

Compression of Partially Specified Test Vectors in an ATE Environment

F. Karimi¹, Y.B. Kim², F. Lombardi² and N. Park³

¹ LTX Corporation, 3930 North First Street, San Jose CA 95134, USA

² Dept of ECE, Northeastern University, Boston, MA 02115, USA, {ybk, lombardi}@ece.neu.edu

³ Dept of CS, Oklahoma State University, Stillwater, OK 74078-1053, USA, npark@a.cs.okstate.edu

Abstract – *The manufacturing test of today's digital chips requires new design considerations for automatic test equipment (ATE). Compression has been used in ATE to reduce storage and application time for high volume data by exploiting the repetitive nature of test vectors. This paper deals with compression of partially specified test sets, i.e. vectors with don't care (unspecified, X) entries. A proper X assignment can be used to enhance the potential for compression by introducing similarities among vectors (correlation). This paper provides a comprehensive treatment of the X assignment for compression within fixed order and reordered test sets. For fixed order the X assignment is based on a novel technique referred to as bit-wise correlation. Using a matrix representation of the test set, X entries located in the same position in the vectors are analyzed and Boolean values assigned based on three criteria (column-wise bit-correlation). Once the fully specified test set is generated, an appropriate compression technique is used. If reordering is allowed, the process of X assignment is substantially different. Reordering is accomplished through a variation of column-wise bit-assignment, namely pair-wise bit-correlation. Pair-wise bit-correlation restricts the analysis to vector pairs as an heuristic condition for generating the new ordered test set. Simulation results are provided to support the viability of the proposed approach to compress test set with partially specified vectors.*

Keywords – *compression, partially specified tests, ATE, correlation.*

I. INTRODUCTION

Test application time is one of the main sources of complexity in testing IP cores as commonly found in today's digital systems such as Systems-on-Chip (SoC). Vectors (in either random or deterministic forms) are either generated by on-chip hardware for on-line testing, or provided by an external tester such as an ATE for manufacturing test. For testing logic blocks, the feasibility of on-chip test generation is mainly restricted to (pseudo) random methods. Area overhead for dedicated on-chip deterministic vector generation in general is cost prohibitive for manufacturing test. Random vectors are usually generated using Linear Feedback Shift Registers (LFSR), so no storage is required. Response evaluation is performed using a signature analyzer that compacts test responses into a signature and compares it with the signature of an error-free reference design. LFSRs introduce a small area overhead; however, test by random vectors requires a long application time due to modest quality. Hybrid schemes are commonly used to reduce test time by reseeding the LFSRs.

Compression has been investigated for resolving some of the

problems in digital testing. Lossless compression is the process of encoding test vectors so that the original data can be uniquely reconstructed by a decoder. A fundamental issue of lossless compression is to decompose a data input set into a sequence of events, then to encode the events using as fewer bits as possible. While compaction reduces the number of vectors at generation time, data compression has a different objective, namely to encode test data (corresponding to the compacted vectors) with the minimum number of bits.

In general for manufacturing test, deterministic solutions for SoC face many challenges such as a high volume of test storage, long test application time through the serial paths and vector sets that are generated by third parties (i.e. from the IC core providers) with limited information. This is made more complicated by the high density integration encountered in the design of the head in an ATE. A compression process for an ATE environment is substantially different from the traditional testing scenario.

- Compression is required to reduce the amount of data traffic between the ATE station and the head.
- Storage in the head is limited because memory utilization must be simultaneously allocated to many test channels (pins).
- The process of data coding/decoding must be lossless so that fault coverage of the test process is not affected.
- Independently of the method used (Figure 1 shows on-chip and off-chip decoding methods within an ATE architecture), hardware due to the decompression unit (decoder) and synchronization with the operation of the head must be simple and DUT independent.

This paper deals with compression of partially specified test sets, i.e. vectors with don't care (unspecified, X) entries. This technique provides a comprehensive treatment of the X assignment for compression within fixed order and reordered test sets. A further technique based on column matching is also analyzed. Simulation results are provided to support the viability of the proposed approach to compress test set with partially specified vectors.

