

ALIASING-FREE COMPACTION IN TESTING CORES- BASED SYSTEM-ON-CHIP (SOC) USING COMPATIBILITY OF RESPONSE DATA OUTPUTS

Sunil R. Das

School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada, and
Department of Computer and Information Science, Troy State University Montgomery, Montgomery, AL 36103, U.S.A.

Mansour H. Assaf

Emil M. Petriu

School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada

Mehmet Sahinoglu

Department of Computer and Information Science, Troy State University Montgomery, Montgomery, AL 36103, U.S.A.

The realization of space-efficient support hardware for built-in self-testing (BIST) is of great importance in the design and manufacture of VLSI circuits. Novel approaches to designing aliasing-free space compaction hardware were recently proposed in the context of testing cores-based system-on-chip (SOC) for single stuck-line faults, extending the well-known concepts of conventional switching theory, specifically those of cover table and frequency ordering commonly utilized in the simplification of switching functions, and of compatibility relation as used in the minimization of incomplete sequential machines, based on optimal generalized sequence mergeability, as developed and utilized by the authors in earlier works. The advantages of these aliasing-free compaction methods over earlier techniques are quite obvious, since zero-aliasing is achieved without any modifications of the module under test (MUT), while keeping the area overhead and signal propagation delay relatively low as contrasted with the conventional parity tree linear compactors. Besides, the approaches could be applied with both deterministic compacted and pseudorandom test patterns. The subject paper, without furnishing details of the different algorithms developed in the implementation of these approaches to designing zero-aliasing space compactors, provides the mathematical basis of selection criteria for merger of an optimal number of outputs of the MUT to achieve maximum compaction ratio in the design, along with some results from simulation experiments conducted on ISCAS 85 combinational and ISCAS 89 full-scan sequential benchmark circuits, with simulation programs ATALANTA, FSIM, and HOPE.

Keywords: *Aliasing-free (zero-aliasing) space compaction, built-in self-testing (BIST) in VLSI, compatibility of response data outputs, cores-based system-on-chip (SOC), module under test (MUT).*

1. Introduction

With ever-increasing complexity in systems design with increased levels of integration densities, better and more effective methods of testing to ensure reliable operations of chips, mainstay of today's many sophisticated digital circuits, are evidently a necessity. The very concept of testing has a broad applicability, and thus finding highly efficient testing techniques (for both hardware and software) to ensure correct system performance has, of late, gained immense importance (Bardell *et al.*, 1987, Rajsuman, 2000, Mourad and Zorian, 2002, Saluja and Karpovsky, 1983, McCluskey, 1985, Li and Robinson, 1987, Reddy *et al.*, 1988, Karpovsky and Nagvagara, 1990, Jone and Das, 1991, Pradhan and Gupta, 1991, Savir, 1996, Das *et al.*, 2001, Das *et al.*, 2003, Sahinoglu *et al.*, 2002, Sahinoglu, 2003, Chakrabarty, 1995, and Assaf, 2003). The conventional testing techniques for permanent hardware faults of digital circuits require application of test patterns generated by a test pattern generator (TPG) to the module under test (MUT) and comparing the responses with known correct responses. However, for large circuits, because of higher storage requirements for the fault-free responses, the test procedures become rather expensive, and hence alternative approaches are sought to minimize the amount of required storage. Built-in self-testing (BIST) is a design approach that provides the capability of solving many of the problems otherwise encountered in testing digital systems. It combines concepts of both built-in test (BIT) and self-test (ST) in one termed built-in self-test (BIST). In BIST, test generation, test application, and response verification are all accomplished through built-in hardware, which allows different parts of a chip to be tested in parallel, reducing the required testing time besides eliminating the need for external test equipment. A typical BIST environment uses a TPG that sends its outputs to an MUT and output streams from the MUT are fed into a test data analyzer. A fault is detected if the module response is different from that of the fault-free module. The test data analyzer is comprised of a response compaction unit (RCU), storage for the fault-free responses of the MUT, and a comparator.

The extra logic representing the compression circuit must be as simple as possible, to be easily embedded within the MUT, and should not introduce signal delays to affect either the test execution time or the normal functionality of the MUT. Besides, the length of the signature must be as short as possible in order to minimize the amount of memory required to store the fault-free responses. In addition, signatures derived from faulty output responses and their corresponding fault-free signatures should not be the same, which unfortunately is not always the case. A fundamental problem with compression techniques is error masking or aliasing (Chakrabarty, 1995, Das *et al.*, 2001, Assaf, 2003), which occurs when the signatures from faulty output responses map into fault-free signatures. Aliasing causes loss of information, which in turn affects the test quality of BIST, and reduces the fault coverage. Several methods have been proposed in the literature for computing the aliasing probability of which the exact computation is known to be NP-hard (Garey and Johnson, 1979).

This paper focuses on certain aspects of some recently suggested techniques to designing zero-aliasing space compactors of digital circuits (Pouya and Touba, 1998, Das *et al.*, 2001) and in the process, proposes novel approaches to implementing such support hardware with applications targeted towards embedded cores-based system-on-chip (SOC) (Rajsuman, 2000), extending the well-known concepts of conventional switching theory, particularly those of cover table and frequency ordering commonly utilized in the simplification of switching functions, and of compatibility relation as employed in the minimization of incomplete sequential machines, using the graph theory algorithm of finding the modified cut-sets (Das, 1973) in the design, based on optimal generalized sequence mergeability, as developed and applied by the authors in earlier works (Das *et al.*, 2001, Assaf, 2003), for detectable single stuck-line faults of the MUT. The advantages of these aliasing-free space compaction methods over earlier techniques are evidently clear – zero-aliasing is achieved without any modifications of the MUT, while the area overhead and signal propagation delay are relatively low

compared to conventional parity tree linear compactors. Furthermore, the approaches work equally well with both deterministic compacted and pseudorandom test sets. Besides, the developed approaches involve less computation in the successive stages compared to the existing techniques that generally use an exhaustive approach at every step to decide on the logic gate to be used for merger of only a pair of output lines of the MUT to achieve zero-aliasing, if at all possible. The paper, on the other hand, takes advantage of mathematically sound selection criteria of merger of an optimal number of output lines of the MUT to decide on the logic gate for zero-aliasing and achieves maximum compaction ratio in the design, as is evident from extensive simulation experiments conducted on ISCAS 85 combinational and ISCAS 89 full-scan sequential benchmark circuits, though only partial results on both ISCAS 85 and ISCAS 89 benchmark circuits are provided in the paper for the sake of brevity.

