology of a circuit, the cells with higher specified probability are often located in the shared areas of two or more different logic cones. Let's consider the figure below:

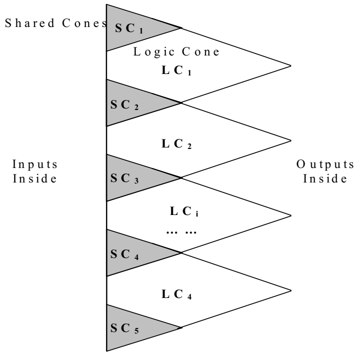

Figure2. Shared logic cones in a circuit

In the Figure 2, *LC* is the logic cone and *SC* is the shared cone between two logic cones. To sensitize a fault *f* in LC_1, the scan cells in SC_1, SC_2, and LC_1 may have to be specified. To sensitize a fault *f'* in LC_2, the scan cells in SC_2, SC_3, and LC_2 may have to be specified. Hence, in general, the scan cells in SC_2 should have higher specified probabilities than other scan cells in LC_1 and LC_2.

As analyzed in Section 2, in order to maximize the compression ratio, the chromatic number should be minimized. The chromatic number of CI-Graph relates to the number of edges and degrees of nodes. In general, the CI-Graph with few edges has a smaller chromatic number. Clustering the scan cells with higher specified probability into few scan chains can help to reduce the edges. The best case scenario is that all the specified cells are clustered in one scan chain; and then only one color is required to cover all nodes. Thus, to reduce the chromatic number, we propose the following scan chain reorganizing scheme: "*The scan cells with higher specified probability should be clustered in few scan chains*". The following example shows the improvement through proposed scan chains reorganization:

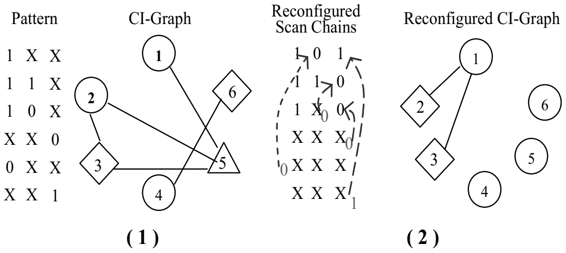

Figure 3. An example of reconfiguration of CI-Graph

In Figure 3(1), the specified cells are distributed in different scan chains. There are five edges in CI-Graph and the chromatic number is 3. The different shapes of nodes represent the different colors. If these specified cells are clustered only in the first, second and third scan chain as shown in Figure 3(2), there are only two edges and the chromatic number will be reduced to 2.

DFT Flow for Soft Cores

(1) Scan Chain Design:

(1.a) Generate the initial uncompacted test sets, and order the scan cells according to the specified probability.

(1.b) Based on the number of scan chains, determine the length of scan chain: *L*. The first *L* scan cells with highest specified probability encountered are assigned to the first scan chain. The following *L* scan cells with higher specified probability are assigned to the second chain. This procedure is repeated until all scan cells are assigned.

(2) Test Patterns Compaction:

(2.a) Run the Patterns Merging Algorithm as in Algorithm 3(Two conditions are considered);

(2.b) Run the Redundant Patterns Reduction Algorithm as in Algorithm 2.

(3) Build the MUXs decompressor based on the building property and Obtain the input data of MUXs decompressor using the decoding equation.

Algorithm 3. The DFT Frame for Soft Core

After the partition of scan chains, the same ATPG algorithm that was used for firm cores is utilized to generate the test patterns. Thus, the DFT Flow for Soft Cores consists of three phases: scan chain design, test patterns compaction and MUXs decompressor building and they are shown in the following hypothetical example.