This paper is organized as follows. In the next section, a brief

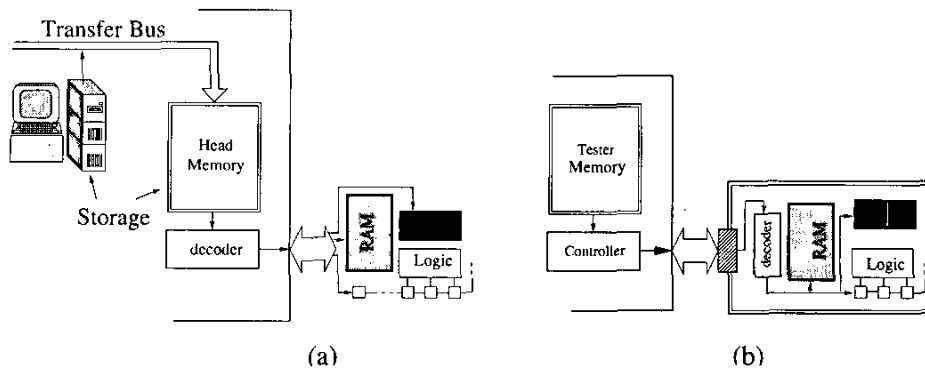


Fig. 1. ATE configuration . (a) off-chip decoding, (b) on-chip decoding

review of compaction and compression methods is presented. The basic principles of the proposed approach are outlined in Section 3. Section 4 presents compression with fixed ordering; Section 5 deals with compression with reordering. Section 6 presents the simulation results using ISCAS benchmarks. Conclusions are given in Section 7.

II. REVIEW

Reduction in test time can be accomplished using different methods such as *test compaction*, *pseudo-deterministic* vector generation and *data compression*. The advantage of using test compaction is two-fold: it reduces storage requirements and decreases the test application time (in this case a smaller number of vectors is applied to a DUT). Static and dynamic compaction techniques have been used to reduce the number of test vectors with the objective of no loss in fault coverage. Static compaction is employed after test generation. Dynamic techniques perform compaction [1] concurrently with the test generation process by often modifying the test generator.

Pseudo-deterministic methods have been considered as intermediate solutions to benefit from the lower hardware complexity of random generators and shorter test application time compared with deterministic methods. [3] uses an accumulator based structure to generate a set of deterministic tests. A run of the accumulator based generator is characterized by the initial value of the accumulator output (r_0) and a seed vector (c). The objective is to find a set of (r_0, c) pairs so that the concatenation of executions initiated by (r_0, c) pairs, cover the test set with a minimum number of vectors.

One of the major challenges facing ATE designers is data management due to high volume. Application of various data compression techniques for test data has been investigated in [5], [22], [23]. Run-length coding, Huffman coding, Lempel-Ziv and arithmetic coding are some of the compression techniques found in the literature. In run-length coding, a sequence of symbols is encoded into two elements, the repeating symbol

and the length of sequence. Run-length coding is efficient for data with long sequences of equal symbols.

Numerous studies have shown that test sets have unique features. Consecutive vectors have a common repetitive part (referred to as a *segment*) which may extend over several non-contiguous bits. In [21] it has been shown that test compression is equivalent to the process of finding a set of *non-overlapping blocks* covering possibly the entire test set. Repeating segments (specified once and reused for all vectors in a block) are used to reduce the size of the data set.

III. BASIC PRINCIPLES

In this paper the don't care assignment is accomplished using the following features:

1. To increase the correlation between bits located in the same position in the test vectors (i.e. among columns in the matrix representation of the test set); this process (referred to as column-wise bit-correlation) however is pursued using heuristic conditions due to the NP completeness of compression. In particular, the minimal number of value transitions is accomplished along each column to properly direct subsequent X assignments.
2. To increase the number of identical columns such that correlation is increased between vectors (identical columns can be removed). This results in a test set with a reduced number of X bits.
3. To use a combination of the above (hybrid).

IV. FIXED ORDER COMPRESSION

Consider initially the case in which the order of the vectors in the test set can not be changed. Let v_1, v_2, \dots, v_n define the fixed order vector set S . S is defined as a two-dimensional matrix; v_i corresponds to the i th row of S . The proposed approach to X assignment utilizes the Hamming distance between vectors as basic criterion. The Hamming distance between two vectors v_i and v_j ($i, j = 1, 2, \dots, n$) in V is defined as the sum

of the Hamming distances between the same bits in v_i and v_j , i.e.

$$H(v_i, v_j) = \sum_{k=1}^n v_i[k] \oplus v_j[k] \quad (1)$$

Therefore, the Hamming distance of a fixed order S is given by

$$H(S) = \sum_{i=1}^{n-1} H(v_i, v_{i+1}) \quad (2)$$

The Hamming distance establishes a relationship (referred to as bit-correlation) between vectors so that they can be analyzed for possible compression. An optimal X assignment is defined as an assignment that results in a minimal $H(S)$ (denoted by $H(S)_{min}$). By inspection of Equation 1 $H(S)_{min}$ can be obtained by finding the optimal assignment for the X entries of S .

