Alternative Run-Length Coding through Scan Chain Reconfiguration for Joint Minimization of Test Data Volume and Power Consumption in Scan Test

Youhua Shi, Shinji Kimura[†], Nozomu Togawa[††], Masao Yanagisawa and Tatsuo Ohtsuki
Dept. of Computer Science, Waseda University, Japan
[†]Grad. School of Infomation, Production and Systems, Waseda University, Japan
[††]Dept. of Information and Media Sciences, The University of Kitakyushu, Japan
shi@yanagi.comm.waseda.ac.jp

Abstract

Test data volume and scan power are two major concerns in SoC test. In this paper we present an alternative run-length coding method through scan chain reconfiguration to reduce both test data volume and scan-in power consumption. The proposed method analyzes the compatibility of the internal scan cells for a given test set and then divides the scan cells into compatible classes. To extract the compatible scan cells we apply a heuristic algorithm by solving the graph coloring problem; and then a simple greedy algorithm is used to configure the scan chain for the minimization of scan power. Experimental results for the larger ISCAS'89 benchmarks show that the proposed approach leads to highly reduced test data volume with significant power savings during scan test.

1. INTRODUCTION

Nowadays, systems with multiple embedded cores, called System-on-a-chip (SoC), are becoming more prevalent in the industry. As a result, the continuous increase in chip complexity leads to a large amount of test data, which should be transferred between the low-speed external testers and the cores under test (CUT) through limited test channels. This poses a serious problem on test because of the cost and the limitations of automated test equipment (ATE). In addition to excessive test data, scan-based test suffers from increased test power; during test a much larger percentage of the scan cells change values in each clock cycle than that in normal operation, hence resulting in increased power dissipation. The consequent test power may exceed the certain power thresholds and damage the chip. Therefore test data volume and power consumption become two major concerns in SoC test.

To reduce test data volume, several techniques have been proposed from different angles in the literature, such as LFSR reseeding [1], selective Huffman coding [4], Golomb coding [5], and FDR coding [6]. There are also some new commercial tools such as TestKompress from Mentor Graphics [2] and SmartBIST from IBM/Cadence [3]. Besides, recently some methods have been proposed to reduce test data volume while considering the power-related problems such as test resource partitioning technique [9] and power-constrained static compaction [12].

In this paper, we propose an alternative run-length coding method for test data compression with minimum scan transitions, which analyzes the compatibility of the internal scan cells for a given test set and uses a dictionary to indicate the run-length of the scan-in data. To extract the compatible scan cells we apply a heuristic by solving the graph coloring problem; and then a simple greedy algorithm is used to configure the scan chain for the minimization of scan power. Unlike the other test data compression schemes, the proposed method explores the characteristics of the scan organization, thus it can achieves significantly higher compression ratio than previous methods with reduced peak and average scan-in power consumption.

The rest of this paper is organized as follows: Section 2 presents the alternative run-length coding method for the minimization of test data volume and scan-in power. The hardware implementation of the proposed test scheme is discussed in Section 3. Finally experimental results and conclusions are given in Section 4 and 5, respectively.

2. ALTERNATIVE RUN-LENGTH CODING

The approach to be described here is an alterative to run-length coding, which intends to reduce the volume of test data needed to be transferred from the ATE to the CUT with reduced scan-in power consumption. In this section we will present our coding strategy and illustrate it in detail.

Proceedings of the 13th Asian Test Symposium (ATS 2004)
0-7695-2235-1/04 $20.00 © 2004 IEEE

2.1. Alternative Run-Length Coding Through Scan Chain Reconfiguration

Due to the sparseness of specified bits in test cubes and the freedom to assign these don't care bits with arbitrary values, it is possible that some scan cells are compatible with others. Therefore we can divide the internal scan cells into compatible classes.