In Figure 2, if the first phase of Algorithm 3 is performed, SC_1, SC_2, SC_3, SC_4, SC_5 may be clustered in a scan chain. The Figure 4 shows a possible assignment result:

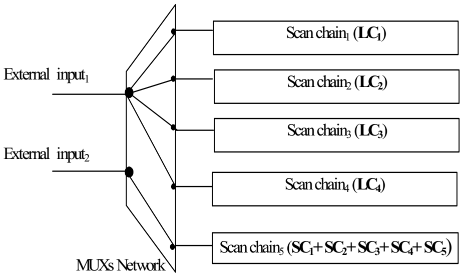

Figure 4. The Scan Cells Assignment of Algorithm 3

4. EXPERIMENTAL DATA

All the algorithms proposed in this paper were implemented using *C* language. The test patterns for single stuck-at fault were generated by using the ATPG tool *ATLANTA* [18] and the fault simulation was conducted on *HOPE* [19]. The chromatic numbers of CI-Graphs were obtained by the J. Culberson's[16] program. The proposed graph-coloring experimental flow is: The incompatible information of scan chains is recorded as DIMACS standard graph format file first. Then the *GREEDY* and *MAXIS* program is called to get the number of independent sets which are another representation of the chromatic number. Finally, Algorithms 1, 2, 3 are run to generate the MUXs decompressor and the test patterns. The MUXs decompressor is simulated by a software simulator. All the programs were run on the Linux platform.

In the first set of experiments, five large benchmark circuits are treated as cores. Each core is regarded as hard core, firm core, and soft core respectively and the proposed three DFT flows are applied to them. In the case of hard core, the *MINTEST*[17] patterns are used as it provides the smallest known test sets for these circuits. The experimental results are presented in Table 1 and Table 2. Table 1 presents the data that can be used to estimate the area overhead of MUXs decompressor. Table 2 presents the data on the compression ratio. In Table 1 and Table 2, *M*, *L*, *N*, *Pat.^{M}, No. Conf.* are the number of scan chains, the length of longest scan chain, the number of external inputs, the number of *MINTEST* patterns and the number of configurations in MUXs decompressor, respectively. *Pat.**, and *Pat.$^+$* are the number of patterns generated by UnityFrame _Compaction&Compression algorithm without RPR algorithm and with RPR algorithm, respectively. The superscripts * and $^+$ in *No. Conf**, *No. Conf$^+$* , *CDV** and *CDV$^+$* have same meaning as *Pat.**, *Pat.$^+$*. *ODV* and *CDV* are the data volumes of original test patterns and compressed input patterns. The number of scan chains is selected as 100 for all cases. For soft cores, the data in "Original Scan Chains" is obtained based on the random assignments of scan cells and "Optimized Scan Chains" implies the scan chains are organized by the first assignment based on Algorithm 3.

Table 1. The area overheads of MUXs decompressor to ISCAS 89 benchmark circuits

Cores	M*L	Hard Core			Firm Core					Soft Core					
										Original Scan Chains.			Optimized Scan Chains.		
		N	Pat^M.	No.Conf	N	Pat.*	Pat.$^+$	No.Conf*	No. Conf$^+$	N	Pat.$^+$	No. Conf$^+$	N	Pat.$^+$	No. Conf.$^+$
s13207	100*7	16	233	22	10	266	248	23	22	10	248	22	7	248	18
s15850	100*7	19	94	13	13	109	108	11	11	13	108	11	11	108	10
s35932	100*18	18	12	19	11	15	15	24	24	11	15	24	10	15	21
s38417	100*17	32	68	31	22	80	74	28	28	20	74	28	15	80	23
s38584	100*15	19	110	26	15	132	122	26	26	15	122	26	10	122	17

Table 2. The data volumes of MUXs decompressor to ISCAS 89 benchmark circuits

Cores	Hard Core				Firm Core					Soft Core					
										Original Scan Chains.			Optimized Scan Chains.		
	N	Pat.	ODV	CDV	N	Pat.*	Pat.+	CDV*	CDV+	N	Pat.+	CDV+	N	Pat.+	CDV
s13207	16	233	163,100	26,096	10	266	248	18, 620	17,360	10	248	17,360	7	248	12,152
s15850	19	94	57,434	12,502	13	109	108	9, 919	9,828	13	108	9,828	11	108	8，316
s35932	18	12	21,156	3,888	11	15	15	2, 970	2，970	11	15	2，970	10	15	2，160
s38417	32	68	113,152	36,992	22	80	74	29, 920	27，676	20	74	27，676	15	80	20, 400
s38584	19	110	161,040	31,350	15	132	122	29, 700	27，450	15	122	27，450	10	122	18，300

To further investigate the reduction of test data volume for the different schemes. We compare our method to the previous scan data compression methods in Table 3. The previous methods are mainly classified as: XOR-based and LFSR-based techniques. For most of the benchmark circuits, the proposed method provides better results. Compared to EDT [12], our method achieved 12. 2% higher compression ratio for four circuits on average. The 3-Stage [25] is an efficient compression scheme since its experimental results are far better than other schemes. However, our architecture provides better results compared to 3-Stage for three out of four large benchmark circuits.