Let S_i represents the bit array corresponding to the i th column of S . An optimal X assignment for S_i corresponds to an assignment that results in a minimal number of transitions (a 0 followed by a 1, or viceversa) in S_i , i.e. redundancy is introduced along each column. For example an optimal assignment for $S = XXX1XX0$ is $S_A = 1111110$, i.e. S_A only has only one value transition (1 to 0) and no other assignment can be found that results in less than one transition. An optimal assignment is not unique, for instance $S_B = 1111000$ is another assignment that only has one transition. In the general case, optimality in the X assignment for a column S_i is accomplished provided the following specific conditions apply:

1. all X 's between two 1's are assigned a value of 1, (e.g. $\dots 1XXX1\dots \rightarrow \dots 1111\dots$) (Criterion 1).
2. all X 's between two 0's are assigned a value of 0, (e.g. $\dots 0XXX0\dots \rightarrow \dots 0000\dots$) (Criterion 2).

A substantially different scenario is applicable when a X is located between 1 and 0 strings in the same bit positions (i.e. among a column); in this case, each X can take either of the boolean values. It is this scenario that yields an exponential process for complexity of the compression problem (and NP completeness for establishing optimality), because in the worst case all possible combinations of values must be computed for the X 's, i.e. if there are k X 's in the test set, then 2^k combinations must be considered to find the optimal assignment.

An assignment process for optimal compression has therefore an exponential complexity that severely limits its applicability for practical circuits (the number of don't care bits in the test vectors for ISCAS benchmark circuits is close to 90%). In the proposed approach the following heuristic condition is

used (Criterion 3): all the X 's between 1 and 0 strings in a column are assigned values of all 0 or all 1 (e.g. $\dots 1XXX0\dots \rightarrow \dots 1111\dots$ or $\dots 1000\dots$). The X assignment can also be performed so that the value transition occurs between the specified values at the beginning or at the end of the string (e.g. $\dots 1XXX0\dots \rightarrow \dots 1110\dots$ or $\dots 1100\dots$). Using the above assignment for the third criterion, the proposed technique achieves the least number of value transitions in each column of S . This condition effectively increases the redundancy in S as transitions are minimized along each column.

Application of the above three criteria in the X assignment process to individual columns of S , results in a completely specified vector set with a Hamming distance denoted by $H(S)$.

V. COMPRESSION WITH REORDERING

If the order of the test vectors in S is permitted to change, then compression consists of a different process. Reordering allows vectors to be arranged such that the Hamming distance of the test set can be changed appropriately, thus enhancing its compression rate. However, as the ultimate goal is a high compression rate, new features must be used for reordering the test set in the process of bit-correlation.

The method proposed in this paper is based on the following steps:

- Correlation is first established between pairs of vectors as an heuristic condition which is amenable to a graph approach. This process considers correlation between bits located in the same positions in pairs of vectors in the test set (pair-wise bit-correlation). This step is referred as the initial X assignment. Different measures (such as minimum Hamming Distance, maximum Hamming Distance and estimated Hamming Distance) must be used in this step.
- The shortest path algorithm is used for reordering of the test vectors based on pair-wise bit-correlation. This step minimizes the sum of the pair-wise bit-correlations in the vectors of S . The sum of the weights of the edges in the shortest path corresponds to the provisional Hamming distance of the reordered test set.
- The final X assignment is then established by column-wise bit-correlation according to the ordering found in the previous step. Note that in this step the final Hamming distance of the reordered test set could be changed compared with the provisional Hamming distance found through the shortest path algorithm. The fully specified test set S is then compressed using an established compression method.

The following notation is used throughout this section.

- For the vector set $S = \{v_1, v_2, \dots, v_n\}$, $S_A = [v_{A[1]}, v_{A[2]}, \dots, v_{A[n]}]$ defines an ordering A of S , where

$v_{A[i]}$ represents the i th vector in the ordered set S_A , and $1 \leq i, A[i] \leq n$. For example if $A = [1, 4, 2, 3]$ as ordering, then $S_A = \{v_1, v_4, v_2, v_3\}$.

