A Reconfigurable Broadcast Scan Compression Scheme Using Relaxation Based Test Vector Decomposition

Aiman H. El-Maleh Mustafa Imran Ali Ahmad A. Al-Yamani
King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
{aimane, miali, alyamani}@kfupm.edu.sa

Abstract

An effective reconfigurable broadcast scan compression scheme that employs test set partitioning and relaxation-based test vector decomposition is proposed. Given a constraint on the number of tester channels, the technique classifies the test set into acceptable and bottleneck vectors. The bottleneck vectors are then decomposed into a set of vectors that meet the given constraint. The acceptable and decomposed test vectors are partitioned into the smallest number of partitions while satisfying the tester channels constraint to reduce the decompressor area. Thus, the technique by construction satisfies a given tester channels constraint at the expense of increased test vector count and number of partitions, offering a tradeoff between test compression, test application time and test decompression circuitry area. Experimental results demonstrate that the proposed technique achieves better compression ratios compared to other test compression techniques.

1. Introduction

This work presents a test vector compression scheme based on reconfigurable broadcast scan approach in which N scan chains are driven using M tester channels ($N \gg M$). Using compatibility analysis [1], test vectors are classified into '*acceptable*' and '*bottleneck*' vectors. Acceptable vectors are those that can be driven by M tester channels while bottleneck vectors are those that cannot be driven by M channels. Acceptable vectors are partitioned into the smallest number of partitions in such a way that all test vectors in a partition can be driven by M tester channels. Each partition corresponds to a test configuration and thus minimizing the number of partitions reduces the decoder complexity. Bottleneck vectors are decomposed into a small subset of test vectors each satisfying the tester channel constraint M, using an efficient relaxation-based test vector decomposition technique [3]. Then, the decomposed test vectors are partitioned minimizing the number of partitions.

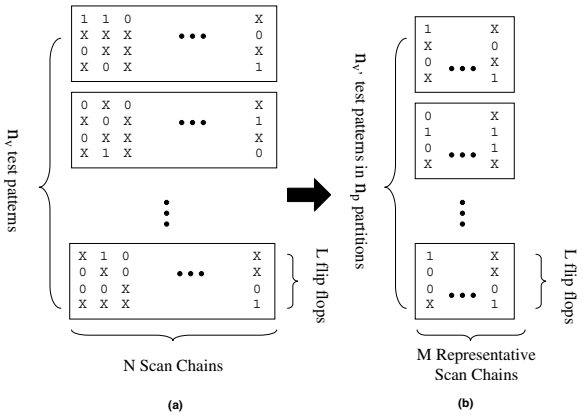

Figure 1. (a) Multiple scan chains test vectors configuration, (b) Algorithm's output.

This work uses a static reconfiguration approach [15], in which the configuration changes between test cubes or after a multiple of them. The advantages of the proposed technique are: (i) ATPG independence, relying only on fault simulations, which reduces the constraints on the ATPG to allow it to better cope with the increasing design complexity, possibly by using parallel ATPG techniques, (ii) being able to utilize compacted test sets to achieve high compression with relatively low test vector counts, and (iii) maximizing compression under area overhead constraints and minimizing area overhead under compression ratio constraints.

2. Proposed Compression Algorithm

The test set configuration used as input is shown in Figure 1(a). It consists of n_v test vectors, each configured into N scan chains of length L each. The compressed output is shown in Figure 1(b). It contains $n_{v'}$ test vectors, where $n_{v'} \geq n_v$, each encoded using M input scan chains (called representative chains) of length L, where in general $N \gg M$. These $n_{v'}$ vectors, called representative vectors, are distributed into n_p partitions.