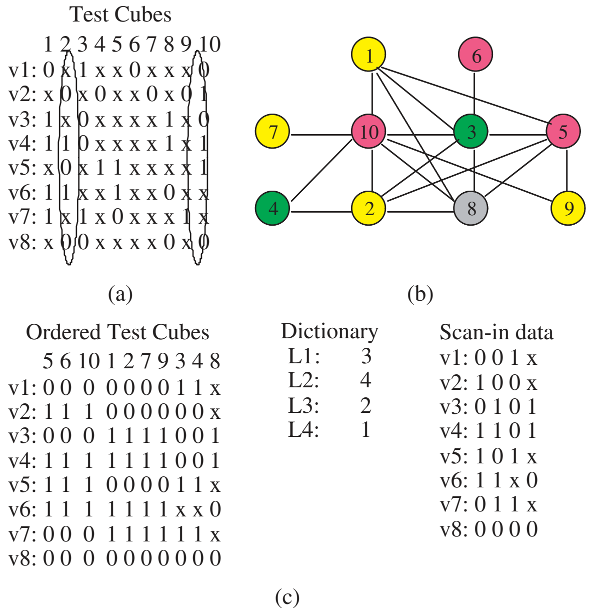
In deciding how to divide the scan cells into compatible classes, we create a conflict graph and solve a coloring problem for it. First we generate a partial specified test set. Here we assume that single scan chain is used. Then the test set can be expressed as a two dimensional matrix where each row is a test vector and each column is the value to be assigned to the corresponding scan cell. Two scan cells (i.e. two columns in this matrix), are compatible if for every row their corresponding logic values are same or at least one of them is a don't-care (X). It should be noted that the compatibility of these scan cells depends on the test set, i.e. if two scan cells are incompatible, it is not possible to conclude they are always incompatible, because it may be possible to be compatible in a different test set.

Next we use the test set matrix to create a conflict graph, which represents the relations of each pair of the incompatible scan cells. A vertex of the graph represents a scan cell. An edge between two vertices exists if and only if the two scan cells corresponding to the vertices are incompatible. The chromatic number of this conflict graph provides the number of compatibility sets, while the sets of vertices with the same color correspond to the sets of compatible scan cells. A dictionary is then built listing the number of compatible scan cells with the same color. At last the scan chain is reordered to packet the compatible scan cells together.

To better illustrate the problem outlined above, an example is illustrated in Fig 1. As a result, the heuristic colors the graph four colors as shown in Fig. 1 (b): s5, s6 and s10 with color 1, s1, s2, s7 and s9 with color 2, s3 and s4 with color 3, and s8 with color 4. The final scan chain configuration is shown in Fig. 1 (c) with the corresponding scan vectors. The example shows that the proposed approach reduces the size of scan-in data for each vector from 10 to 4 bits. When combined with the dictionary, overall 40 bits are needed to reproduce the original 80 bits test data. This is 50% saving in test data volume and the corresponding transfer time.

2.2. Optimization for Scan Power Minimization

In the above procedure, how to order the compatible scan cell classes to configure the scan chain does not affect the compression ratio, while it can be tailored to reduce the scan power consumption. In this subsection, first we make an analysis on scan power dissipation, and then model the scan chain configuration problem (SCCP) as the problem of finding an optimal order of compatible scan cell classes, such that the scan-in power is minimized.



Test Cubes

	1	2	3	4	5	6	7	8	9	10
v1:	0	x	1	x	x	0	x	x	x	0
v2:	x	0	x	0	x	x	0	x	0	1
v3:	1	x	0	x	x	x	x	1	x	0
v4:	1	1	0	x	x	x	x	1	x	1
v5:	x	0	x	1	1	x	x	x	x	1
v6:	1	1	x	x	1	x	x	0	x	x
v7:	1	x	1	x	0	x	x	x	1	x
v8:	x	0	0	x	x	x	x	0	x	0

(a)

(b)

Ordered Test Cubes

	5	6	10	1	2	7	9	3	4	8
v1:	0	0	0	0	0	0	0	1	1	x
v2:	1	1	1	0	0	0	0	0	0	x
v3:	0	0	0	1	1	1	1	0	0	1
v4:	1	1	1	1	1	1	1	0	0	1
v5:	1	1	1	0	0	0	0	1	1	x
v6:	1	1	1	1	1	1	1	x	x	0
v7:	0	0	0	1	1	1	1	1	1	x
v8:	0	0	0	0	0	0	0	0	0	0