- S^c represent S after an X assignment. If k denotes the number of don't care entries in S , then the X assignment for S can be performed in 2^k different ways. A possible X assignment for S (out of $2^5 = 32$ possibilities) is given by

$$\begin{array}{l} v_1 : 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \\ v_2 : 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \\ v_3 : 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \\ v_4 : 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \end{array}$$

- $d_{MIN}(v_i, v_j)$ represents the minimum distance between two vectors (v_i and v_j) in S and is defined as

$$d_{MIN}(v_i, v_j) = \sum_{k=1}^n v_i[k] \oplus v_j[k] \quad (3)$$

where $X \oplus 0 = X \oplus 1 = X \oplus X = 0$.

- $d_{MAX}(v_i, v_j)$ represents the maximum distance between v_i and v_j and is defined as

$$d_{MAX}(v_i, v_j) = \sum_{k=1}^n v_i[k] \oplus v_j[k] \quad (4)$$

where $X \oplus 0 = X \oplus 1 = X \oplus X = 1$.

Throughout our analysis the following vector set S is used in the examples.

$$\begin{array}{l} v_1 : 1 \quad x \quad 0 \quad 0 \quad x \quad 1 \quad 1 \quad 0 \\ v_2 : 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \\ v_3 : 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \\ v_4 : 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad x \quad x \quad x \end{array}$$

For the above test set, for example $d_{MIN}(v_1, v_2) = (0 \oplus 1) + (x \oplus 0) + (0 \oplus 1) + (0 \oplus 0) + (x \oplus 1) + (1 \oplus 0) + (1 \oplus 0) + (0 \oplus 0) = 1 + 0 + 1 + 0 + 0 + 1 + 1 + 0 = 4$ and $d_{MAX}(v_1, v_2) = (0 \oplus 1) + (x \oplus 0) + (0 \oplus 1) + (0 \oplus 0) + (x \oplus 1) + (1 \oplus 0) + (1 \oplus 0) + (0 \oplus 0) = 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 = 6$.

A. Pair-wise Bit-Correlation

Consider initially pair-wise bit-correlation. This is different from column-wise correlation (as used for the fixed order case) because the Hamming distance is not strictly related to the number of transitions of the test set due to reordering. Three measures are proposed; these measures are used within a common graph approach to generate the new ordering for the vectors in S .

1. Minimum Hamming Distance (denoted by OMIN with ordering L).

2. Maximum Hamming Distance (denoted by OMAX with ordering U).
3. Estimated Hamming Distance (denoted by OE with ordering E).

Let $G(V, E)$ be the fully connected graph representation of S where each vertex $v_i \in V$ corresponds to a vector v_i in S and each v_i is connected to every v_j ($i=1, \dots, n$ and $j=1, \dots, n$, $i \neq j$).

The following two weighted graphs based on the definition of minimum and maximum Hamming distances, are generated from G :

- $G_{MIN}: G(V, E, W)$ where $w_{i,j} = d_{MIN}(i, j)$ and $w_{i,j}$ is the weight of an edge $e_{i,j}$ in E of G . Fig. 2 (a) shows G_{MIN} for the example S .
- $G_{MAX}: G(V, E, W)$ where $w_{i,j} = d_{MAX}(i, j)$. Fig. 2 (c) shows G_{MAX} for the example S .

Pair-wise (minimum and maximum) Hamming Distances have different implications on bit-correlation and compression.

1. Minimum distance implies that the two vectors are similar, thus data redundancy is already present to an extent given also by the number of specified bits. However, the X assignment has limited effectiveness because correlation can be established only through the already specified bits.
2. Maximum distance implies that provided a large number of X entries exists, a substantial flexibility is possible in the X assignment and compression.

A heuristic method based on an estimate of the distance between vectors prior to a specific ordering is also proposed. This is an additional heuristic condition that can be used in the process of pair-wise correlation and is based on the distance estimate for the transition count of the columns in S . The probability that after a specific ordering, a X is placed between a 0 and a 1 is higher if the corresponding column has a high transition count. Accordingly, the Hamming distance estimate (denoted by $d_e(v_i, v_j)$) is defined as

$$d_e(v_i, v_j) = \sum_{k=1}^{n-1} p_k \quad (5)$$

where $p_k = 1$ if $v_i[k]$ and $v_j[k]$ have different (complement) values; else, p_k is given by the probability of a transition in the k th column. Hence, it is possible to define from G a new weighted graph G_E (similar to G_{MIN} and G_{MAX}) but the weights are given by the distance estimates as in Equation 5.