The basis of partitioning is to encode a test set using a

given M representative chains. The number of representative scan chains and the groups of compatible chains can vary among test vectors in a test set depending upon the amount of specified bits present and their relative positions in each test vector. In a partition, member test vectors are all encoded using M scan chains and the same compatibility classes for all the members. However, it may be the case that many test vectors are not M colorable due to the relatively large number of specified bits present and their conflicting relative positions. These are called ' *bottleneck*' vectors while other test vectors are labeled '*acceptable*' vectors. The idea of test vector decomposition (TVD) is used to derive new acceptable vectors from the original bottleneck vectors by increasing the number of unspecified bits per vector. TVD is the process of decomposing a test vector into its atomic components. An atomic component is a child test vector that is generated by relaxing its parent test vector for a single fault f. That is, the child test vector contains the assignments necessary for the detection of f. Besides, the child test vector may detect other faults in addition to f.

The objective of the proposed algorithm is to compress a given test set using M tester channels while minimizing the increase in the number of test vectors (due to decomposition) and total partitions, and maintaining the original fault coverage (%FC). The increase in test vectors increases the test application time and beyond a certain point decreases the compression ratio, while the increase in number of partitions increases the area overhead of the decoder. To minimize the decomposition needed, the approach used is to minimize the number of undetected faults associated with each subsequently decomposed bottleneck vector as it directly affects the amount of decomposition required: the fewer the undetected faults, the lesser the decomposition required to derive new acceptable test vectors. Since a representative vector derived from broadcasted scan values is more specified than the original test vector, it detects more faults. To benefit from this fact, all acceptable vectors present in the input test set are partitioned and faults detected by the representative vectors obtained after partitioning are dropped. Then, during bottleneck vector decomposition, each derived subvector is partitioned and its representative vector is fault simulated to drop newly detected faults before any further decomposition. However, in this approach the fault coverage depends upon the representative vectors and if they are modified, the fault coverage changes. This may happen because a partition changes as new vectors are made members of existing partitions. If a partition attains a different configuration of compatibility classes to accommodate a new vector, all the representative vectors previously created for existing members of this partition need to be updated. The consequence is that some faults that are detected by the old set of representative vectors may become undetected. This can happen for faults that are essential in the original test set and detected by a bottleneck vector. When such faults are covered by some other representative vector, they are dropped and not considered during bottleneck vector's decomposition. However, the representative vector may be modified after the bottleneck vector has been decomposed, making the fault undetected, unless it is detected surreptitiously during the remaining pass by some representative vector. Surreptitious detection is more likely for non-essential faults. Two approaches can be used to deal with this problem: (i) either not to allow any previous essential fault detection to change while partitioning, or (ii) allow faults detection to be disturbed but address it by creating new vectors if needed. These two approaches can give different solutions in terms of total partitions created and the final test vectors count. The first approach tends to create more partitions as a vector will not be included in an existing partition if there is disturbance of any previous essential fault detection. On the other hand, the second approach may create more new vectors but has a potential to give fewer total partitions as a new vector is most likely to be included in an existing partition because of high percentage of don't cares present. This work uses the first approach to minimize new vectors needed and maximize compression. The proposed algorithm is given in Figure 2.

Compression Algorithm

1. *Fault simulate test set to mark essential and non-essential faults.*
2. *Analyze the compatibility of each test vector to get its* representative *scan chains (*representative count*). M is the maximum* representative *count that is acceptable, called the* threshold.
3. *Include vectors with* representative count $> M$ *in set* bottleneck*, otherwise in set* acceptable.
4. *Perform partitioning of all acceptable vectors.*
5. *Get representative test vectors for all partitioned test vectors, fault simulate them and drop all detected faults.*
6. *Incrementally decompose each bottleneck vector into subvector(s) for all its undetected faults. Partition each subvector and fault simulate its representative vector to drop faults before any further decomposition of the bottleneck vector.*
7. *Fault simulate the set of all representative vectors to check %FC. If %FC $<$ original, generate atomic components for all undetected faults and attempt merging them with existing partitions.*
8. *For remaining undetected faults, atomic components for these faults are merged into the smallest set of subvectors satisfying the threshold and are partitioned without disturbing existing fault detection.*

Figure 2. The main algorithm.