Dictionary

L1:	3
L2:	4
L3:	2
L4:	1

Scan-in data

v1:	0	0	1	x
v2:	1	0	0	x
v3:	0	1	0	1
v4:	1	1	0	1
v5:	1	0	1	x
v6:	1	1	x	0
v7:	0	1	1	x
v8:	0	0	0	0

(c)

Figure 1. An example to explain the proposed alternative run-length coding procedure: (a) the example test set for the original scan chain; (b) the conflict graph; (c) the test set after reordering the scan cells with the dictionary and the corresponding scan-in data.

It has been confirmed in a real industrial ASIC that the transition power in the scan chains is the dominant contributor to test power [10]. In our work like that in [9][11], we use the weighted transitions metric (WTM) introduced in [12] to estimate the power consumption. The WTM metric models the fact that the scan-in power for a given vector depends not only on the number of transitions in it but also on their relative positions. We should mention that due to the uncontrollability of scan out data, in this paper we only address the scan-in power.

Consider a scan chain of length l and a scan vector $V_i = V^*_{i,1}V^*_{i,2}V^*_{i,3}\ldots V^*_{i,l}$ with $V^*_{i,1}$ scanned in before $V^*_{i,2}$, and so on. As it is shown in [12], the weighted transitions metric for V_i, denoted as WTM_i, is given by $WTM_i = \sum_{j=1}^{l-1} V^*_{i,j} \oplus V^*_{i,j+1}$. If the entire test set contains N_v vectors, then the average scan-in power P_{avg} and the peak scan-in power P_{peak} can be estimated as follows: $P_{avg} = \sum_{i=1}^{N_v}(\sum_{j=1}^{l-1}(l-j) * (V^*_{i,j} \oplus V^*_{i,j+1}))/N_v$ and

```
/* construct a conflict graph */
For each scan cell
  if scan cell ci and ck are incompatible
    add an edge (ci,ck) to the graph;

/* apply the graph coloring algorithm */
Apply the graph coloring algorithm to the conflict graph and
extract the compatible scan cell classes with the same color;

/* reconfigure the scan chain */
For each scan cells in the same coloring class
  fill the don't-cares;
Build a dictionary listing the numbers of scan cells with the
 same color;
Reorder the scan chain;
```

Figure 2. The proposed coding procedure

$$P_{peak} = \max_{i \in 1,2,\ldots,N_v} \{\sum_{j=1}^{l-1} (l-j) * (V^*_{i,j} \oplus V^*_{i,j+1})\}.$$

Since in the proposed method, the internal scan cells are divided into compatible classes, the scan cells in each class can be assigned to the same value. Assume that the scan chain is divided into m classes, thus every test vector can be expressed as $V_i = V_{i,1}V_{i,2}V_{i,3}\ldots V_{i,m}$, where the k^{th} class has the length of l_k and $\sum_{j=1}^{m} l_j = l$. Then in our method WTM_i can be computered as:

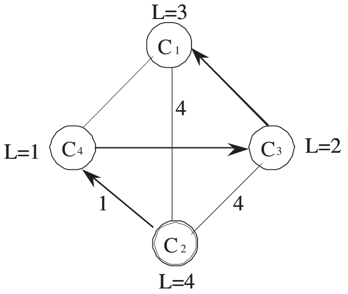
$$WTM_i = \sum_{j=1}^{m-1} (V_{l,j} \oplus V_{l,j+1}) * \sum_{k=j+1}^{m} l_k$$

From the above equation, we can find that in our coding method, the scan-in power for a given vector depends not only on the state transitions between each successive compatible classes but also on their relative positions. Thus in our work we use a cost graph to model the SCCP. We begin by forming a cost graph. Each node represents a scan cell class. An edge is placed between each pair of the nodes. The weight attached to the edge is the minimum transitions for the whole test set if the two corresponding scan cell classes are placed next to each other.

