Decompression Hardware Determination for Test Volume and Time Reduction through Unified Test Pattern Compaction and Compression

Ismet Bayraktaroglu
Test Technology Group
Sun Microsystems
Sunnyvale, CA 94085
ismet.bayraktaroglu@sun.com

Alex Orailoglu
Computer Science & Engineering Department
University of California, San Diego
La Jolla, CA 92093-0114
alex@cs.ucsd.edu

Abstract

A methodology for the determination of decompression hardware that guarantees complete fault coverage for a unified compaction/compression scheme is proposed. Test cube information is utilized for the determination of a near optimal decompression hardware. The proposed scheme attains simultaneously high compression levels and reduced pattern counts through a linear decompression hardware. Significant test volume and test application time reductions are delivered through the scheme we propose while a highly cost effective hardware implementation is retained.

1 Introduction

The level of integration capabilities achievable today, thanks to Moore's Law, enable transistor counts in excess of hundreds of millions. Due to increased design complexity, utilization of sequential or functional test patterns as a primary way of achieving high structural fault coverage is almost completely abandoned. Consequently, near full scan methodologies have become the primary method of achieving high structural coverage due to the ease of implementation and high fault coverage level attainment.

As integration capabilities keep increasing with no sign of any slowdown, test pattern generation for scan-based designs has been able to keep up with dramatic transistor counts. However, the corresponding increase in test sizes has been pushing test costs up due to both the increase in test application time and the need for higher cost ATEs with high pins counts and high memory bandwidth. Various methodologies have been proposed to reduce both test data volume and test application time. The main trend in reducing both test application time and test data volumes has been the utilization of compression schemes in conjunction with on chip test pattern decompression hardware.

While various compression methodologies, including run-length coding, frequency-directed coding, and LFSR-based reseeding methodologies have been proposed, commercially only a subset of such schemes have found applicability. Commonly, test application and test data volume reduction in commercially viable schemes is attained by basically utilizing a higher number of internal scan chains driven through a smaller number of external scan chains. A decompression network between the external and internal scan chains, based on a simple interconnect, a linear XOR-based network, or an LFSR based approach, maps the external scan chain inputs to the internal scan chain inputs. The increase in the number of internal scan chains, with a much smaller increase in test pattern counts, results in both reduced test application times and test data volume.

While higher internal scan chain counts deliver significant benefits, additional constraints imposed by the decompression network may cause certain faults to become redundant. Alternative access mechanisms have been proposed to preserve the fault coverage levels of the decompression scheme. We propose in this work a methodology for the generation of the decompression network that causes no faults to become redundant, and hence necessitates no costly alternative access mechanisms to the scan chains.

In this work, an initial test pattern set is employed for the determination of the decompression network. While the ratio of the internal to external scan chain counts determines the compression ratios, achieving high compression at the cost of a sacrifice in compaction levels results in higher test data volume and application times. Therefore, we investigate a methodology that aims at achieving both high compression and compaction levels.

Section 2 provides a summary of the previous work in this area. Section 3 outlines the decompression architecture assumed and the test pattern generation algorithm employed in this work. Section 4 provides the detailed description of

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

how the deterministic decompression network is created by employing a set of test patterns. While section 5 provides experimental results, section 6 summarizes the significance of the results achieved in this work.

2 Previous Work

A number of schemes have been proposed for test data volume reduction of scan-based deterministic test by improving the effectiveness of test compaction [1, 3, 8, 16] and compression schemes [2, 5, 9, 11, 12, 13, 18, 19]. While both compaction and compression schemes aim at reducing test data volume, the way that they attack the problem of test data volume reduction differs. While compaction schemes aim at reducing the number of test patterns by increasing the number of faults detected by each vector, compression schemes aim at reducing the number of bits required to represent each test vector.

Both compaction and compression schemes have been widely researched. Since in this work we propose a compression scheme that aims at reducing the test data volume for scan-based deterministic patterns, we provide a summary of the previous research in this area.