3. Decompression Hardware

The decompression hardware for compatibility based compression is simply a fan-out structure. The decompression hardware required to support partitioning of the test set can be realized by MUXs. Essentially, since each partition has a different set of compatibility classes, it requires it's own fan-out structure. The MUXs allow different fan-out structures to be switched in as required. Thus, the number of MUXs required is equal to the number of fan-out scan chains feeding the core, i.e., N. The number of data inputs on each of these MUXs is equal to the number of partitions i.e. n_p and are connected to one of the M ATE inputs. The cost of hardware is proportional to n_p times N. Since N is mostly a design parameter, the n_p required to achieve the desired compression determines the cost. However, it should be noted that the actual MUX sizes can be optimized by the synthesis tool, e.g., by utilizing the common tester channel inputs within a single MUX or across different MUXs.

4. Experiments

The algorithm was implemented in C/Linux. It uses HOPE simulator [8] and DSATUR graph coloring implementation by J. Culberson [2]. Full-scan versions of largest seven ISCAS-89 benchmark circuits have been used with MINTEST-generated [4] static compacted test sets. These test sets were relaxed using a bit-wise relaxation approach [3] to obtain don't cares.

The detailed results are given in Tables 1 and 2. In these tables, n_B, n'_v, n_p and $CR\%$ give the number of bottleneck vectors, the final test vector count, the number of partitions and the compression ratio, respectively. The compression ratio (CR%) is defined as $CR\% = \frac{n_v \times L \times N - n_{v'} \times L \times M}{n_v \times L \times N} = 1 - \frac{M}{N} \times \frac{n_{v'}}{n_v}$.

The highest compression that is achieved without any increase in test vectors is shown underlined while the overall highest compression achieved is shown in bold. With increasing compression, there is invariably an increase in the number of partitions required because of increased conflicts among the vectors when colored together, owing to a stricter constraint on the desired number of colors. Furthermore, as decomposition creates more vectors, these may require new partitions to satisfy the constraint. However, it should be noted that the test vectors count does not increase till a certain point even though the decomposition of bottleneck vectors takes place. This happens when a bottleneck vector(s) is decomposed only into a single vector because a portion of its faults are already covered by other representative vectors. It can also be observed that with decreasing M, the resulting compression increases until the increase in test set size overcomes any gains from reduced M. Compared to other test cases, s35932 and s38417 have a relatively small percentage of Xs and consequently a much larger number of bottleneck vectors. Hence, the compression ratios for these test cases are low, especially when smaller number of scan chains are used. Since the percentage of don't cares in current industrial designs are much higher, these two test cases are exceptions rather than indicative of the algorithm's performance.

Table 1. Results for s5378 and s9234.

s5378 ($N = 54$)					**s9234** ($N = 62$)				
M	n_B	n_p	n'_v	CR%	M	n_B	n_p	n'_v	CR%
11	2	22	97	79.44	10	1	34	105	83.81
10	3	29	97	81.31	9	2	40	105	85.43
9	6	36	97	83.18	8	3	45	105	<u>87.04</u>
8	18	37	97	<u>85.05</u>	7	8	54	107	88.45
7	23	50	99	86.65	6	13	67	108	90.01
6	46	57	108	**87.51**	5	37	81	129	**90.05**
5	67	52	148	85.74	4	70	92	166	89.76

In Table 3, a comparison with some multiple scan chains based techniques is given. Results for the proposed scheme are given with and without test vector increase over the MINTEST static compacted test set. The T_E values with increment in test vector counts also indicate the associated vector counts. For the smallest two circuits near 64 scan chains are used while for the larger five circuits both near 100 and 200 scan chains are used. The scan chains and test sets used in other schemes are as follows: Shi et al. [13] (FCSCAN) use a commercial ATPG tool and compression is reported with 200 scan chains, Rao et al. [11] (FLN) use ATLANTA ATPG tool [7] and compression is reported for scan chain lengths of 2, Li et al. [9] (3SC) use 64 and 200 scan chains respectively for the smallest two and larger five circuits with MINTEST test sets without compaction, Hayashi et al. [6] (SDI) use X-maximal program for test set generation with scan chains lengths varying between 16 and 64 in powers of 2, Tang et al. [14] (ONC) do not specify the specifics of test sets and all results are reported with 256 scan chains except for s15850, where 128 are used. Han et al. [5] (PAM) use ATLANTA ATPG tool with 100 scan chains, Samaranayake et al. [12] (RSS) use Synopsys TetraMax with near 100 scan chains while EDT [10] does not give the specifics of test sets and scan chains. The best results among the strategies compared with are shown in bold. The proposed technique achieves higher compression in all test cases except for s35932 at comparable or much lower vector counts, even though a conservative number of scan chains are used as compared to some of these schemes [11, 14].