2. Mathematical Basis

Property 1 Let A and B represent two of the outputs of an MUT. Let these MUT outputs be merged by a gate from the logic family AND/NAND, OR/NOR, and XOR/XNOR, and let the gate output be z_1 . Then we can envisage the undernoted scenarios:

Case 1 Fault-free (FF) outputs = Faulty (F) outputs (outputs subject to the condition of MUT having faults), viz. $FF = F \Rightarrow$ Outputs A and B of the MUT do not detect any faults, and faults are hence not detectable at z_1 .

Case 2 Only those faults that occur at A and B (subject to the condition of MUT having faults) are detectable at $z_1 \Rightarrow FF \neq F$.

Case 3 Faults occur at A and B but not all or some are detectable at $z_1 \Rightarrow FF \neq F$. That is, these missed faults at z_1 are detected additionally at other outputs of the MUT besides A and B.

Definition 1 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Cases 1-2 above (but not Case 3). If the MUT outputs A, B are merged by an AND(NAND) gate, we define output lines A, B to be **strongly AND(NAND) compatible**, written as

(AB) **s-AND(NAND) compatible**.

Definition 2 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Case 3 (but not Cases 1-2). If the MUT outputs A, B are merged by an AND(NAND) gate, we define output lines A, B to be **weakly AND(NAND) compatible**, written as

(AB) **w-AND(NAND) compatible**.

Definition 3 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to none of the conditions as specified by Cases 1-3. If the MUT outputs A, B are merged by an AND(NAND) gate, we define output lines A, B to be AND(NAND) **incompatible**, written as

(AB) AND(NAND) **incompatible**.

Definition 4 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Cases 1-2 above (but not Case 3). If the MUT outputs A, B are merged by an OR(NOR) gate, we define output lines A, B to be **strongly OR(NOR) compatible**, written as

(AB) **s-OR(NOR) compatible**.

Definition 5 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Case 3 (but not Cases 1-2). If the MUT outputs A, B are merged by an OR(NOR) gate, we define output lines A, B to be **weakly OR(NOR) compatible**, written as

(AB) **w-OR(NOR) compatible**.

Definition 6 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to none of the conditions as specified by Cases 1-3. If the MUT outputs A, B are merged by an OR(NOR) gate, we define output lines A, B to be **OR(NOR) incompatible**, written as

(AB) **OR(NOR) incompatible**.

Definition 7 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Cases 1-2 above (but not Case 3). If the MUT outputs A, B are merged by an XOR(XNOR) gate, we define output lines A, B to be **strongly XOR(XNOR) compatible**, written as

(AB) **s-XOR(XNOR) compatible**.

Definition 8 Let A, B, C, ... be the different outputs of an n-input m-output MUT. Let the faults detected at the MUT outputs A, B, C, ... be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Case 3 (but not Cases 1-2). If the MUT outputs A, B are merged by an XOR(XNOR) gate, we define output lines A, B to be **weakly XOR(XNOR) compatible**, written as

(AB) **w-XOR(XNOR) compatible**.

Definition 9 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to none of the conditions as specified by Cases 1-3. If the MUT outputs A, B are merged by an XOR(XNOR) gate, we define output lines A, B to be XOR(XNOR) **incompatible**, written as

(AB) XOR(XNOR) **incompatible**.

Definition 10 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to either one of the three conditions as specified by Cases 1-3, but unknown to us. If the MUT outputs A, B are merged under these conditions by an AND(NAND), OR(NOR) or XOR(XNOR) gate, then we define output lines A, B to be **simply** AND(NAND), OR(NOR) or XOR(XNOR) **compatible**, written as

(AB) AND(NAND), OR(NOR) or XOR(XNOR) **compatible**.

Theorem 1 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Cases 1-2 above, so that the outputs A, B are **s-AND(NAND) compatible**. Similarly, let the outputs B, C be **s-AND(NAND) compatible**, and outputs A, C be **s-AND(NAND) compatible**. Then (ABC) is **s-AND(NAND) compatible** and all faults are detected at z_1 .

Theorem 2 Let A_1, A_2, \dots, A_m be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A_1, A_2, \dots, A_m be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A_1, A_2, \dots, A_m conforms to conditions of Cases 1-2 above, so that the outputs A_1, A_2, \dots, A_m are **s-AND(NAND) compatible**. Then all faults are detected at z_1 .

Theorem 3 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Cases 1-2 above, so that the outputs A, B are **s-OR(NOR) compatible**. Similarly, let the outputs B, C be **s-OR(NOR) compatible**, and outputs A, C be **s-OR(NOR) compatible**. Then (ABC) is **s-OR(NOR) compatible** and all faults are detected at z_1 .

Theorem 4 Let A_1, A_2, \dots, A_m be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A_1, A_2, \dots, A_m be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault

situation at the outputs A_1, A_2, \dots, A_m conforms to conditions of Cases 1-2 above, so that the outputs A_1, A_2, \dots, A_m are **s-OR(NOR) compatible**. Then all faults are detected at z_1 .

Theorem 5 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Cases 1-2 above, so that the outputs A, B are **s-XOR(XNOR) compatible**. Similarly, let the outputs B, C be **s-XOR(XNOR) compatible**, and outputs A, C be **s-XOR(XNOR) compatible**. Then (ABC) is **s-XOR(XNOR) compatible** and all faults are detected at z_1 .

Theorem 6 Let A_1, A_2, \dots, A_m be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A_1, A_2, \dots, A_m be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A_1, A_2, \dots, A_m conforms to conditions of Cases 1-2 above, so that the outputs A_1, A_2, \dots, A_m are **s-XOR(XNOR) compatible**. Then all faults are detected at z_1 .

Theorem 7 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Case 3 above, so that the outputs A, B are **w-AND(NAND) compatible**. Similarly, let the outputs B, C be **w-AND(NAND) compatible**, and outputs A, C be **w-AND(NAND) compatible**. Then (ABC) is **w-AND(NAND) compatible** and all faults may or may not be detected at z_1 .