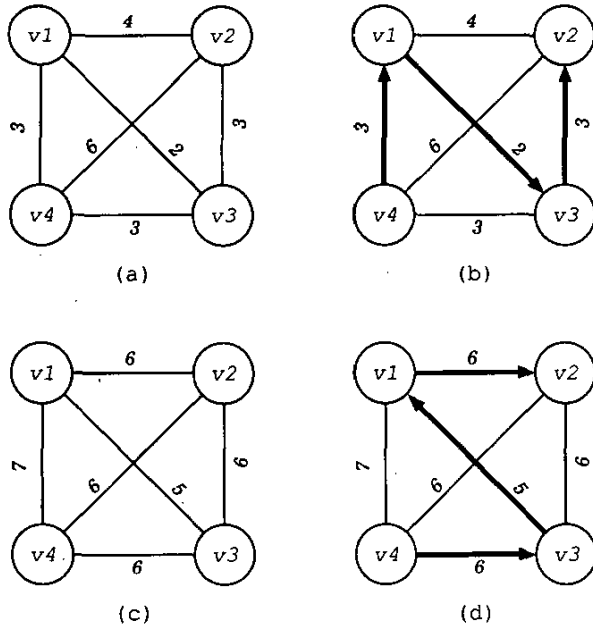


Fig. 2. (a) G_{MIN} , (b) shortest path in G_{MIN} , (c) G_{MAX} , (d) shortest path in G_{MAX}

B. Reordering

The second step of this process establishes the reordering of S . Reordering consists of modifying the order of the vectors of S , thus generating a new reordered test set (denoted by S^c). Consider the four vectors in the example test set S given previously; S can be reordered in $4!$ possible ways. For each ordering, the number of value combinations for the 5 don't care entries is 2^5 . Therefore, a total of $2^5 \times 4! = 768$ combinations are possible.

To avoid this exponential complexity (due to its NP completeness), a heuristic technique is proposed in this paper. This technique is based on the minimization of the sum of the Hamming distances (weights) of the edges in a path which connects all nodes in the path, i.e. the new ordering corresponds to the vertices in the shortest (weighted) path of G . This minimizes the sum of pair-wise Hamming distances between pairs of vertices (pair-wise correlation) not the final correlation (which is based on column-wise bit correlation), i.e. $H(S^c)$ is reduced.

Let S_L (S_H) represent the ordering corresponding to the shortest path in G_{MIN} (G_{MAX}). Fig. 2(b) illustrates the shortest path in G_{MIN} (in bold lines) which corresponds to $L = [v_4, v_1, v_3, v_2]$ as ordering. The bold path in Fig. 2(d) shows the shortest path in G_{MAX} which corresponds to $U = [v_4, v_3, v_1, v_2]$ as ordering.

In this analysis two extreme cases are therefore possible:

- $H(S_L)$: let A represent the ordering for S^c . As for any

arbitrary X assignment $d_{MIN}(v_i, v_j) \leq d(v_i^c, v_j^c)$, then

$$\sum_{i=1}^{n-1} d_{MIN}(v_{A[i]}, v_{A[i+1]}) \leq \sum_{i=1}^{n-1} d(v_{A[i]}^c, v_{A[i+1]}^c) \quad (6)$$

As S_L corresponds to the shortest path in G_{MIN} , so

$$\sum_{i=1}^{n-1} d_{MIN}(v_{L[i]}, v_{L[i+1]}) \leq \sum_{i=1}^{n-1} d_{MIN}(v_{A[i]}, v_{A[i+1]}) \quad (7)$$

From 6 and 7

$$H(S_L) = \sum_{i=1}^{n-1} d_{MIN}(v_{L[i]}, v_{L[i+1]}) \leq H(S_A^c) \quad (8)$$