5. Conclusions and Future Work

An effective test vector compression technique has been proposed in this work that uses test set partitioning and bottleneck test vector decomposition through relaxation. The technique targets a user specified number of ATE channels to achieve test data compression and it can

Table 2. Results with N approaching 100 for the largest five test cases.

s13207 ($N = 100$)					**s15850** ($N = 88$)					**s35932** ($N = 98$)					**s38417** ($N = 98$)					**s38584** ($N = 98$)				
M	n_B	n_p	n'_v	CR%	M	n_B	n_p	n'_v	CR%	M	n_B	n_p	n'_v	CR%	M	n_B	n_p	n'_v	CR%	M	n_B	n_p	n'_v	CR%
19	1	8	233	81.0	13	1	22	94	85.1	68	1	8	12	30.6	32	7	40	68	67.3	39	5	18	110	60.0
11	1	14	233	89.0	12	2	24	94	86.2	67	2	8	13	25.9	30	9	44	68	69.3	24	7	26	110	75.4
10	2	16	233	90.0	11	3	26	94	87.4	38	7	14	18	41.8	28	11	48	68	71.4	19	10	34	112	80.2
9	2	18	233	91.0	10	11	29	99	87.9	33	8	15	19	46.6	26	18	52	70	72.7	17	12	37	112	82.3
8	4	20	233	92.0	9	12	32	100	89.0	22	9	17	24	55.1	24	27	57	73	**73.7**	15	13	43	112	84.3
7	6	22	233	93.0	8	15	36	103	89.9	10	12	25	34	71.1	22	37	62	80	73.6	12	21	52	115	87.1
6	9	28	235	93.9	7	20	42	106	90.9	8	12	25	37	74.8	20	46	64	88	73.6	10	25	59	115	89.3
5	15	35	239	94.9	6	26	50	112	91.8	6	12	29	43	78.0	18	56	66	102	72.4	8	20	80	117	91.3
4	26	48	248	95.7	5	37	62	125	92.4	4	12	28	56	80.9	16	61	69	117	71.9	4	107	132	203	92.4
3	47	67	264	**96.6**	4	54	76	144	**93.0**	3	12	26	65	**83.4**	15	63	69	124	72.1	3	109	164	257	**92.8**

Table 3. Comparison with other multiple scan chain schemes.

	Proposed						[13] FCSCAN		[11] FLN		[9] 3SC		[6] SDI		[14] ONC		[5] PAM		[12] RSS		[10] EDT
	No incr.		**With Incr.**																		
	100	**200**	**100**		**200**																
Circuits	T_E	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	**#TV**	T_E	T_E
s5378	3104*	N/A	*108*	2592*	*N/A*	N/A	*N/A*	N/A	*N/A*	N/A	*395*	11180	*99*	5748	*N/A*	N/A	N/A	N/A	*N/A*	N/A	**5676**
s9234	3360*	N/A	*129*	2580*	*N/A*	N/A	*N/A*	N/A	*N/A*	N/A	*471*	18410	*110*	**8872**	*N/A*	N/A	N/A	N/A	*N/A*	N/A	9534
s13207	11417	4660	*248*	6944	*239*	3824	*251*	10920	*317*	13948	*477*	14087	*233*	13114	*415*	**4980**	248	17360	*296*	6512	10585
s15850	7238	3384	*144*	4032	*132*	2112	*148*	**7072**	*309*	13596	*422*	15907	*97*	11372	*386*	7720	108	9828	*448*	8960	9805
s35932	14688	8532	*37*	5328	*35*	1575	*35*	8045	*38*	**836**	*147*	3308	*12*	7252	*45*	1260	15	2970	*N/A*	N/A	N/A
s38417	31212	14076	*73*	29784	*68*	14076	*183*	29550	*678*	63732	*487*	69274	*86*	30404	*692*	**19376**	74	27676	*781*	21868	31458
s38584	39600	12320	*117*	14040	*121*	5808	*288*	21020	*477*	25758	*510*	54878	*111*	28140	*537*	**12888**	122	27450	*636*	16536	18568