After the complete graph is given, the SCCP can be approximately represented as a well-known graph problem, traveling salesperson's problem (TSP). In TSP, each node must be visited once and only once and the total cost must be minimized. There are optimal algorithms for TSP, such as the use of integer linear programming. But it is impractical because of its high complexity in the large problem domain. Furthermore from above, we can observe that each compatible class may have different number of scan cells, so the length of each class should also be considered. Therefore we proposed a simple heuristic addressed the scan-in process to solve this problem.

```
Greedy Algorithm for SCCP
begin
  select the longest compatible class Cx;
  put C1 in the head of the list;
  K=K-1;
  while(K>0) {
    Cy: an unchosen class with the least (Cx, Cy) weight;
    append Cy to the list;
    Cx=Cy;
    K=K-1;
  }
end;
```

Figure 3. The proposed greedy for SCCP.



Final Test Cubes

	1	2	7	9	8	3	4	5	6	10
v1:	0	0	0	0	0	1	1	0	0	0
v2:	0	0	0	0	0	0	0	1	1	1
v3:	1	1	1	1	1	0	0	0	0	0
v4:	1	1	1	1	1	0	0	1	1	1
v5:	0	0	0	0	0	1	1	1	1	1
v6:	1	1	1	1	0	0	0	1	1	1
v7:	1	1	1	1	1	1	1	0	0	0
v8:	0	0	0	0	0	0	0	0	0	0

Figure 4. The cost graph for the example.

The heuristic selects and processes one node at a time until all nodes are processed. Unlike other heuristics, the node being selected first C_1 is the one with the maximal length (i.e. the corresponding compatible class has the greatest number of scan cells), which means that the scan cell belonged to the longest class are put at the end of the scan chain (i.e. the corresponding scan data is firstly scanned in). By doing so the maximal length will not be counted in the above Equation, which may result in reduced WTM. After determining the first node, we are going to select the next node in each iteration of the procedure until all nodes are ordered. The node to be selected is the one with the least weight with the previously selected one. If some nodes have the same weight, the longest is selected. So the procedure begins with a defined node and keeps tracking the edges with the least weights until all nodes are traversed. The greedy algorithm is described in Fig.3, and a step-by-step ordering process for the previous example is shown in Fig.4.

The heuristic has a complexity of $O(m^2)$, or more precisely, $\frac{m*(m-1)}{2}$, because for every jth node the next one is selected from the remaining (m-j) nodes. The greedy algorithm is a simple and yet effective method. In our experiments, it can achieve near optimal results.

After configuring the scan chain, we next analyze the upper bound of WTM_{max} for our method and compare it with previous works. The maximum value for the WTM_i is obtained for a test pattern that each group has alternating ones and zeros (i.e. every two successive groups have different values) and thus has the maximal switching activity. Therefore the upper bound of WTM_{max} (WTM_{um}) in our method is given by

$$WTM_{um} = \sum_{j=1}^{m-1} \sum_{k=j+1}^{m} l_k = \sum_{j=1}^{m} l_j * (j-1)$$

In the literature the peak power estimation for Golomb code [9] and FDR code [11] has been presented as $WTM_{pm} = \frac{l*(l-1)}{2}$, we then make a comparison to show the upper bound of WTM_{max} is reduced using our method, which is proved as follows:

$$\begin{aligned} & WTM_{um} - WTM_{pm} \\ &= \sum_{j=1}^{m} l_j * (j-1) - \frac{l*(l-1)}{2} \\ &\leq m * \sum_{j=1}^{m} l_j - l - \frac{l*(l-1)}{2} \\ &= m*l - l - \frac{l*(l-1)}{2} \\ &= \frac{l}{2} * (2m - l - 1) \leq 0 \end{aligned} \quad (1)$$

as long as $m \leq \frac{l+1}{2}$.