Corollary 7.1 Let A_1, A_2, \dots, A_m be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A_1, A_2, \dots, A_m be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A_1, A_2, \dots, A_m conforms to conditions of Case 3 above, so that the outputs A_1, A_2, \dots, A_m are **w-AND(NAND) compatible**. Then all faults may or may not be detected at z_1 .

Theorem 8 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Case 3 above, so that the outputs A, B are **w-OR(NOR) compatible**. Similarly, let the outputs B, C be **w-OR(NOR) compatible**, and outputs A, C be **w-OR(NOR) compatible**. Then (ABC) is **w-OR(NOR) compatible** and all faults may or may not be detected at z_1 .

Corollary 8.1 Let A_1, A_2, \dots, A_m be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A_1, A_2, \dots, A_m be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests $\tau, \tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A_1, A_2, \dots, A_m conforms to conditions of Case 3 above, so that the outputs A_1, A_2, \dots, A_m are **w-OR(NOR) compatible**. Then all faults may or may not be detected at z_1 .

Theorem 9 Let A, B, C, \dots be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A, B, C, \dots be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A, B conforms to conditions of Case 3 above, so that the outputs A, B are **w-XOR(XNOR) compatible**. Similarly, let the outputs B, C be **w-XOR(XNOR) compatible**, and outputs A, C be **w-XOR(XNOR) compatible**. Then (ABC) is **w-XOR(XNOR) compatible** and all faults may or may not be detected at z_1 .

Corollary 9.1 Let A_1, A_2, \dots, A_m be the different outputs of an n -input m -output MUT. Let the faults detected at the MUT outputs A_1, A_2, \dots, A_m be θ where $\theta \leq \beta$, the total number of detectable faults at the MUT output when subjected to a compact set of deterministic tests τ , $\tau \leq 2^n$, τ might not be a minimal or nonminimal but complete set of tests, or to pseudorandom tests. Assume that the fault situation at the outputs A_1, A_2, \dots, A_m conforms to conditions of Case 3 above, so that the outputs A_1, A_2, \dots, A_m are **w-XOR(XNOR) compatible**. Then all faults may or may not be detected at z_1 .

The proofs of all the above theorems follow rather obviously and hence are omitted here in the paper.

In actual situations, we do not know (and also almost impossible to know) whether the merged outputs conform to conditions specified by Cases 1-3 as discussed, and as such we have to deal exclusively with the case of **simply compatible**.

3. Experimental Results

To demonstrate the feasibility of the proposed zero-aliasing space compaction schemes, independent simulations were conducted on various ISCAS 85 combinational and ISCAS 89 full-scan sequential benchmark circuits. Approaches based on heuristics have been adopted and implemented so that we can get the results within an acceptable CPU time. We used ATALANTA, FSIM, and HOPE (Lee and Ha, 1991, 1992, 1993, Pomeranz *et al.*, 1991) as test generation tools to generate the deterministic and pseudorandom test vectors needed to test the benchmark circuits using reduced or compacted test sets accompanied with a random testing session. For each circuit, we determined the number of test vectors used to construct the compaction tree, simulation CPU time, number of applied test vectors, and percentage fault coverage.

Because of space constraints, the details of the pseudocode descriptions of the algorithms and many other implementation details and examples could not be given; however, only some results on ISCAS 85 combinational and ISCAS 89 full-scan sequential benchmark circuits are given below. The overall procedures, in short, include:

- algorithm to compute all the incompatible pairs of the MUT output lines;
- algorithm to find all the maximal compatibility classes (MCs) from the MUT response data outputs, corresponding to possible merger by AND(NAND), OR(NOR) and XOR(XNOR) gates, viz. MCs(AND/NAND), MCs(OR/NOR) and MCs(XOR/XNOR), based on knowledge of pairs of incompatibles, using the modified cut-set algorithm by Das (1973); and
- final algorithm to generate the zero-aliasing compaction trees.

The procedures, in brief, could be simply stated as follows:

A maximal compatibility class comprised of more than one MUT response data outputs (possibly an MC class with the largest cardinality) is randomly picked up first, and the corresponding hardware, COMPACTOR, is added to the MUT output. The stuck-at-logic faults are then injected into the MUT, and input test vectors are applied to the MUT and COMPACTOR. The response patterns are observed and compared with the corresponding fault-free responses. If all injected faults are detected at the

output of the MUT and COMPACTOR, then the maximal compatibility class selected will be retained (to ensure optimal generalized sequence mergeability), and the procedure will be repeated until a compaction tree with a maximal compaction ratio is generated. Otherwise, another maximal compatibility class of the MUT response data outputs (possibly any of the remaining MC classes with the largest cardinality) or a subset thereof will be randomly picked up and the process will be carried out all over again.

As far as the experimental results are concerned, Tables 1-3 show the number of maximal compatibility classes, fault coverage, and hardware overhead, respectively, for all ISCAS 89 full-scan sequential benchmark circuits, to generate aliasing-free compactors under optimal generalized sequence mergeability, using deterministic compacted test sets, while Tables 4-6 show the number of maximal compatibility classes, fault coverage, and hardware overhead, respectively, for all ISCAS 89 benchmark circuits, to generate aliasing-free compactors under optimal generalized sequence mergeability, using pseudorandom test patterns. Figures 1-3, on the other hand, show, respectively, the number applied input test patterns, compaction ratio, and area overhead for all ISCAS 85 combinational benchmark circuits to generate aliasing-free compactors under optimal generalized sequence mergeability, using deterministic compacted test sets, and Figures 4-6 show, respectively, the number applied input test patterns, compaction ratio, and area overhead to generate aliasing-free compaction networks for the benchmark circuits under optimal generalized sequence mergeability, using pseudorandom testing.

Figures 7-8 show two single-line compactors for the benchmark circuit c432 using deterministic compacted testing, while Figures 9-10 show two such single-line space compactors for c432 using pseudorandom testing.