* near 64 scan chains used

explore tradeoffs among compression ratio and area overhead. The technique relies on an efficient test relaxation algorithm and can work with compacted test sets to achieve high compression with much lower vector counts, thus minimizing test application time. The results clearly show that the proposed technique achieves significantly greater compression compared to other recent work. Further improvements may be achieved by directing test relaxation to avoid conflicting bit positions. This is being explored as a potential enhancement to this work.

Acknowledgment: This work is supported by King Fahd University of Petroleum & Minerals under project FT-2006/18.

References

[1] C. Chen and S. K. Gupta. Efficient BIST TPG Design and Test Set Compaction via Input Reduction. *IEEE T*, 17(8):692 –705, August 1998.

[2] J. Culberson. Graph Coloring Page.

[3] A. El-Maleh and A. Al-Suwaiyan. An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits. In *VTS '02*, page 53, 2002.

[4] I. Hamzaoglu and J. H. Patel. Test Set Compaction Algorithms for Combinational Circuits. In *Proc. Int. Conf. Comput.-Aided Des.*, pages 283–289, 1998.

[5] Y. Han, X. Li, S. Swaminathan, Y. Hu, and A. Chandra. Scan Data Volume Reduction Using Periodically Alterable MUXs Decompressor. In *ATS '05*, pages 372– 377, 2005.

[6] T. Hayashi, H. Yoshioka, T. Shinogi, H. Kita, and H. Takase. Test Data Compression Technique Using Selective Don't-Care Identification. In *ASP-DAC '04*, pages 230–233, 2004.

[7] H. K. Lee and D. S. Ha. On the Generation of Test Patterns for Combinational Circuits. Technical Report 12-93, Dept. of Electrical Eng., Virginia Polytechnic Institute and State University, 1993.

[8] H. K. Lee and D. S. Ha. HOPE: An Efficient Parallel Fault Simulator for Synchronous Sequential Circuits. *IEEE TCAD*, 15(9):1048–1058, September 1996.

[9] L. Li, K. Chakrabarty, S. Kajihara, and S. Swaminathan. Three-Stage Compression Approach to Reduce Test Data Volume and Testing Time for IP Cores in SOCs. *IEE Proc. Comput. Digit. Tech.*, 152(6):704–712, November 2005.

[10] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee. Embedded Deterministic Test. *IEEE TCAD.*, 23(5):776–792, May 2004.

[11] W. Rao, A. Orailoglu, and G. Su. Frugal Linear Network-based Test Decompression for Drastic Test Cost Reductions. In *ICCAD '04*, pages 721–725, 2004.

[12] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur, and T. Williams. A Reconfigurable Shared Scan-in Architecture. In *VTS'03*, April 2003.

[13] Y. Shi, N. Togawa, S. Kimura, M. Yanagisawa, and T. Ohtsuki. FCSCAN: An Efficient Multiscan-Based Test Compression Technique for Test Cost Reduction. In *ASP-DAC '06*, pages 653–658, 2006.

[14] H. Tang, S. M. Reddy, and I. Pomeranz. On Reducing Test Data Volume and Test Application Time for Multiple Scan Chain Designs. In *ITC '03*, pages 1079–1087, 2003.

[15] N. A. Touba. Survey of Test Vector Compression Techniques. *IEEE D&T of Computers*, pages 294–303, 2006.