The scan vector compression schemes proposed so far either operate on a given test set or interact with the test generation algorithm to better utilize unspecified bits in the test cubes. Compression schemes that operate on a set of test vectors utilize various coding schemes such as run-length coding [12] and statistical coding [11, 5], or employ a synthesized combinational circuit [18]. While compression of the original test vectors provides some level of test volume reduction, a number of these schemes also try to compress the difference between the successive vectors to improve compression efficiency [12, 11, 5]. Such schemes operate directly on test vectors and therefore necessitate no modifications to the test pattern generation process. A set of alternative compression schemes, on the other hand, work in conjunction with test pattern generation. Such schemes include driving multiple scan chains with the same inputs [9] and driving internal scan chain inputs through an XOR-based decompression network [2], a ring generator [17], or an LFSR based decompression hardware [13].

Touba and McCluskey proposed the use of a transformation circuitry from a sequence of pseudo-random patterns to a set of deterministic patterns [20]. While the proposed scheme provides high fault coverage with a relatively small number of patterns in a BIST environment, subsequent research has proposed the use of a transformation circuitry for input width reduction [6, 4, 10], hence enabling utilization of pseudo-exhaustive testing. Even though the latter schemes result in higher number of patterns, their hardware overhead is significantly smaller than that of the former scheme. Width reduction for BIST has been achieved by merging directly and inversely compatible inputs [6], by merging decoder(d)-compatible inputs [4], and finally by merging Combinational(C)-compatible inputs [10]. While d-compatible inputs constrain all vectors to have no more than one '1', C-compatibility necessitates that an input can be driven by a combinational function of other inputs.

3 Unified Compaction & Compression

In this section, we briefly describe the proposed on-chip decompression hardware, its effect on test application time and test data volume, and the unified compaction/compression algorithm employed in this work.

Test application time and test data volume is affected by a number of parameters, such as the number of patterns, P, the number of scan chains, S, the number of scan cells, N, and the scan frequency, F. Test application time, D, and test data volume, V, can be determined by the following equations if the scan chains are assumed to be perfectly balanced.

$$D = P \times (\lceil N/S \rceil + 1) \times F^{-1} \quad (1)$$
$$V = P \times \lceil N/S \rceil \times S \quad (2)$$

Let's assume that through a decompression hardware, the number of internal scan chains is increased to S_i and the number of test patterns increases slightly to P_c. Figure 1 depicts a possible implementation of the decompression scheme, in which the S bit data provided by the tester is decompressed to generate S_i bit data to the internal scan chains at every clock cycle.

In this case, the new test data volume and test application time can be calculated by the following equations.

$$D_c = P_c \times (\lceil N/S_i \rceil + 1) \times F^{-1} \quad (3)$$
$$V_c = P_c \times \lceil N/S_i \rceil \times S \quad (4)$$

The percentage reduction due to compression in test application time and test data volume can be computed as $(D - D_c)/D$ and $(V - V_c)/V$, respectively. Equation 5 provides an approximation for both test data volume and

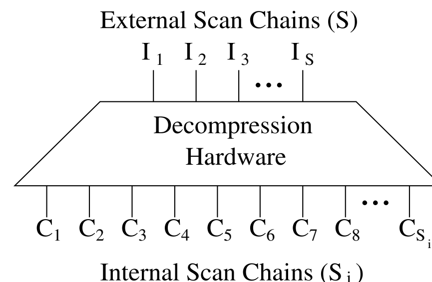


Figure 1. Decompression Hardware

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

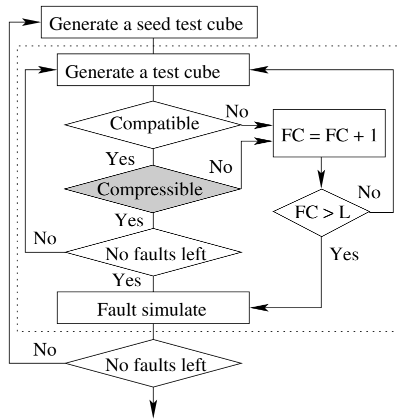


Figure 2. Compaction & Compression

test application time reduction.