We then use the test set in Fig.1 (a) as an example to show the reduced transitions can be achieved using the proposed method. If the don't-cares are all mapped to zeros as that in FDR code [6], there will totally be 103 transitions in the flip-flops and the maximal WTM is 27 (WTM_7). If these don't-cares are mapped to binary values to minimize the WTM [7] and [9], then DXX…XD',($D \in \{0,1\}$) must be mapped to DDD…DD'. This ensures that the unavoidable transitions occur late during scan in. For this method, the total transitions will be 55 and $max(WTM) = WTM_1 = 13$. While using our method overall there are only 41 transitions in the scan chain when scan-in the entire test set and $max(WTM) = WTM_6 = 9$. From the simple example, it can be observed that the weighted transitions metric is clearly higher if the don't-cares are always mapped to zero. While using our method, both the maximal WTM and the total WTM can be reduced greatly when compared with the previous methods.

3. HARDWARE IMPLEMENTATION

In this section, we go to describe the decompression architecture for the proposed alternative run-length coding test data compression method. Since in our method the original test data is compressed into a run-length dictionary and the input test data, our decoding hardware consists of a small RAM for the dictionary and a decoder as a control unit. The decoder consists of a flip-flop, a $\lceil \log_2 K \rceil$-bit counter (counter1),a $\lceil \log_2 M \rceil$ -bit counter (counter2) and a selector, where M is the maximal length of the compatible classes and K is the number of compatible classes (i.e. the number of entries in the dictionary table). The decoder can read the data from the dictionary (i.e. the run-length of the scan data), and then send the scan data to the CUT. It controls the decoding process in the following way: In the beginning counter1 is reset to zero (i.e. the address starts from the beginning of the dictionary). The run-length at the address selected by counter1 is sent to counter2. While the scan-in data is read, counter2 is decremented and the scan-in data is sent to CUT with the enable signal (not shown in Fig.5), which is used to control the scan clock for the synchronization between the decoder and the CUT. When counter2 reaches zero, an ack. signal is sent to the ATE to require the next scan-in data. With the coming data, counter1 is incremented to determine the address of the next run-length. If the address goes to the end of the dictionary, counter1 will be set to zero, which indicates that the shift-in of a scan vector has finished.

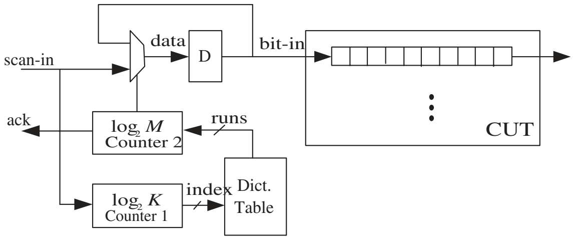


Figure 5. Decompression architecture.

The value of Counter1 is used as an address of the dictionary table and the contents at that address is loaded into counter2 to indicate the run-length of the scan-in data. The two counters are the core of the decoder, which not only control the decompression process, but also take charge of the synchronization among the ATE, the CUT and the decoder.

Our decompression logic is simple and flexible to be reused for different cores in a system. To fit in a SoC test environment, the RAM and the two counter of our structure should be designed according to the maximum number of compatible classes and the longest length of the compatible classes for the embedded cores so that it can reproduce the test data for all the embedded cores without introducing significant hardware overhead.

Table 1. Characteristics of benchmark circuits

circuit	s13207	s15850	s38417	s38584	s9234
FFs	638	534	1636	1426	211
Vectors	240	101	114	117	122
Xs	93.6%	78.1%	82.4%	91.9%	72.7%

Table 2. Results for the proposed method

	original test	Alternative Run-length Coding		
Circuit	data (bits)	scan-in	dictionary	CR
s13207	153120	30960	645	79.4%
s15850	53934	21513	852	58.5%
s38417	186504	37050	2975	78.6%
s38584	166842	18486	1422	88.1%
s9234	25742	10492	430	57.6%

4. EXPERIMENTAL RESULTS

To validate the efficiency of the proposed method, experiments were performed on the full-scan version of the largest ISCAS 89 benchmark circuits. For all the full-scan circuits, we consider a single scan chain. Table 1 shows the characteristics of the five benchmark circuits used in our experiment. For each circuit, the number of scan cells, the number of test vectors, and the don't care bit densities are listed respectively. In the table the number of scan cells is the number of sequential elements in the circuit. A commercial ATPG tool is used to generate the test vectors that provide 100% coverage of detectable faults for each circuit. The number of test vectors and the percentage of don't-cares shown in this table are obtained after the ATPG tool compaction.