- $H(S_H)$: let A represent an ordering for S^c . As for any arbitrary X assignment $d(v_i^c, v_j^c) \leq d_{MAX}(v_i, v_j)$, then:

$$\sum_{i=1}^{n-1} d(v_{A[i]}^c, v_{A[i+1]}^c) \leq \sum_{i=1}^{n-1} d(v_{H[i]}^c, v_{H[i+1]}^c) \quad (9)$$

because the ordering A represents the shortest path in the S^c set. Also as $d(v_i^c, v_j^c) \leq d_{MAX}(v_i, v_j)$,

$$\begin{aligned} & \sum_{i=1}^{n-1} d(v_{A[i]}^c, v_{A[i+1]}^c) \\ & \leq \sum_{i=1}^{n-1} d(v_{H[i]}^c, v_{H[i+1]}^c) \\ & \leq \sum_{i=1}^{n-1} d_{MAX}(v_{H[i]}, v_{H[i+1]}) = H(S_H) \end{aligned} \quad (10)$$

C. Final Assignment and Compression

Having established the new ordering for the test set, it is possible to generate the final X assignment. This is accomplished using column-wise bit-correlation as for fixed ordering on S^c . The fully specified test set is then compressed using an appropriate coding method.

VI. SIMULATION RESULTS

Let P_t be the average probability (in %) for a bit transition to occur in a column in the reordered test set after the provisional X assignment. Let P_X denote the percentage of don't care entries in the original test set (i.e. S) of a benchmark. Table I shows P_t obtained by the proposed methods for the ISCAS85 benchmarks (in most cases the X entries are the majority of bits in the test set). In most cases the application of SPRMAX results in a test set with the smallest number of transitions compared with the other two methods. SPMIN and SPMAX give the length of the shortest path (lower and upper bound respectively).

Table II shows the compression rates for the ISCAS85 benchmarks for fixed ordering and reordering (for the three proposed

	P_X	P_t				
		SPMIN	SPRMIN	SPRE	SPRMAX	SPMAX
c7552	90.52	0	2.05	1.17	0.79	91.75
c6288	50.37	0	5.79	4.51	3.97	56.22
c5315	94.96	0	1.59	0.93	0.76	90.56
c3540	75.45	0	5.46	3.08	3.93	80.13
c2670	94.62	0	1.05	2.11	1.87	92.78
c1908	47.48	1.58	8.52	8.35	7.74	90.15
c1355	19.74	4.02	8.06	7.29	6.69	27.36
c880	85.01	0	3.09	2.38	2.19	88.61
c499	2.49	6.91	7.76	7.63	7.21	10.86
c432	62.46	1.3	5.99	4.70	4.55	69.56

TABLE I
BIT TRANSITIONS AFTER REORDERING AND PROVISIONAL X
ASSIGNMENT

	M	Huffman($B=4$)				Huffman($B=8$)			
		Fixed	OMIN	OMAX	OE	Fixed	OMIN	OMAX	OE
c7552	516	65	62.3	66.74	65.78	74.13	71.22	77	75.12
c6288	69	38.56	41.1	59.70	43.3	39.77	54.08	68.61	61.76
c5315	457	72.29	65.23	66.38	71.17	85	74.41	76.66	82.17
c3540	266	42.32	43.34	47.63	51.07	46.2	49.01	53.31	60.76
c2670	323	65.3	71.11	69.94	65.56	75.74	86.77	81.04	75.02
c1908	121	43.50	42.29	55.38	44.98	47.68	45.91	62.35	48.11
c1355	123	53.02	55.05	54.38	49.90	56.64	58.12	56.52	52.65
c880	164	52.3	57.02	56.92	56.09	62.08	69.9	64.89	63.33
c499	66	48.36	46.23	49.19	46.80	53.07	50.2	56.44	51.01
c432	79	61.66	57.45	60.58	61.09	68.69	64.23	68.49	70.28

TABLE II
COMPRESSION RATES

ordering methods, OMIN, OMAX and OE). Note that M denotes the number of test vectors in S generated using HITECH. Compression has been accomplished using Huffman's method (for $B=4,8$). Bold entries identify the largest compression rate (for a given B) for each benchmark. As in a previous paper [21], Huffman's coding with $B=8$ provides the highest compression rate.

VII. CONCLUSION

This paper has presented a technique for compressing partially specified test vectors, i.e. X (don't care) entries are present in the vectors. This technique is based on a process by which bit positions in the vectors are analyzed and boolean values are assigned in place of the X 's (X assignment). This process is referred to as bit-correlation and effectively enhances the ability to compress the vectors by appropriately modifying the bits located in the same position.