In order to demonstrate the effectiveness of the proposed compression technique for industrial circuits, we present the experimental results on two production circuits from IBM: CKT1 and CKT2. These circuits are same as the CKT 1 and CKT 2 presented in [27]. CKT 1 is a logic core consisting of 51082 gates and its test set provides 99.80% fault coverage. There are 12256 scan cells and primary inputs. They are partitioned into 200 scan chains. CKT 2 is a logic core consisting of 94340 gates and its test set provides 99.76% fault coverage. There are 22216 scan cells and primary inputs. They are also partitioned into 200 scan chains. They are conducted as the hard cores. The experimental results are listed in Table 4. They show the volume of our proposed approach is 1/2.5 and 1/3.5 of the results in [27].

Table 3. The comparison data of different pattern compression methods

Cores	XOR-based			LFSR-based				[26]	[21]	Proposed
	[11]	[20]	[9]	[10]	[24]	[12]	[25]			
s13207	25,344	14,145	15,783	121,788, 824	11,285	10,585	11,320	99,252	11,402	12, 152
s15850	22,784	13,919	10,798	76,020	12,438	9,805	11,584	--	***7,240***	8，316
s35932	7,128	4,492	3,972	--	--	--	--	17,368	***2,347***	2，160
s38417	89,856	5,2793	42,264	308,508	34,767	31,458	30,560	820,715	51,739	20, 400
s38584	38,796	2,6644	22,636	202,370	29,397	18,568	27,248	303,072	30,752	18, 300

Table 4. The comparison to the input reduction technique [27]

Circuits	*Size of T_o*	*No. of Scan Chains*	*Len. Of Scan Chains*	Proposed approach		[27]	
				No. of external inputs	Size of *T_E (bits)*	No. of external inputs	Size of T_E *(bits)*
CKT1	46,180,608	128	96	11	3,979,008	16	6,374,400
		200	62	13	3,037,008	19	8,944,554
		400	31	20	2,336,160	26	6,560,840
		600	21	28	**2,215,584**	32	**4,906,272**
CKT2	58,561,376	128	174	8	3,669,312	-	-
		200	112	10	2,952,320	-	-
		400	56	13	1,919,008	16	7,705,600
		600	38	22	2,203,696	20	7,505,760
		800	28	22	1,623,776	23	5,314,288
		1000	23	25	**1,515,700**	27	**5,132,565**

5. CONCLUSIONS

A periodically alterable MUXs decompressor based scan data compression/decompression technique was presented to reduce the scan data volume and test application time. In order to handle different kinds of embedded cores, three different DFT flows were presented to design the scan chains and compact the test patterns. Experimental results show that if the proposed MUXs decompressor and the ATPG algorithms are used, the number of patterns and the test application time can be reduced simultaneously. Furthermore, the proposed method can be applied to the embedded cores, or it can be used as a stand-alone compression technique for large circuits.

REFERENCES:

[1] ITRS. *2001 Edition. http://public.itrs.net.*

[2] K. J. Lee, J. J. Chen, and C. H. Huang, "Using a single input to support multiple scan chains", *Proc. of ICCAD*, pp.74-78, 1998.

[3] I. Hamzaoglu and J.H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores", *Proc. of DFT,* pp. 260-267, 1999.

[4] M. Sharma, J. H. Patel, J. Rearick, "Test Data Compression and Test Time Reduction of Longest-Path-Per-Gate Tests based on Illinois Scan Architecture", *Proc. of VTS*, pp.25-20, 2003.

[5] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur, T. W. Williams, "A Reconfigurable Shared Scan-in Architecture", *Proc. of VTS, pp. 9-14, 2003.*

[6] A. Chandra, K. Chakrabarty, "System-on-a-Chip Test Data Compression and Decompression Architectures Based on Golomb Codes", *Trans. on CAD,* 20:113-120, Mar, 2001.