Table 1 Compatibility Classes Computed at the First Compaction Stage Using Deterministic Compacted Testing

Circuit name	No of input lines in MUT	No of output lines in MUT	No of gates in MUT	No of AND incompatible pairs used to determine the compatibility classes	No of OR incompatible pairs used to determine the compatibility classes	No of XOR incompatible pairs used to determine the compatibility classes	No of AND maximal compatibility classes	No of OR maximal compatibility classes	No of XOR maximal compatibility classes
s27	4	1	17	0	0	0	1	1	1
s208	11	2	115	1	0	0	2	1	1
s298	3	6	136	15	10	2	6	4	4
s713	35	23	447	101	165	1	70	46	2
s838	35	2	457	1	0	0	2	1	1
s953	16	23	440	175	133	134	11	512	182
s1196	14	14	561	91	53	3	14	27	8
s1238	14	14	540	91	53	3	14	27	8
s1488	8	19	667	171	110	3	19	49	8
s1494	8	19	661	171	110	3	19	49	8
s5378	35	49	2993	433	1037	72	170	112	161
s9234	36	39	5844	263	497	208	67	76	71

Table 2 Fault Coverage for ISCAS 89 Benchmark Circuits Using Deterministic Compacted Test Patterns for Input Test Vector Generation

Circuit name	No of input lines in MUT	No of output lines in MUT	No of gates in MUT	No of injected faults	No of applied input test vectors	No of output lines in MUT and COMPACTOR	No of levels in COMPACTOR	Compaction ratio	CPU simulation time (secs)	Fault coverage (%) (MUT and COMPACTOR)
s27	4	1	17	32	44	1	0	1/1	0.033	100.00
s208	11	2	115	91	670	1	1	1/2	0.167	100.00
s298	3	6	136	184	571	2	1	2/6	0.167	100.00
s713	35	23	447	921	459	1	1	1/23	0.633	100.00
s838	35	2	457	187	269	1	1	1/2	0.233	100.00
s953	16	23	440	81	248	2	2	2/23	0.300	100.00*
s1196	14	14	561	1026	964	1	3	1/14	1.167	100.00
s1238	14	14	540	1037	964	1	2	1/14	1.150	100.00
s1488	8	19	667	775	786	1	2	1/19	0.617	100.00*
s1494	8	19	661	805	786	2	2	2/19	0.683	100.00*
s5378	35	49	2993	2180	951	3	3	3/49	6.650	100.00*
s9234	36	39	5844	357	50	3	2	3/39	0.817	100.00*

Note: * indicates submaximal compatibility classes were used in constructing the compressor.

Table 3 Estimates of Hardware Overhead for Deterministic Compacted Testing

Circuit name	No of fanin in COMPACTOR	No of gates in COMPACTOR	Average fanin in COMPACTOR	No of gates in MUT	Average fanin in MUT	No of gates in MUT and COMPACTOR	Area overhead (%)
s27	0	0	-	17	1.80	17	-
s208	2	1	2.00	115	1.72	116	1.00
s298	6	2	3.00	136	2.05	138	2.12
s713	23	1	23.00	447	1.50	448	3.42
s838	2	1	2.00	457	1.72	458	0.25
s953	25	4	6.25	440	1.88	444	2.99
s1196	17	4	4.25	561	1.91	565	1.57
s1238	15	2	7.50	540	2.05	542	1.35
s1488	21	3	7.00	667	2.12	670	1.47
s1494	21	4	5.25	661	2.15	665	1.46
s5378	52	5	10.40	2993	1.52	2998	1.14
s9234	43	6	7.16	5844	1.42	5850	0.51

Table 4 Compatibility Classes Computed at the First Compaction Stage Using Pseudorandom Testing

Circuit name	No of input lines in MUT	No of output lines in MUT	No of gates in MUT	No of AND incompatible pairs used to determine the compatibility classes	No of OR incompatible pairs used to determine the compatibility classes	No of XOR incompatible pairs used to determine the compatibility classes	No of AND maximum compatibility classes	No of OR maximum compatibility classes	No of XOR maximum compatibility classes
s27	4	1	17	0	0	0	1	1	1
s208	11	2	115	1	0	0	2	1	1
s298	3	6	136	15	11	1	6	4	2
s713	35	23	447	41	92	0	89	39	1
s838	35	2	457	1	0	0	2	1	1
s953	16	23	440	175	133	134	11	512	182
s1196	14	14	561	68	20	0	24	47	1
s1238	14	14	540	74	25	1	22	53	2
s1488	8	19	667	171	134	3	19	27	4
s1494	8	19	661	171	134	3	19	27	4
s5378	35	49	2993	2753	1929	541	3121	4258	253
s9234	36	39	5844	347	605	279	93	163	79

Table 5 Fault Coverage for ISCAS 89 Benchmark Circuits Using Pseudorandom Test Patterns for Input Test Vector Generation (Initial Seed for the Random Number Generator: 1)

Circuit name	No of input lines in MUT	No of output lines in MUT	No of gates in MUT	No of injected faults	No of applied input test vectors	No of output lines in MUT and COMPACTOR	No of levels in COMPACTOR	Compaction ratio	CPU simulation time (secs)	Fault coverage (%) (MUT and COMPACTOR)
s27	4	1	17	32	85	1	0	1/1	0.033	100.00
s208	1	2	115	110	813803	1	1	1/2	55.633	100.00
s298	3	6	136	222	2000000	1	2	1/6	135.733	92.342
s298	3	6	136	222	1095619	1	3	1/6	95.667	100.00*
s713	35	23	447	474	824562	1	2	1/23	181.967	100.00*
s838	35	2	457	218	1185848	1	1	1/2	176.367	100.00
s953	16	23	440	81	248	2	2	2/23	0.067	100.00*
s1196	14	14	561	1238	1407989	1	3	1/14	531.850	100.00
s1238	14	14	540	1284	1407989	1	2	1/14	621.750	100.00*
s1488	8	19	667	1198	1954247	1	2	1/19	879.850	100.00*
s1494	8	19	661	1222	1954247	2	2	2/19	712.500	100.00*
s5378	35	49	2993	2929	1847028	2	3	2/49	3138.183	100.00*
s9234	36	39	5844	352	2000000	2	4	2/39	29440.617	100.00*

Note: * indicates submaximal compatibility classes were used in constructing the compressor.