$$1 - \frac{S}{S_i} \times \frac{P_c}{P} \tag{5}$$

Figure 2 depicts the unified compaction/compression algorithm employed in this work. The algorithm is an augmented version of the one used in [2]. The algorithm starts with a seed test cube and merges compatible test cubes to the seed. In case a test cube is not compatible, a *fail counter*, FC, is increased and another test cube is generated. If the test cube is compatible, it is temporarily merged with the seed test cube and the resultant test cube is checked for compressibility. If it is not compressible, the fail count is increased. The algorithm performs fault simulation when the fail count exceeds a predetermined limit, L, and continues by generating a new seed test cube. The fail counter is added in this work in order to reduce the number of test cubes that need to be generated and hence reduces the time complexity of the algorithm.

4. Problem Definition

We start our description of the proposed method by defining the variables utilized in describing the decompression hardware determination problem. The variables that are already defined in the previous section are used in these definitions with no modification.

Definitions:

- $K = \lceil N/S_i \rceil$: The number of scan cells in the longest scan chain
- $C = [C_0, C_1, ..., C_{S_i}]$: The internal scan inputs
- $C(p,k)$: The value of internal scan cell inputs corresponding to the kth internal scan cell of the pth pattern of the initial test pattern set
- $I = [I_1, I_2, ..., I_S]$: The external inputs
- $I(p,k)$: The value of the external inputs I corresponding to the kth scan cell of the pth pattern. $I(p,k)$ are the unknowns of the compression algorithm
- $F(I) = [F_0(I), F_1(I), \cdots, F_{S_i}(I)]$: The set of functions defining the output of the decompression network

The decompression network defined by F needs to ensure that the $C = F(I)$ assignment does not result in any redundant faults. While infinitely many solutions may exist to this problem, we in this work utilize a solution that is based on a given set of initial test cubes. If the initial set of test cubes can be generated by the decompression hardware, defined by F, the decompression hardware will not result in any redundant faults, provided that the initial test cube set detects all irredundant faults.

Under this assumption, the problem can be reduced to the following. The equation $C(p,k) = F(I(p,k))$ is solvable for $I(p,k)$. A further simplification of the problem may be achieved, if the set of functions F is limited to the set of linear functions defined by the XOR gates. In this case, the equations can be formulated as the following matrix multiplication.

$$\forall p,k;\ C(p,k) = D_{S_i,S} I(p,k) \tag{6}$$

wherein $D_{S_i,S}$ is matrix whose elements are composed of 0's and 1's, the multiplication operation is replaced by a logical AND operation and the addition operation is replaced by an XOR operation. Furthermore, we partition the problem into two subproblems.

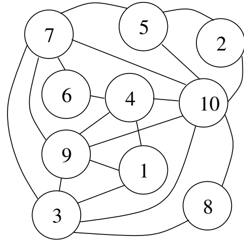
- Given the set of initial test cubes, $C(p,k)$, partition the internal scan inputs, C_i, into compatibility sets.
- Given the compatibility sets, create a decompression network that satisfies certain properties. The properties that need to be satisfied depend on the compatibility definition.

We will initially utilize the compatibility condition that is traditionally utilized in test cube compaction. The two internal scan chain inputs are compatible, if $C_i(p,k)$ and $C_j(p,k)$ are identical or at least one of the two is unspecified for all p,k. By checking the compatibility between all pairs of the internal scan chain inputs, we can create a compatibility graph, and the minimum number of cliques in this graph defines the number of compatibility sets while the vertices involved in a clique constitute the elements of the compatibility sets. Equivalently, the chromatic number of the incompatibility (conflict) graph provides the number of compatibility sets. We, subsequently, formally define this problem.