References

- [1] E. M. Rudnick and J. H. Patel, "Simulation-based techniques for dynamic test sequence compation", *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 67-73, 1996.
- [2] S. K. Bommur, S. T. Chakradhar and K. B. Doreswamy, "Static test sequence compaction based on segment reordering and accelerated vector restoration", *Proc. IEEE International Test Conference*, pp. 954-961, 1998.
- [3] R. Dorsch, H. J. Wunderlich, "Accumulator based deterministic BIST", *Proc. IEEE International Test Conference*, pp. 412-421, 1998.
- [4] A. Jas, J. G. Dastidar and N. Touba, "Scan vector compression/decompression using statistical coding," *Proc. IEEE VLSI Test Symposium*, pp. 114-120, 1999.

- [5] T. Yamaguchi, M. Tilgner, M. Ishida, D. S. Ha, "An efficient method for compression test data", *Proc. IEEE International Test Conference*, pp. 79-88, 1997.
- [6] M. Ishida, D. S. Ha and T. Yamaguchi, "COMPACT: A hybrid method for compression test data", *Proc IEEE VLSI Test Symposium*, pp. 62-69, 1998.
- [7] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression", *IEEE Transactions on Information Theory*, vol. IT-23, no. 5, pp. 337-343, 1977.
- [8] J. Ziv and A. Lempel, "A compression of individual sequences via variable-rate coding", *IEEE Transactions on Information Theory*, vol. IT-24, no. 9, pp. 530-538, 1978.
- [9] T. A. Welch, "A technique for high-performance data compression", *IEEE Computer*, vol. IT-17, No. 6, pp. 8-19, 1984.
- [10] J. A. Storer and T. G. Szymanski, "Data compression via textual substitution", *Journal of ACM*, vol. 29, No. 4, pp. 928-951, 1982.
- [11] E.L. Lawler, A. H. Rinnooy-Kan, "The traveling salesman problem : a guided tour of combinatorial optimization," John Wiley & Sons, 1985.
- [12] A. Jas and Nur Touba, "Using an embedded processor for efficient deterministic testing of systems-on-a-chip", *Proceedings of the IEEE International Conference on Computer Design*, pp. 418-423, 1999.
- [13] J. A. Storer and M. Cohn Eds., "Fast and efficient lossless image compression", *Proc. IEEE Data Compression Conference*, pp. 351-360, 1993.
- [14] A. Chandra and K. Chakrabarty, "Test data compression for system-on-chip using Golomb codes", *Proc. IEEE VLSI Test Symposium*, pp. 113-120, 2000.
- [15] A. Jas and N. Touba, "Test vector decompression via cyclical scan chains and its application to testing core-based design," *Proc. IEEE International Test Conference*, pp. 458-464, 1998.
- [16] S. W. Golomb, "Run-Length encoding", *IEEE Trans. Inform. Theory*, Vol. IT-12, no. 4, pp. 399-401, 1966.
- [17] R. F. Rice, "Some practical universal noiseless coding techniques", Jet Propulsion Laboratory, Pasadena, California, JPL Publication 79-22, Mar. 1979.
- [18] R. Gupta and M. A. Breuer, "BALAST: A methodology for partial scan design," *Proc. 19th IEEE International Symposium on Fault Tolerant Computing*, pp. 118-125, 1989.
- [19] D. Kagaris, S. Tragoudas, and A. Majumdar, "On the use of counters for reproducing deterministic test sets, *IEEE Trans. on Computers*, Vol. 45, 1405, 1996.
- [20] A. Jagota, "An Adaptive, Multiple Restarts Neural Network Algorithm for Graph Coloring", *European Journal of Operational Research*, second half of 1996.
- [21] F. Karimi, Z. Navabi, W. Meleis and F. Lombardi, "Data Compression for System-On-Chip testing Using ATE," *Proc IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, 2002 (to appear).
- [22] K. Chakrabarty, B. Murray, B.T. Liu and M. Zhu, "Test width compression for built-in self-testing," *Proc IEEE International Test Conference*, pp. 328-337, 1997.
- [23] A. El-Maleh, S. Al-Zahir and E. Khan, "A geometric primitives based compression scheme for testing SOCs," *Proc. IEEE VLSI Test Symposium*, pp. 540-549, 2001.