Table 6 Estimates of Hardware Overhead for Pseudorandom Testing

Circuit name	No of fanin in COMPACTOR	No of gates in COMPACTOR	Average fanin in COMPACTOR	No of gates in MUT	Average fanin in MUT	No of gates in MUT and COMPACTOR	Area overhead (%)
s27	0	0	-	17	1.80	17	-
s208	2	1	2.00	115	1.72	116	1.00
s298	8	3	2.66	136	2.05	139	2.80
s298	9	4	2.25	136	2.05	140	3.13
s713	25	3	8.33	447	1.50	450	3.70
s838	2	1	2.00	457	1.72	458	0.25
s953	25	4	6.25	440	1.88	444	2.99
s1196	18	5	3.60	561	1.91	566	1.66
s1238	15	2	7.50	540	2.05	542	1.35
s1488	21	3	7.00	667	2.12	670	1.47
s1494	20	3	6.66	661	2.15	664	1.39
s5378	53	6	8.83	2993	1.52	2998	1.16
s9234	45	7	6.42	5844	1.42	5851	0.54

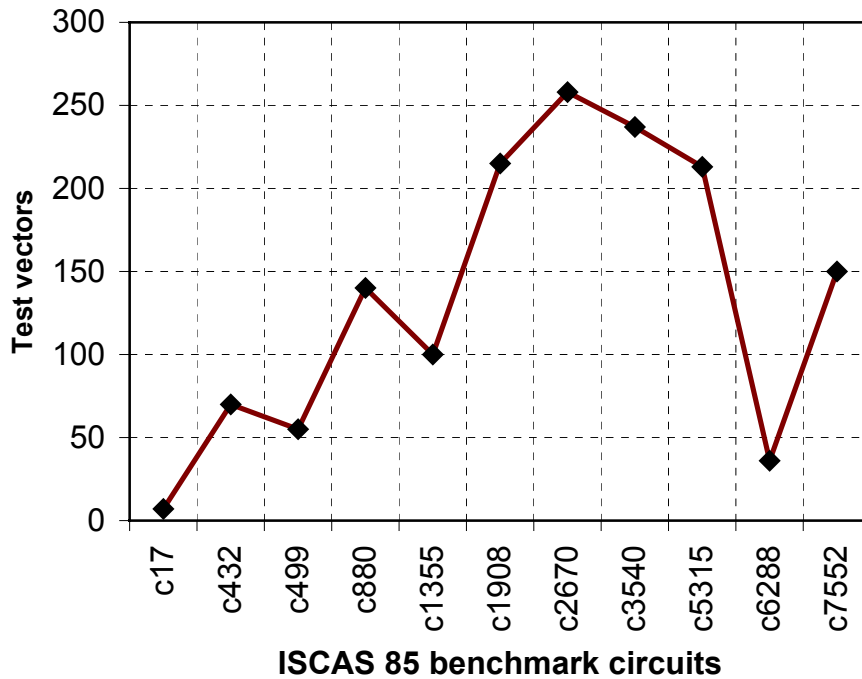