Definition: We define a conflict graph with S_i nodes, each node corresponding to an internal scan chain, $G(V,E)$, with an edge E_{ij} between the nodes V_i and V_j if and

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY



$S_1 = \{1, 6, 10\}$
$S_2 = \{2, 7, 8\}$
$S_3 = \{3, 4, 5\}$
$S_4 = \{9\}$

Figure 3. Conflict Graph

only if $\exists p, k$ such that $(T_i(p,k) = 1 \wedge T_j(p,k) = 0) \vee (T_i(p,k) = 0 \wedge T_j(p,k) = 1)$. The time complexity of the generation of the conflict graph is $O(P * K * S_i^2)$.

To better illustrate the formulation outlined in this section, we utilize an imaginary compatibility graph for $S_i = 10$ in figure 3. The chromatic number of this conflict graph provides the number of compatibility sets, while the sets of inputs with the same color correspond to the sets of compatible inputs. While the graph coloring problem is known to be NP-complete, simple heuristics [7] that provide near minimal color counts exist. The compatibility sets for the conflict graph of figure 3 are also provided on the figure. It can be trivially observed that the conflict graph has a chromatic number of 4.

Subsequent to the identification of the compatibility sets, the condition that needs to be satisfied by F in terms of the compatibility sets needs to be identified. Let S_c denote the compatibility sets of a conflict graph, wherein c ranges between 1 and C, the chromatic number, and let $S_c(i)$ denote the elements of the compatibility sets. As the internal scan chain inputs in distinct compatibility groups may need to have distinct values for a given (p, k) pair, the rows of the matrix D_{S,S_i} corresponding to the internal scan chains in distinct compatibility groups needs to be linearly independent. The condition imposed on the decompression network can be shown to be equivalent to the condition that the set of outputs $S_1(i), S_2(j), \cdots, S_C(z)$ be linearly independent for all possible $i, j, \cdots, z$ combinations. While this condition has direct correlation with the definition of the conflict graph, the compatibility definition utilized in the derivation of the conflict graph imposes one more condition.

Internal scan chain inputs that fall into the same compatibility set may be required to have the same value for some (p, k) pair. Therefore, the decompression hardware, or the set of functions F, needs to assign all inputs in the same compatibility set to the same value as F is independent of the (p, k) pair. Under these conditions, the decompression network degenerates into an interconnect-based network. Such an implementation is similar to the *Parallel Serial Full Scan* (PSFS) implementation discussed in [9]. The decompression network corresponding to the conflict graph in figure 3 is depicted in figure 4.

$C_1 = I_1$ $C_6 = I_1$ $C_{10} = I_1$
$C_2 = I_2$ $C_7 = I_2$ $C_8 = I_2$
$C_3 = I_3$ $C_4 = I_3$ $C_5 = I_3$
$C_9 = I_4$

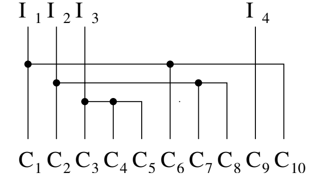


Figure 4. Interconnect-Based Decompression

While such an interconnect-based decompression network is capable of compressing test cubes, the assignment of multiple internal scan chain inputs to a single external input reduces the compaction capability significantly. For example, if two test cubes to be compacted have conflicting specified values in internal input positions C_1 and C_{10} or C_7 and C_8 as shown in figure 5, they cannot be compacted,.

While such a deterministic decompression hardware guarantees that all the test cubes are encodable, hence introducing no redundant faults, the repetition of the inputs introduces high levels of linear dependencies to the decompression network. The introduced linear dependencies, though they do not generate any redundant faults, reduce the *compaction* efficiency of the proposed compression algorithm significantly, resulting in a higher number of test patterns, as confirmed by the results provided in section 5. We examine the possibility, consequently, of an alternative solution that does not result in such a high number of linearly dependent input combinations.