Table 2 shows the compression results for the proposed alternative run-length coding method. The first two columns show the names of the CUT and the bits of the original test data. Then the volume of scan-in data to be transferred from the ATE to the CUT is reported with the required dictionary size. The last column lists the compression ratio (CR), which is computed as $CR = \frac{|T_D|-|T_E|}{|T_D|} * 100\%$. In the table, the dictionary size is computed as $\lceil \log_2 M \rceil * K$, where M is the maximal length of the compatible classes and K is the number of compatible classes. The table clearly shows that significant compression, up to 88.1%, can be achieved when using our alternative run-length coding method.

To give an idea of how the amount of test data compression for the proposed alternative run-length coding method compared with other test schemes also based on run-length coding, we then present a comparison of the compression results with the previously published techniques such as FDR coding [6], extended FDR coding [8] and alternating run-length coding [9] in Table 3. Since the test data set is essential to the compression ratio, here we applied each method of [6][8][9] to the same test set we used in Table 2, and each method was implemented in C++ so as to provide a uniform basis for the comparison. As seen from the table, compared to these techniques, the proposed approach leads to more significant compression results in all cases. Consider for example the largest circuit s38584, there is as much as 29.48% (when compared to FDR coding) increase in compression.

The second set of experimental results is on the peak and average scan-in power consumption. As illustrated in Section 2, here we use WTM as a metric to estimate the power consumption. Table 4 compares the average and the maximal WTM using different don't-cares mapping methods. Let P_{peak} and P_{avg} represent the peak and average power consumption for each method. The table clearly shows that the peak power and average power are significantly less if the proposed alternative run-length coding method is used for test data compression and decompressed vectors are applied during scan testing. On average, the peak (average) power is 88.9% (94.2%) less in this case than for the Mintest test sets. While when compared with the other two methods up to 79.7% (78.3%) reduction can be achieved. We also present the results of power consumption for different don't-cares mapping methods over s38417 as a sample representative in Fig.6. It also confirms the upper bound of WTM derived above. The curve for the proposed method is always below the curve for the other methods. Thus the experimental results demonstrate that substantial reduction in test data volume is also accompanied by significant power savings during scan testing. It should be mentioned that although scan-out test power is as important as scan-in power consumption, due to its uncontrollability, we didn't address it in this paper.

Finally it should be mentioned that layout constraints may restrict the scan chain reconfiguration as two scan cells to be placed in neighboring positions may corresponding to two flip-flops located far away from each other. We are currently investigating this aspect by incorporating the layout constraints into the SCCP to achieve power reductions with no layout violations.

5. CONCLUSIONS

In this paper we have proposed an alternative run-length coding method based on scan chain reconfiguration, which overcomes the limitations of the previous run-length codes. Based on the experimental results over ISCAS'89 benchmark circuits, it has been shown that the proposed test data compression method outperformed the previously published approaches, and up to 88.1% compression ratio and

Table 3. Comparison of compression results

Circuit	No. of bits in oiginal test set	No. of bits in Mintest test set	Proposed		Compression ratio (%)		
			compressed bits	CR (%)	FDR coding [6]	EFDR coding [8]	ARL coding [9]
s13207	153120	148654	31605	79.36	71.3	72.6	74.19
s15850	53934	50196	22365	58.53	50.7	51.93	53.83
s38417	186504	111248	40025	78.54	51.17	52.28	56.72
s38584	166842	156860	19908	88.1	58.62	59.25	66.18
s9234	25742	22155	10922	57.6	42.21	43.1	47.1
average	–	–	–	69.42	53.02	54.06	57.23