Fig. 1 Input test vectors for deterministic compacted testing

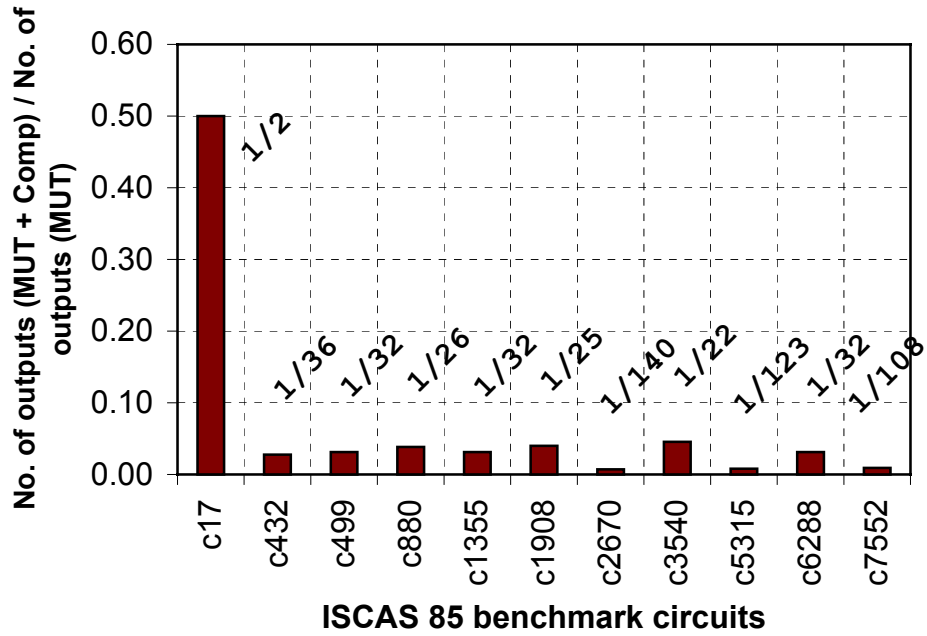


Fig. 2 Compaction ratio for deterministic compacted testing

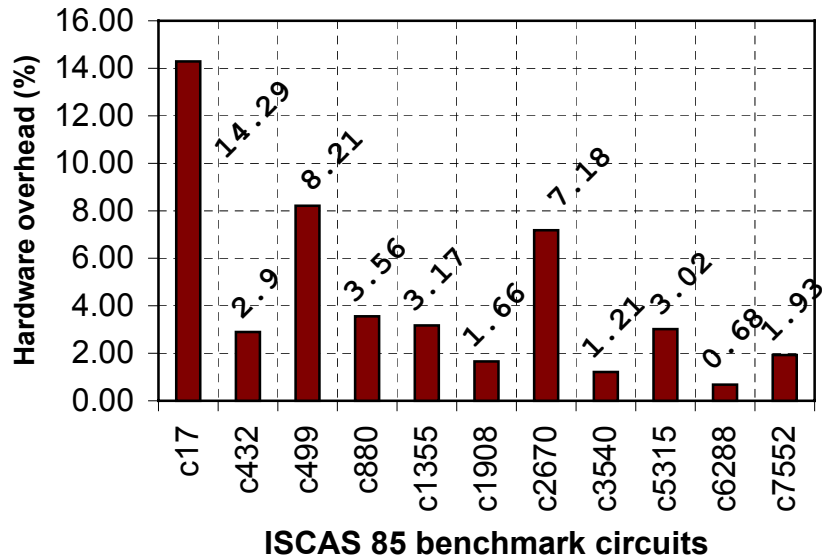


Fig. 3 Estimates of hardware overhead for deterministic compacted testing

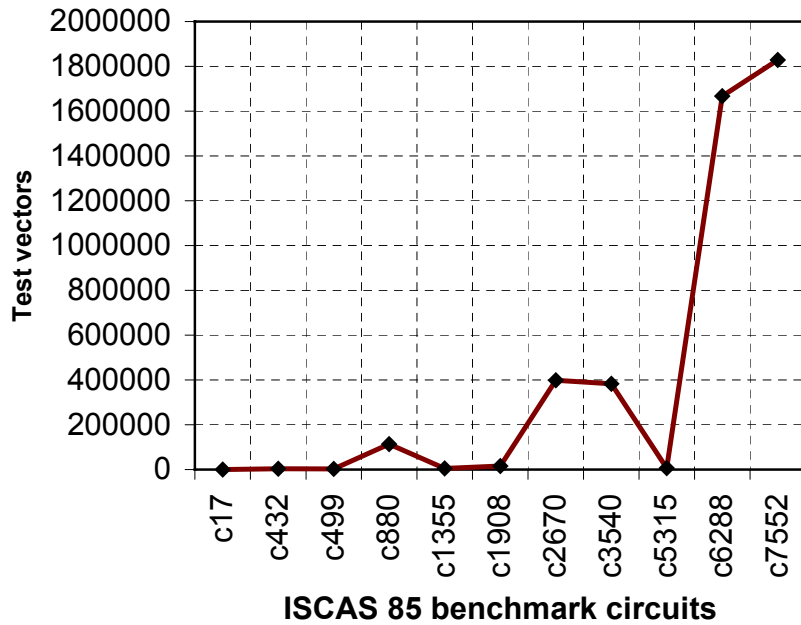


Fig. 4 Input test vectors for pseudorandom testing

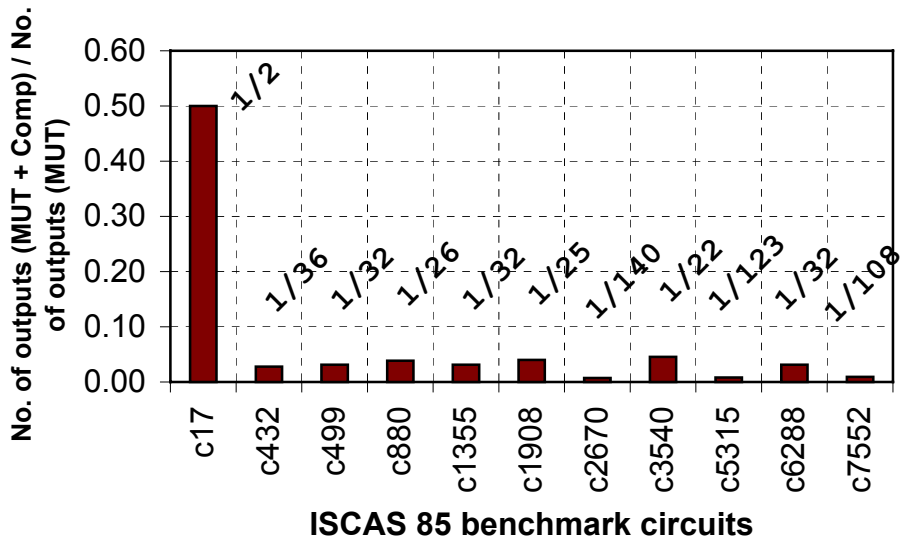


Fig. 5 Compaction ratio for pseudorandom testing

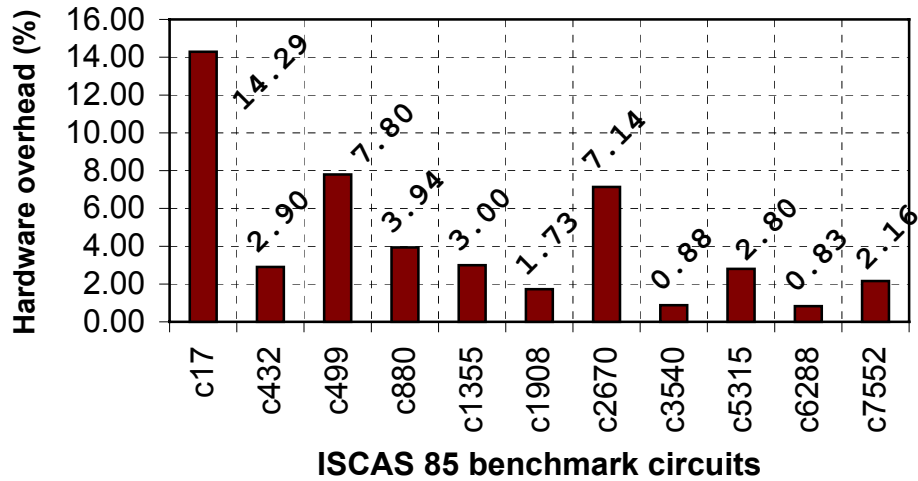


Fig. 6 Estimates of hardware overhead for pseudorandom testing

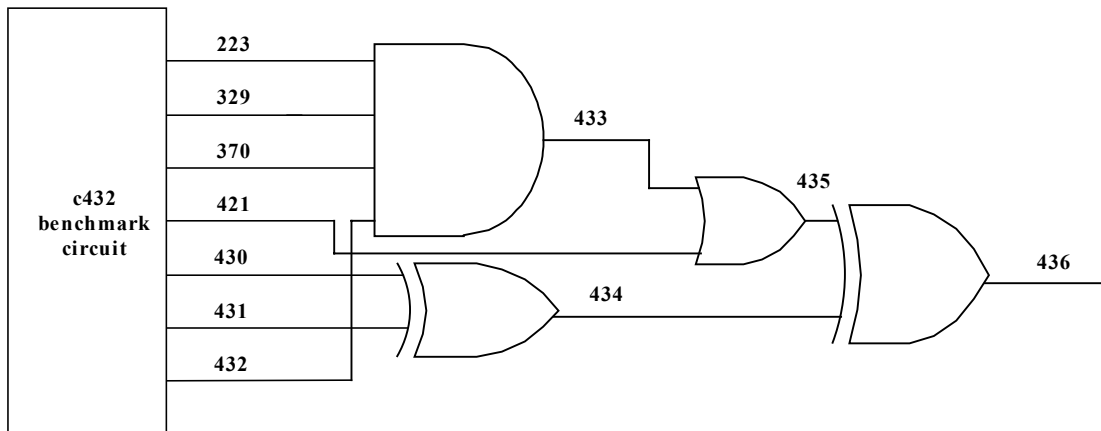


Fig. 7 A single-line compactor circuit for c432 using deterministic compacted testing