The traditional compatibility condition is a “relaxed” condition in that two specified inputs are assumed to be compatible if they have the same value. Due to this condition, the compatible inputs in the compatibility sets need to be exactly the same, restricting the decompression network to be nothing but an interconnect-based network.

If a decompression network is to allow higher levels of *compaction* with a possible reduction in compression ratio, we must be able to set the scan chain inputs in a compatibility group to distinct values. However, doing so should result in no violation in the linear independence condition mentioned before.

In order for the elements of a compatibility set to be set independently, at most one of the elements of a compatibility set should be specified. If two internal scan chains are assumed to be incompatible whenever they require a specified value for some (p, k) pair, only one of the elements in the compatibility sets will require a specified value. With only a slight modification to the compatibility definition,

$$\begin{array}{rcl} & & \quad C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7\ C_8\ C_9\ C_{10} \\ C(p_1,k) & = & [\ 1\ \ 0\ \ x\ \ 1\ \ x\ \ 1\ \ 0\ \ x\ \ 0\ \ x\ ] \\ C(p_2,k) & = & [\ x\ \ x\ \ 0\ \ x\ \ 0\ \ x\ \ x\ \ 1\ \ x\ \ 0\ ] \\ Merged & = & [\ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ ] \end{array}$$

Figure 5. Compaction Example

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

the previous solution can be used to attain the compatibility sets. However, in this case, the chromatic color of the conflict graph will be higher due to a stricter compatibility definition. With the modified definition of compatibility, the only condition that remains to be satisfied in the second part of the problem is the linear independence condition.

Let S_c's be sorted according to their cardinality. We are going to prove that a network that is constructed in the following way is capable of satisfying the linear independence condition. The elements of S_1 are connected to I_1, $I_1 \oplus I_2$, $\cdots$, $I_1 \oplus I_{|S_1|-1}$. The elements of I_c are connected to I_c, $I_c \oplus I_{c+1}, \cdots, I_c \oplus I_{c+|S_c|-1}$.

We are going to prove by contradiction that the linear independence condition is satisfied by this hardware. Let's assume that $S_\alpha(i), S_\beta(j), \cdots, S_\gamma(k)$ are linearly dependent and $1 \leq \alpha < \beta < \cdots < \gamma \leq C$. Linear dependence necessitates that $S_\alpha(i)$ be representable by the linear combination of the rest of the outputs. Due to the construction of the decompression network, the input I_α can be utilized by an output only if the output is in $S_c, c \leq \alpha$. Therefore, a linear combination of the rest of the outputs cannot result in $S_\alpha(i)$, implying that linear dependencies cannot exist in the proposed deterministic hardware implementation.

The proposed 2-input network, given a set of compatibility sets, can be generated if and only if $|S_c| \leq C+1-c, \forall c$, wherein S_c's are sorted in descending order in terms of their cardinality. While this constraint would not be consistently satisfied by a set of compatibility sets generated through graph coloring, the color assignment to the nodes of the incompatibility graph can be modified so as to satisfy the condition. In case the condition cannot be satisfied through color modifications, the chromatic number of the graph needs to be increased until the condition is satisfied.

Figure 6 shows the augmented compatibility classes, corresponding input assignment and the equations for the decompression network attained from the conflict graph in figure 3. Utilization of additional terms, as in the 2-input network, reduces the possibility of linear dependencies, and therefore increases compaction efficiency of the algorithm. A 3-input network can similarly be generated to improve compaction efficiency even further. In case of a 3-input deterministic network, the constraints on the sizes of compatibility sets need to be augmented. The following section provides compression results for both interconnect-based and XOR-based deterministic compression networks.