[7] P. T. Gonciari, B. Al-Hashimi, and N, Nicolici, "Integrated Test Data Decompression and Core Wrapper Design for Low-Cost System-on-a-Chip Testing", *Proc. of ITC,* pp. 64-73, 2002.

[8] S. Kajihara, Y. Doi, L. Li and K. Chakrabarty, "On Combining Pinpoint Test Set Relaxation and Run-Length Codes for Reducing Test Data Volume". *Proc. of ICCD*, pp.387-396, 2003.

[9] S. Reda, A. Orailoglu, "Reducing Test Application Time Through Test Data Mutation Encoding", *Proc. of DATE,* pp. 387-393, 2002.

[10] A. Jas, B. Pouya, N.A. Touba, "Virtual Scan Chains: A Means for Reducing Scan Length in Cores", *Proc. of VTS*, pp.73-78, 2000.

[11] I. Bayraktaroglu and A. Orailoglu, "Test volume and application time reduction through scan chains concealment", *Proc. of DAC,* pp. 151-155, 2001.

[12] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, "Embedded Deterministic Test", *Trans. On CAD*, 23(5), 2004.

[13] N. Sitchinava, S. Samaranayake, R. Kapur, E. Gizdarski, F. Neuveux, T. W. Williams, "Changing the Scan Enable during Shift", *Proc. of VTS,* pp. 73-78, 2004.

[14] L. Wang, X. Wen, H. Furukawa, F. Hsu, S. Lin, S. Tsai, K. S. Abdel-Hafez, and S. Wu, "VirtualScan: A New Compressed Scan Technology for Test Cost Reduction", *Proc. of ITC,* 2004.

[15] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur, T.W. Williams, "A Reconfigurable Shared Scan-In Architecture", *Proc. of VTS*, pp. 9-14, 2003.

[16] J. C. Culberson, "Iterated Greedy Graph Coloring and the Difficulty Landscape", Technical Report , TR-92-07, University of Alberta, Canada, 199, http://www.cs.ualberta.ca/~joe/Coloring/

[17] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits", *Proc. of ICCAD*, pp. 283-289, 1998

[18] H. K. Lee and D. S. Ha, "On the generation of test patterns for combinational circuits". Technical Report 12.93, Department of Electrical Eng., Virginia Polytechnic Institute and State University.

[19] H. K. Lee and D. S. Ha, "An efficient parallel fault simulator", *Proc. of DAC*, pp. 336-340, 1992.

[20] W. Rao, I. Bayraktaroglu and A. Orailoglu, "Test Application Time and Volume Compression Through Seed Overlapping", *Proc. of DA*C, pp. 732-737, 2003.

[21] X. Sun, L. Kinney and B. Vinnakota, "Combining Dictionary Coding And LFSR Reseeding For Test Data Compression", *Proc. of DAC*, pp. 944-947, 2004

[22] M. Tehranipour, M. Nourani, K. Chakrabarty, "Nine-Coded Compression Technique with Application to Reduced Pin-Count Testing and Flexible On-Chip Decompression", *Proc. of DATE*, pp. 1284-1289, 2004.

[23] I. Bayraktaroglu, A. Orailoglu, "Decompression Hardware Determination for Test Volume and Time Reduction through Unified Test Pattern Compaction and Compression", *Proc. of VTS*, pp. 113-118, 2003

[24] C.V. Krishna, A. Jas, and N. A. Touba, "Reducing test data volume using LFSR reseeding with seed compression", *Proc. of ITC*, pp. 321-330, 2002

[25] C.V Krishna, N. A. Touba, "3-Stage Variable Length Continuous-Flow Scan Vector", *Proc. of VTS*, pp.79-86, 2005.

[26] Lei L., K. Chakrabarty, "Test Data Compression Using Dictionaries with Fixed-Length Indices", *Proc. of VTS*, pp. 219-224, 2003.

[27] Lei L., K. Chakrabarty, S. Kajihara, S. Swaminathan, "Efficient Space/Time Compression to Reduce Test Data Volume and Testing Time for IP Cores", Proc. of VLSI Design, pp. 53~58, 2005.