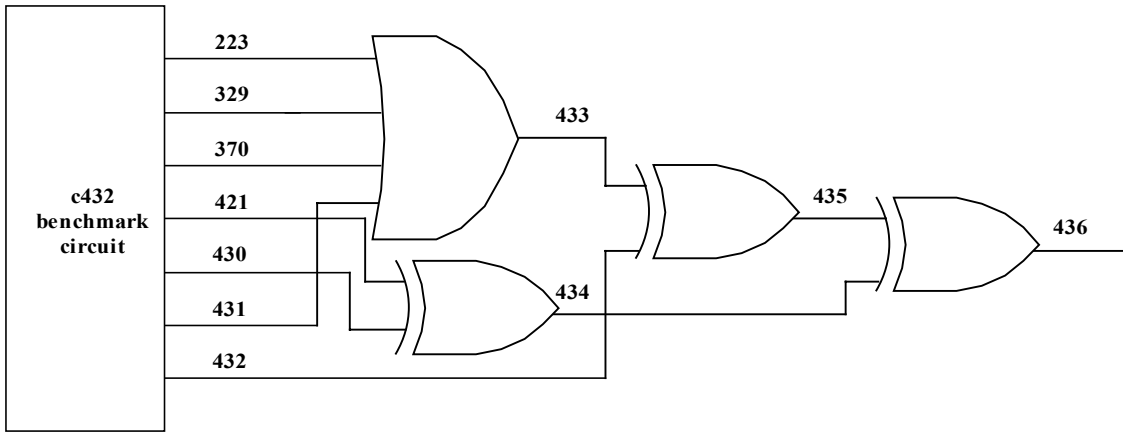


Fig. 8 Another single-line compactor circuit for c432 using deterministic compacted testing

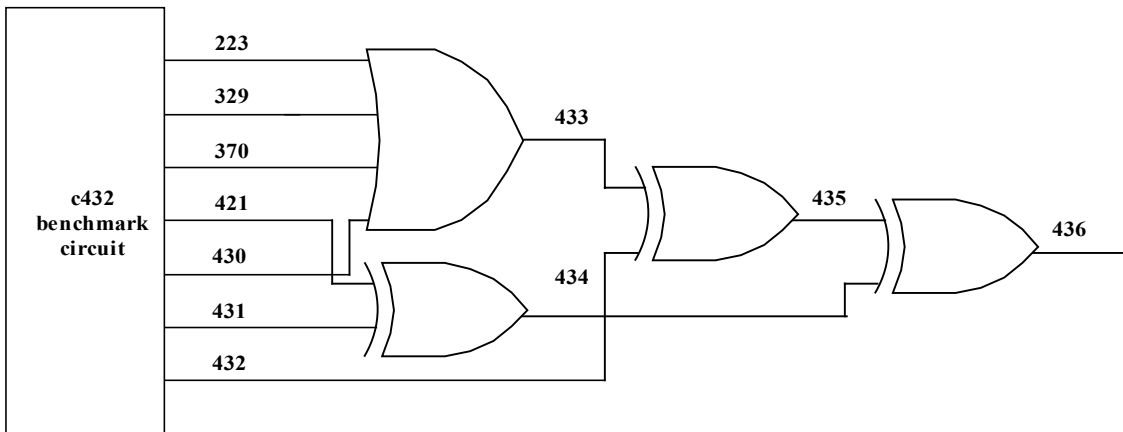


Fig. 9 A single-line compactor circuit for c432 using pseudorandom testing

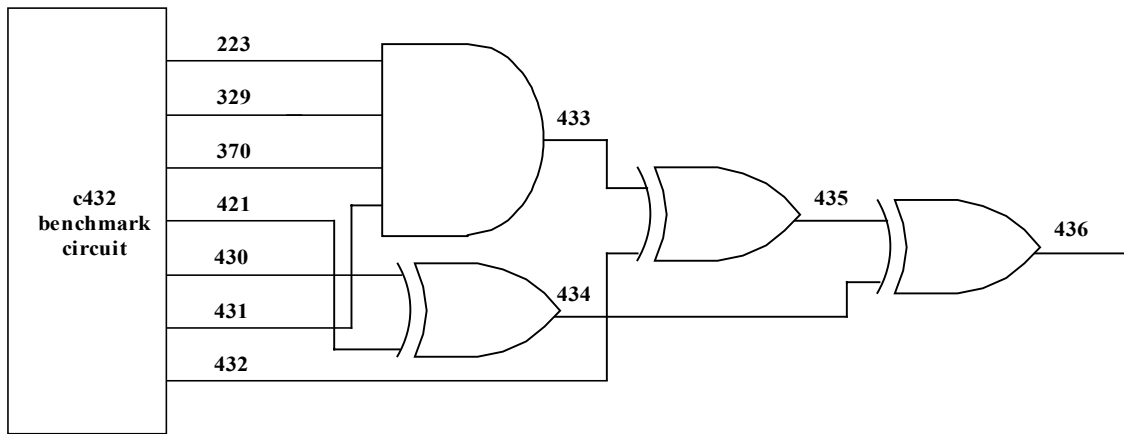


Fig. 10 Another single-line compactor circuit for c432 using pseudorandom testing