$S_1 = (2,3,4)$	$O_2 = X_0$	$O_3 = X_0 \oplus X_1$	$O_4 = X_0 \oplus X_2$
$S_2 = (1,6,7)$	$O_1 = X_1$	$O_6 = X_1 \oplus X_2$	$O_7 = X_1 \oplus X_3$
$S_3 = (5,9)$	$O_5 = X_2$	$O_9 = X_2 \oplus X_3$	
$S_4 = (0)$	$O_0 = X_3$		
$S_5 = (8)$	$O_8 = X_4$		

Figure 6. Decompression Network Generation

Circuit	N	S_i	Comp	S	BS	S	NS2	NS3	NS4
s13207	700	175	247	15	350	19	261	**258**	261
s15850	611	153	144	15	308	18	180	**167**	168
s35932	1,763	196	24	**7**	**41**	20	35	34	34
s38417	1,664	185	178	19	533	19	358	330	**317**
s38584	1,464	183	164	14	358	19	209	192	**185**

Table 1. Compaction & Compression Results

5 Experimental Results

The compression algorithm depicted in figure 2 is implemented in C and interfaced with Atalanta [14] for test pattern generation and with Hope [15] for fault simulation. The algorithm for determination of the decompression network is also implemented in C.

Table 1 provides the test pattern count attained for the largest ISCAS89 benchmark circuits. The number of scan cells and internal scan chains are given in columns two and three, respectively. Column four provides the number of patterns required when compaction alone is performed. Columns five and six provide the number of external scan chains required and the number of patterns for the interconnect-based decompression network. While column seven provides the number of external scan chains required for the XOR-based decompression network, the remaining 3 columns provide the number of patterns for 2, 3, and 4-input XOR-based decompression networks.

The results provided in table 1 indicate that while an interconnect-based decompression network necessitates a smaller number of external scan chains and hence results in higher compression ratios, the XOR-based decompression network with lower compression ratios in general outperforms it due to high compaction efficiency. An interconnect-based decompression network is superior only for s35932, which is highly random pattern testable.

Table 2 provides the test application time in terms of the number of cycles and test data volume calculated by equations 1 through 4 for the compaction only case and for the best compression case, which is indicated in boldface in table 1. The results in columns 6 and 7 indicate that reductions in exess of 80% in both test time and test data volume are achievable by the proposed deterministic decompression hardware with no coverage loss whatsoever.

Circuit	D	V	D_c	V_c	%D	%V
s13207	9,386	173,641	1,036	19,608	88.7	88.9
s15850	5,040	88,128	672	12,024	86.3	86.6
s35932	6,072	42,336	315	2,583	93.9	94.8
s38417	15,842	297,616	2,862	54,207	81.8	81.9
s38584	12,956	243,048	1,488	28,120	88.4	88.5

Table 2. Reductions in Test Time and Volume

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

Circuit	Broadcast Scan [9]	FDR Coding [5]	SCC [2]	Proposed	Imp. (%)
s13207	82,546	**20,368**	25,344	19,608	3.7
s15850	76,030	**21,590**	22,784	12,024	44.3
s35932	9,136	20,946	**7,128**	2,583	63.8
s38417	127,932	**57,066**	89,856	54,207	5.0
s38584	129,580	70,328	**38,796**	28,120	27.5
Imp. (%)	73.6	40.1	40.1		

Table 3. Comparison to Prior Work

Table 3 provides a comparison of the test data volumes to the previous work. We provide a comparison to Broadcast scan [9], FDR coding [5], and Scan Chain Concealment (SCC) based on an XOR network [2]. The best test data volume achieved by any of the previous work is indicated in boldface. The results for the proposed scheme are given in column 5. While the final column provides the improvement over the best test volume for each circuit, the final row summarizes the average of the improvements to each prior technique individually, indicating that the proposed scheme outperforms all previously proposed compression schemes even in comparison to the best results in each benchmark and certainly dramatically in individual comparisons.

6 Conclusion

A decompression hardware generation scheme for linear decompression network based test pattern compression is proposed in this work. We investigate the problem of generating a decompression hardware capable of ensuring no fault coverage loss starting from a set of test cubes that detect all faults of interest in the circuit.