Table 4. Comparison of power consumption

CUT	Mintest set		Don't-cares mapping method					
			All zeros		Min WTM		Proposed	
	WTM_{max}	WTM_{avg}	WTM_{max}	WTM_{avg}	WTM_{max}	WTM_{avg}	WTM_{max}	WTM_{avg}
s13207	135607	122031	58891	16282	34900	8315	15442	5059
s15850	100228	90899	61842	22635	44134	14572	16244	7357
s38417	683765	601840	249544	138474	148602	86616	50538	39829
s38584	572618	535875	185943	71552	104278	39569	37774	17016
s9234	17494	14630	10935	4658	7273	2265	2469	1011

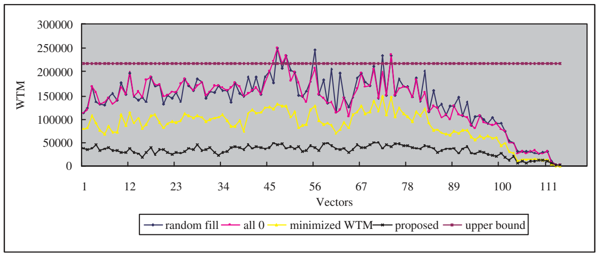


Figure 6. Power consumption with different don't-cares mapping methods over s38417.

93.4% (96.8%) peak (average) scan-in power savings are achievable. In addition the decompression architecture is reusable and easier to synchronize with the ATE and with the CUT.

References

[1] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, "Built-in Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers," *IEEE Trans. on Comp.*, Vol. 44, No. 2, pp. 223-233, Feb. 1995.

[2] J. Rajski, et. al, "Embedded Deterministic Test for Low Cost Manufacturing Test," *in Proc. Int. Test Conf.*, pp. 301-310, 2002.

[3] B. Koenmann, "SmartBIST," *Presentation by IBM in ITC2000*, 2000.

[4] A. Jas, J. Ghosh-Dastidar, M.-E. Ng, and N.A. Touba, "An Efficient Test Vector Compression Scheme Using Selective Huffman Coding," *IEEE Trans. on CAD*, Vol. 22, No. 6, pp.797-806, June 2003.

[5] A. Chandra, K. Chakrabarty, "System-on-a-chip test data compression and decompression architectures based on Golomb codes," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 20, No. 3, pp. 355-368, March 2001.

[6] A. Chandra, and K. Chakrabarty, "Frequency-Directed Run-Length (FDR) Codes with Application to System-on-a-Chip Test Data Compression," *in Proc. VLSI Test Symp.*, pp. 42-43, 2001.

[7] S. Hellebrand, and A. Wurtenberger, "Alternating Run-Length Coding - A Technique for Improved Test Data Compression," *in Proc. Int. Workshop on Test Resource Partitioning*, 2002.

[8] A. El-Maleh, and R. Al-Abaji, "Extended Frequency-Directed Run-Length Codes with Improved Application to System-on-a-Chip Test Data Compression," *in Proc. Int. Conf. Electronics, Circuits and Systems*, pp. 449-452, 2002.

[9] A. Chandra, and K. Chakrabarty, "Reduction of SoC Test Data Volume, Scan Power and Testing Time Using Alternating Run-length Codes," *in Proc. Int. Design Autom. Conf.*, pp. 673-678, 2002.

[10] P.Rosinger, P.Gonciari, B.M.Al-Hashimi, and N.Nicolici, "Analysing trade-offs in scan power and test data compression for Systems-on-a-chip," *IEE Proceedings on Computers and Digital Techniques*, Vol. 149, pp.188-196, 2002.

[11] A. Chandra and K. Chakrabarty, "Low-power scan testing and test data compression for system-on-a-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 21, pp. 597-604, 2002.

[12] R.Sankaralingam, R.R. Oruganti, and N.A. Touba, "static compaction techniques to control scan vector power dissipation," *in Proc. VLSI Test Symp.*, pp.35-40, 2000.