4. Conclusions

The design of space-efficient BIST support hardware is of great importance in the synthesis and fabrication of ICs. The present paper reports zero-aliasing compression techniques of test data outputs for digital embedded cores-based system-on-chips which facilitate the design of such space-efficient support hardware. The suggested techniques take advantage of some well-known concepts of conventional switching theory, particularly those of cover table and frequency ordering as commonly utilized in the minimization of switching functions, together with those of strong and weak compatibilities of response data outputs, in the selection of specific gates for merger of an arbitrary but optimal number of output bit streams from the MUT. They are novel in the sense that zero-aliasing is achieved without any modification of the MUT, while maximal compaction is achieved in most cases in reasonable time utilizing some simple heuristics. The techniques, illustrated with details of design of space compactors for ISCAS 85 combinational and ISCAS 89 full-scan sequential benchmark circuits with ATALANTA, FSIM, and HOPE simulation programs, confirm the usefulness of the suggested approaches, their simplicity, resulting low area overhead, and full fault coverage for single stuck-line faults, making them suitable in a VLSI design environment as BIST support hardware. With advances in computational resources in future, the heuristics adopted in the design algorithms can be further improved to significantly lower the CPU time.

5. Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A 4750. A preliminary version of the work was presented at the 7th World Conference on Integrated Design and Process Technology held in Austin, TX during December 3-6, 2003.

6. References

- Assaf, M. H., 2003, "Digital Core Output Test Data Compression Architecture Based on Switching Theory Concepts", Ph.D. Dissertation, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada.
- Bardell, P. H., McAnney, W. H., and Savir, J., 1987, "Built-In Test for VLSI : Pseudorandom Techniques", Wiley Interscience, New York, U.S.A.

- Chakrabarty, K., 1995, "Test Response Compaction for Built-In Self-Testing", Ph.D. Dissertation, Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI, U.S.A.
- Das, S. R., 1973, "On a New Approach for Finding All the Modified Cut-Sets in an Incompatibility Graph", IEEE Transactions on Computers, Vol. C-22, pp. 187-193.
- Das, S. R., Assaf, M. H., Petriu, E. M., Jin, L., Jin, C., Biswas, D., and Sahinoglu, M., 2004, "Testing Embedded Cores-Based System-on-a-Chip (SoC) – Test Architecture and Implementation", Proceedings of the 23rd IASTED International Conference on Modeling, Identification and Control, pp. 300-306.
- Das, S. R., Assaf, M. H., Petriu, E. M., Jone, W. -B., and Chakrabarty, K., 2001, "A Novel Approach to Designing Aliasing-Free Space Compactors Based on Switching Theory Formulation", Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Vol. 1, pp. 198-201.
- Das, S. R., Ramamoorthy, C. V., Assaf, M. H., Petriu, E. M., and Jone, W. -B., 2001, "Fault Tolerance in Systems Design in VLSI Using Data Compression Under Constraints of Failure Probabilities", IEEE Transactions on Instrumentation and Measurement, Vol. 50, pp. 1725-1747.
- Garey, M. R. and Johnson, D. S., 1979, "Computers and Intractability : A Guide to the Theory of NP-Completeness", Freeman, San Francisco, CA, U.S.A.
- Jone, W. -B. and Das, S. R., 1991, "Space Compression Method for Built-In Self-Testing of VLSI Circuits", International Journal of Computer Aided VLSI Design, Vol. 3, pp. 309-322.
- Karpovsky, M. and Nagvajara, P., 1990, "Optimal Robust Compression of Test Responses", IEEE Transactions on Computers, Vol. C-39, pp. 138-141.
- Lee, H. K. and Ha, D. S., 1991, "An Efficient Forward Fault Simulation Algorithm Based on the Parallel Pattern Single Fault Propagation", Proceedings of the International Test Conference, pp. 946-955.
- Lee, H. K. and Ha, D. S., 1992, "HOPE : An Efficient Parallel Fault Simulator for Synchronous Sequential Circuits", Proceedings of the Design Automation Conference, pp. 336-340.
- Lee, H. K. and Ha, D. S., 1993, "On the Generation of Test Patterns for Combinational Circuits", Technical Report 12-93, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, U.S.A.
- Li, Y. K. and Robinson, J. P., 1987, "Space Compression Method with Output Data Modification", IEEE Transactions on Computer-Aided Design, Vol. 6, pp. 290-294.
- McCluskey, E. J., 1985, "Built-In Self-Test Techniques", IEEE Design & Test of Computers, Vol. 2, pp. 21-28.
- Mourad, S. and Zorian, Y., 2002, "Principles of Testing Electronic Systems", Wiley, New York, U.S.A.
- Pomeranz, I., Reddy, L. N., and Reddy, S. M., 1991, "COMPACTEST : A Method to Generate Compact Test Sets for Combinational Circuits", Proceedings of the International Test Conference, pp. 194-203.
- Pouya, B. and Touba, N. A., 1998, "Synthesis of Zero-Aliasing Elementary-Tree Space Compactors", Proceedings of the VLSI Test Symposium, pp. 70-77.
- Pradhan, D. K. and Gupta, S. K., 1991, "A New Framework for Designing and Analyzing BIST Techniques and Zero Aliasing Compression", IEEE Transactions on Computers, Vol. C-40, pp. 743-763.
- Rajsuman, R., 2000, "System-on-a-Chip : Design and Test", Artech House, Boston, MA, U.S.A.
- Reddy, S. M., Saluja, K. K., and Karpovsky, M. G., 1988, "Data Compression Technique for Test Responses", IEEE Transactions on Computers, Vol. C-37, pp. 1151-1156.
- Sahinoglu, M., 2003, "An Empirical Bayesian Stopping Rule in Testing and Verification of Behavioral Models", IEEE Transactions on Instrumentation and Measurement, Vol. 52, pp. 1428-1443.
- Sahinoglu, M., Bayrak, C., and Cummings, T., 2002, "High Assurance Software Testing in Business and DoD", Transactions of the SDPS, Vol. 6, pp. 107-114.
- Saluja, K. K. and Karpovsky, M., 1983, "Testing Computer Hardware Through Compression in Space and Time", Proceedings of the International Test Conference, pp. 83-88.
- Savir, J., 1996, "Reducing the MISR Size", IEEE Transactions on Computers, Vol. C-45, pp. 930-938.