An efficient methodology for the determination of a decompression network is outlined. The traditional compatibility definition results in very small external scan chain counts. Nonetheless, the modified compatibility definition delivers decompression networks with higher compaction efficiency, achieving smaller test data volume and test application times.

The experimental results outlined in this work indicate that a compression algorithm with concurrent application of compaction and compression is superior to compression schemes that perform compaction and compression serially. The deterministic nature of the decompression hardware furthermore eliminates the need for alternative access mechanisms that are usually required to handle redundant faults introduced due to the decompression network.

References

[1] B. Ayari and B. Kaminska. A new dynamic test vector compaction for automatic test pattern generation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(3):353–358, March 1994.

[2] I. Bayraktaroglu and A. Orailoglu. Test volume and application time reduction through scan chain concealment. In *IEEE Design Automation Conference*, pages 151–155, June 2001.

[3] S. Bhatia and P. Varma. Test compaction in a parallel access scan environment. In *Asian Test Symposium*, pages 300–305, November 1997.

[4] K. Chakrabarty and B. T. Murray. Design of built-in test generator circuits using width compression. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(10):1044–1051, October 1998.

[5] A. Chandra and K. Chakrabarty. Test resource partitioning for SOCs. *IEEE Design & Test of Computers*, 18(5):80–91, September-October 2001.

[6] C.-A. Chen and S. K. Gupta. A methodology to design efficient BIST test pattern generators. In *IEEE International Test Conference*, pages 814–823, October 1995.

[7] A. H. Gebremedhin and F. Manne. Scalable parallel graph colouring algorithms. *Concurrency: Practice and Experience*, 12(12):1131–1146, October 2000.

[8] I. Hamzaoglu and J. H. Patel. Test set compaction algorithms for combinational circuits. In *IEEE International Test Conference*, pages 283–289, October 1998.

[9] I. Hamzaoglu and J. H. Patel. Reducing test application time for full scan embedded cores. In *International Symposium on Fault-Tolerant Computing*, pages 260–267, June 1999.

[10] I. Hamzaoglu and J. H. Patel. Reducing test application time for built-in-self-test test pattern generators. In *IEEE VLSI Test Symposium*, pages 369–375, May 2000.

[11] A. Jas, J. Ghosh-Dastidar, and N. A. Touba. Scan vector compression/decompression using statistical coding. In *VLSI Test Symposium*, pages 25–29, April 1999.

[12] A. Jas and N. A. Touba. Test vector decompression via cyclical scan chains and its application to testing core-based designs. In *IEEE International Test Conference*, pages 458–464, October 1998.

[13] A. Khoche, E. Volkerink, J. Rivoir, and S. Mitra. Test vector compression using EDA-ATE synergies. In *IEEE VLSI Test Symposium*, pages 97–102, May 2002.

[14] 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.

[15] H. K. Lee and D. S. Ha. HOPE: An efficient parallel fault simulator. In *IEEE Design Automation Conference*, pages 336–340, June 1992.

[16] I. Pomeranz and S. M. Reddy. Static test compaction for scan-based designs to reduce test application time. In *IEEE Asian Test Symposium*, pages 198–203, December 1998.

[17] J. Rajski, J. Tyszer *et al*. Embedded deterministic test for low cost manufacturing test. In *IEEE International Test Conference*, pages 301–310, October 2002.

[18] S. M. Reddy, K. Miyase, S. Kajihara, and I. Pomeranz. On test data volume reduction for multiple scan chain designs. In *IEEE VLSI Test Symposium*, pages 103–108, May 2002.

[19] L. Schafer, R. Dorsch, and H.-J. Wunderlich. RESPIN++ - deterministic embedded test. In *IEEE European Test Workshop*, pages 37–44, May 2002.

[20] N. A. Touba and E. J. McCluskey. Transformed pseudo-random patterns for BIST. In *IEEE VLSI Test Symposium*, pages 410–416, May 1995.

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY