

# An Effective Hybrid Test Data Compression Method Using Scan Chain Compaction and Dictionary-based Scheme

Taejin Kim, Sunghoon Chun, Yongjoon Kim, Myung-Hoon Yang, and Sungho Kang  
*Dept. of Electrical and Electronic Engineering, Yonsei University*  
134 Shinchon-dong, Seodaemun-gu, Seoul, Korea  
Tel: +82-2-2123-2775, Fax: +82-2-2123-7726  
{tj2010, shchun, yjkim, mhyang}@soc.yonsie.ac.kr, shkang@yonsei.ac.kr

**Abstract**— In this paper, we propose a new test data compression method for reducing test data volume and test application time. The proposed method consists of two steps: scan chain compaction and dictionary-based compression scheme. The scan chain compaction provides a minimum scan chain depth by using compaction of the compatible scan cells in the scan chain. The compacted scan chain is partitioned to the multiple internal scan chains for using the fixed-length index dictionary-based compression scheme that provides the high compression ratio and the fast testing time. The proposed compression method delivers compressed patterns from the ATE to the chip and drives a large number of multiple internal scan chains using only a single ATE input and output. Experimental results for the ISCAS-89 test benches show that the test data volume and testing time for the proposed method are less than previous compression schemes.

**Keywords**— test data compression, test application time, full-scan circuit, scan chain compaction, dictionary-based compression scheme

## I. INTRODUCTION

As the complexity of very large scale integration (VLSI) circuit increases, testing plays an important role in today's system design [1]. One of the most important factors in driving up the test cost is increasing the amount of test data volume, which is a result of the large size of the designs and the new types of defects appearing in the advanced manufacturing process. A large amount of test data must be stored in the automatic test equipment (ATE) and transferred deep into the chip as fast as possible. Since the channel capacity and the size of memory of ATE are limited, the test application time and the test power have been significantly increased. The test data compression technique offers a promising solution to the problem of increased the test data volume for SoC testing.

Several test data compression schemes have been proposed for reducing test data volumes. These previous schemes of compression techniques can be categorized into two groups: code-based schemes and structural schemes. In the code-based schemes, a number of techniques based on FDR coding [2], VIHC coding [3], RF-Huffman coding [4], and Selective-Huffman coding [5, 6], have been proposed to reduce test data volume. Although these compression methods can achieve high compression ratio, the test application time is not sufficiently reduced, because of the

long shifting time to drive into the full-scan chain. Additional synchronization and handshaking between the SoC and the ATE can be occurred in decompression decoder by using multiple clocks to increase testing speed. Several dictionary-based compression methods have been presented to reduce SOC test data volume in [7 – 9]. To get high compression results, these methods need the memory or large size hardware circuit to store dictionary entries.

In the structural compression method, the Illinois scan architecture [10] use an external input port to drive multiple internal scan chain inputs without any logic gates; it provides the reduction of test data volume and testing time. However, additional fault simulation and test generation are necessary in [10] to achieve high fault coverage. Additionally, if the faults are hard to be detected by using the parallel mode of the Illinois scan architecture, the test vector needs to be driven serially into a single long scan chain.

The scan chain reconfiguration method has been proposed in the form of an XOR-based [11]. This method identifies the compatibility of scan cells in the scan chain, and then the scan chain network is constructed by XOR gates. The scan chain network has a shorter scan chain depth than a traditional scan chain. It can provide the reduction of test data volume and test application time simultaneously. However, if the test data cube does not contain enough compatible scan cells, the effect of reducing scan chain depth is restricted.

In this paper, we propose a new effective test data compression method using the scan chain compaction and the dictionary-based compression scheme. The first method, the scan chain compaction constructs a scan chain network which has minimum number of scan slices; the number of scan slices is scan depth, similarly in [11]. The small size of scan chain depth can reduce total data volume and test application time; because it needs less scan chain shifting operation to drive test data bits into the scan chain. However, the reduction of scan chain depth by the compaction is limited, if the test vector does not have enough compatible scan cells. To achieve high compression ratio, we combine the dictionary-based compression scheme with the scan chain compaction. The scan chain network composed by the scan chain compaction is divided to the multiple internal scan chains to reduce the scan shifting time using the dictionary-based compression scheme. The decoder of

proposed dictionary-based scheme drives the test vector into the whole multiple internal scan chain inputs at the same time, as it can help to reduce the test data volume and test application time. The hardware overhead can be kept reasonable size because we use the dictionary which has fixed-length indices and the small amount of the index entries which are already compressed by the scan chain compaction. The proposed compression scheme uses the test data vector generated by the traditional ATPG and delivers high compression ratio and fast application test while retaining the original faults coverage without any additional fault simulation and test generation. The proposed compression architecture is implemented with only one single scan-in and scan-out pin externally, although the multiple scan chains are consisted internally. Experimental results for large ISCAS-89 benchmark circuits show that proposed compression method could reduce test data volume and test application time significantly.

## II. OVERVIEW OF PROPOSED SCHEME

### A. Definitions

To present the proposed test data compression algorithm simply, several definitions are used.

**Definition 1. Scan Frame:** A scan frame is a vector of inputs applied to the same scan cells in each test pattern, which contains the number of 0s, 1s and don't-cares (Xs).

**Definition 2. Direct compatible:** If scan frames of two scan cells in the scan chain never receive conflicting binary values in totality for the same test cube, these two scan chain are direct compatible, and can be driven the same test data .

**Definition 3. Inverse compatible:** If scan frames of two scan cells in the scan chain never receive identical binary values in totality for the same test cube, these two scan chain are inverse compatible, and can be driven the same test data by inserting inverter on the scan path.

### B. Overall algorithm of the proposed scheme

The overall framework of our compression scheme for scan based on circuit is shown in Fig. 1. The proposed scheme uses the test data cube of the full-scanned circuit, the test cube is generated by traditional ATPG without any additional fault simulation and test generation. Since the test vectors generated by the ATPG have large number of don't care bits, there are many compatible or inverse compatible scan cells in the full-scanned circuit. All compatible or inverse compatible scan cells are identified and composed to groups of scan cells which have same test values for all test vectors. To find the maximal number of scan cells in a group, the conflict graph for all scan cells is used. The scan chain can be constructed to the compacted scan chain network using the conflict graph of scan cells. The compacted scan chain network has a shorter scan chain depth than a conventional scan chain. The short scan chain depth can provide the reduction of the test data volume and test application time simultaneously. Although the scan chain is compacted to the scan chain network with

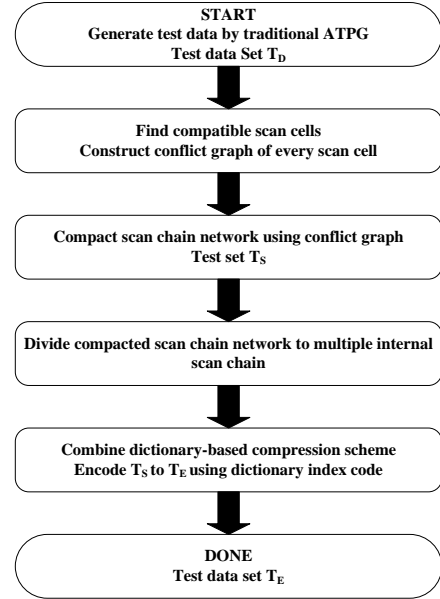


Fig. 1 Overall framework of the proposed scheme

short scan depth using the conflict graph, the reduction of scan chain depth is limited when the test vector does not have enough compatible scan cells. To alleviate this limit and achieve high compression ratio, the new dictionary-based compression scheme is combined with the scan chain compaction. To use the dictionary-based compression scheme, the compacted scan chain network is divided to the multiple internal scan chains. Each slice vector of the multiple internal scan chains is driven into the all scan chain inputs at the same time. The slice vector has a fixed length and which is a symbol of the dictionary. The symbol is encoded to the fixed length dictionary index that is shorter than the length of the symbol. Therefore the proposed dictionary-based compression is the vertical compression that presents not only the reduction of the test data volume and application time but also the use of only a single ATE channel. The original test vector  $T_D$  can be compacted and encoded to the significantly compressed test vector  $T_E$  by combining the scan chain compaction and the dictionary-based compression scheme.

## III. SCAN CHAIN COMPACTION

The proposed scan chain compaction methodology aims at constructing a scan network with minimum scan depth. Such a solution perfectly exploits the structural relationship between scan cells to deliver the highest compression ratio while retaining the original faults coverage. There are many scan cells which are direct or inverse compatible with other scan cells in the full scan chain. These scan cells compose the compatible scan cell group, all scan cells in the same group are always driven identical bits (direct compatible) or never identical bits (inverse compatible). All scan cells in the same group can be injected the same test data value at the same clock cycle. Therefore the compacted scan chain network can be constructed and each compatible scan cell group becomes

	Scan in	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>	F <sub>7</sub>	F <sub>8</sub>	F <sub>9</sub>	F <sub>10</sub>	F <sub>11</sub>	F <sub>12</sub>	F <sub>13</sub>	Scan out
T1		1	X	0	0	0	0	0	0	0	X	X	X	0	
T2		1	0	0	1	0	X	1	1	0	0	0	0	1	
T3		1	1	1	1	1	X	0	0	1	1	X	X	0	
T4		1	X	X	1	1	1	X	1	1	X	X	1	1	
T5		0	X	1	0	1	0	1	1	X	X	0	X	X	

Fig. 2 Sample of single scan chain and test vector

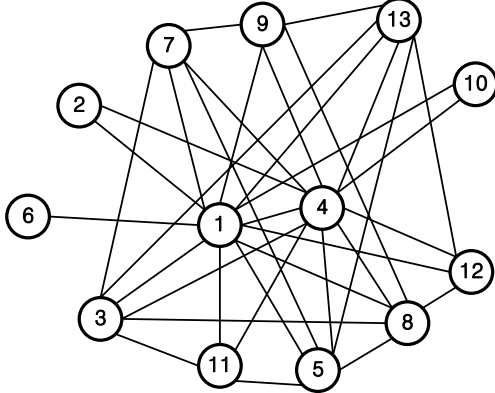


Fig. 3 Conflict graphs for scan cells in Fig 2.

one depth of the scan chain network. The scan network expands and compacts depending on the compatible scan cell group sizes resulting of varying numbers of scan cells. To construct the scan chain network, the types of expansion and compaction mechanisms within the scan network are defined.

The expansion mechanism in the form of a fan-out network allows in scan cells within the compatible scan cell group to always receive identical bits, although the group sizes may vary. The expansion expands the test data value from a single scan-in pin or small size groups to the large size groups with the fan-out network. The compaction mechanism uses XOR gates to compact the test data value from the larger size group to the next smaller size group. A series of such expansions and compactions construct the compacted scan chain network between a single scan-in and scan-out pins. The number of the compatible scan cell groups between scan-in and scan-out pins constitutes the scan depth, and as many shift cycles will be required for each test vector. We present two rules on group sizes to ensure the delivery of always identical bits to the scan cells within the same group, and avoid certain aliasing problem in the expansion and compaction mechanisms. First, XOR gates with only an odd number of inputs are used to compact a group into the subsequent group to avoid constant value of generation by the XOR gates. Secondly, the number of outputs of every expanding fan-out should be odd in order to avoid self-aliasing [11]. According to these two rules, each compatible scan cell group should consist of an odd number of scan cells.

To achieve the highest compression ratio, the number of depth in the scan chain network is minimized, that depends on the number of compatible scan cell groups. The conflict graph and greedy-graph-coloring heuristic algorithm are used to construct the most compact scan chain network. The conflict graph represents the relationship between scan cells for all

#### Graph\_Coloring\_Heuristic

```

{
  While Graph have vertices
    Select minimum conflict vertex and add to Group G;
    Repeatedly add to G minimum conflict vertex which are non-
    conflict to all the vertices in G;
  If |G| is even
    Remove the last added vertex from G;
  Color in all vertices in G with same color;
  Remove from Graph all the colored vertices and conflict line;
  Return;
}

```

Fig. 4 Greedy-graph-coloring heuristic algorithm

scan cells in the scanned circuit. The vertices of the graph represent each scan cell, an edge between two vertices exists if and only if two scan cells corresponding to the vertices are incompatible that means these two scan cells cannot belong to the same group. Fig. 3 shows the conflict graph for all scan cells of the single scan chain in Fig. 2. The conflict graph should be colored in minimum numbers of colors in order to identify compatible scan cell groups. The graph coloring problem [12] finding minimum kinds of colors is a well-known NP complete problem. As we use a greedy-graph-coloring heuristic, to find minimum color of the vertices in the conflict graph. These heuristic algorithms are provided in Fig. 4. By the heuristic algorithm, coloring of the conflict graph in Fig. 3 leads that five sets of independent vertices are {6, 2, 10, 11, 12}, {3, 5, 9}, {7, 8, 13}, {4} and {1}. To avoid expansion and compaction problems mentioned above, every color is used on an odd number of vertices to make that every group contains an odd number of scan cells.

After all compatible scan cell groups are achieved, the scan cell groups are ordered such that slices are almost same size, for the maximum expansion is possible, to reduce the area overhead of the XOR gates in the compaction. The area overhead of the proposed scan chain network comes from the number of compacting XOR gates between the scan cell groups. A group of  $n$  scan cells that is compacted into another slice of  $m$  cells, needs  $\lfloor (n-m)/2 \rfloor$  3-input XOR gates. The equivalent area overhead in terms of 2-input XOR gates can be denoted by  $n-m$ .

As mentioned in Section II, scan-cell compatibility can be the form of direct or inverse compatibility. Although the proposed method so far has concentrated on the direct compatible case, inverse compatibility is also equally beneficial. If two inverters are inserted into the scan path, the inverse compatible can still be treated as compatible.

The example of the scan chain compaction is shown below. Fig. 2 illustrates a traditional single scan chain in which flip-flops are connected in series with 13 scan cells. The number of scan cells in the scan chain is a scan depth. The amount of test data for the scan chain is computed as the scan depth times the number of test vectors. In case of Fig. 2 with five test vectors, the volume of test data cube is 65 (=13\*5) bits. Then we construct the conflict graph of these scan cells showed in Fig. 3 and use the greedy-graph-coloring heuristic algorithm to find the compatible scan cell groups, {F2, F6,

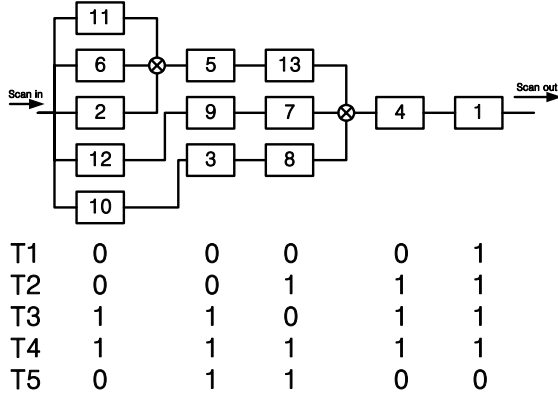


Fig. 5 Sample of the compacted scan chain network

F10, F11, F12}, {F3, F5, F9}, {F7, F8, F13}, {F4} and {F1}. To reduce the number of XOR gates for the compaction mechanism, the groups are ordered in descending. These ordered scan cell groups are stitched via expanding fan-outs and compacting XOR gates, the sample compacted scan chain network produced by the proposed scheme are given in Fig. 5. The compacted scan network in Fig. 5 reduces the depth of scan chain from 13 to 5, then the test data volume is reduced to 25 ( $=5 \times 5$ ) bits. Similarly, the test application time for each test vector is reduced from 13 clock cycles to 5 clock cycles. The area overhead of the compacted scan chain network is two 3-input XOR gates that are used for the compaction mechanism.

#### IV. DICTIONARY-BASED COMPRESSION SCHEME

The compacted scan chain network, which is mentioned in the previous section, reduces the scan depth of the scan chain, and reduces the test cost. However, if the test data cube does not contain enough compatible scan cells, the effect of the short scan depth is restricted. Then we combine the dictionary-based compression scheme with the compacted scan chain network to achieve the high compression ratio and the fast application time. To address the dictionary-based compression scheme, the compacted scan chain network which has a single scan-in and single scan-out, is partitioned to the multiple scan chains. We divide the scan network to proper number of scan chains, and balance the depth of each scan chain. If the scan chains are unbalanced, additional dummy scan cells are inserted to make that all the scan chains have the same scan depth. The multiple scan chain composed from the compacted scan chain network presents the reduction of the test application time, however the test data volume is unchanged and additional scan inputs and outputs are required.

The proposed dictionary-based compression scheme compresses the test data cube of the multiple scan chains in vertically. The decompress decoder of the scheme receives the compressed test data from the ATE, decompresses the compressed test data, and drives the test vector into the all multiple scan chain input at the same time, it can help to reduce the test data volume and test application time. Fig. 6 illustrates the block diagram of the proposed compression

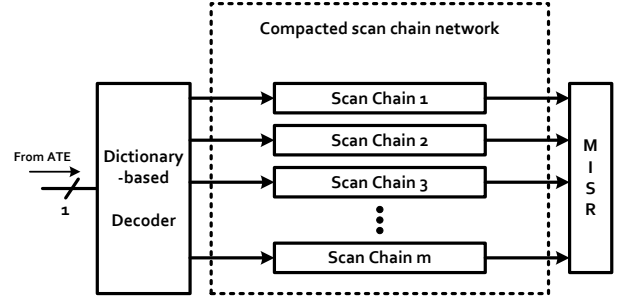


Fig. 6 Illustration of the proposed compression scheme

scheme dictionary-based test data compression scheme. In the scheme, an ATE scan-in is driven into the  $m$  multiple scan chains of the compacted scan network through the dictionary-based decoder. Each scan slice vector becomes an  $m$ -bits symbol, which is encoded to codeword. During the test application, the codeword is shifted into the decoder, after that  $m$ -bits symbol is immediately generated by the decoder and driven into the multiple scan chain inputs. The test responses shifted from each scan chain outputs can be compacted by using MISR or other techniques. The proposed decompression architecture assume a  $1$ -to- $m$  scan configuration, in which the number of internal scan chains is  $m$  times the number of external scan I/O pins, so significant reductions in the test data volume and test application time can be achieved.

The codeword which is encoded from the  $m$ -bits scan slice symbol is composed of a prefix and tail. The 1-bits prefix indicates whether the following tail is a dictionary index or an uncompressed test data. If the prefix is 1, the following tail is dictionary index. On the other hands, if the prefix is 0, the following tail is uncompressed test data. The lengths of the dictionary index and uncompressed test data are different. In the whole test vector from the scan chain compaction, there are various kinds of scan slices vectors, which are the symbols. Some symbols, which have more frequencies of occurrence than the other symbols, are selected to the entries

```

SV : the set of the scan slice test vector (length is m-bits)
N : the size of SV
FC[i] : the frequency of occurrence of the i-th SV
Merge_scan_slice_vector
{
  qsort_dc(SV); // sort the SV by descending order of FC
  for(i=1; i<N; i++) {
    for(j=0; j<i; j++) {
      if(SV[i] == NULL || SV[j] == NULL) continue;
      chk = is_merge(SV[i], SV[j]);
      // check that SV[i] can be merged with SV[j]
      if( chk == OK ) { // possible to merge
        SV[j] = SV[i];
        FC[j] = FC[j] + FC[i];
        SV[i] = NULL;
      } } }
  qsort_dc(SV); // sort the SV by descending order of FC
  count_SV(); // count the number of modified SV
  Fill_don't_care(); // don't cares are all mapped to 0s.
}

```

Fig. 7 Algorithm of finding maximize encoded codewords

of the dictionary and encoded the dictionary index codes. The number of dictionary index codes is decided by the size of the dictionary. On the contrary, symbols, which cannot belong to the dictionary, are used intact to the codeword, the uncompressed test data. For example, there are 6 kinds of 8-bits symbols in descending order of frequencies of occurrence in the test cube. The total bits of the test cube is 48 bits ( $=6 \times 8$ ). The size of the dictionary is 4 and the length of the dictionary index is 2. Three bits codeword are needed to encode the first 4 symbols in the list, 1 bit is needed for prefix and 2 bits are required for the dictionary index. The rest 2 symbols need 9 bits codeword, 1 bit for prefix and 8 bits for the uncompressed symbol data. Therefore, the total size of the compressed data is  $3 \times 4 + 2 \times 9 = 30$  bits, which corresponds to a compression of 47.38%.

Since the length of the dictionary index is much smaller than an  $m$ -bits symbol, the compression ratio is greater if more tails can be obtained from the dictionary. However, the dictionary must be reasonably small to keep the hardware overhead low. Therefore, selecting the entries of the dictionary is an important step in the dictionary-based compression. In the proposed scheme, the statistical method is used to maximize the number of encoded symbols by using their frequencies of occurrence. The algorithm to find the maximum number of encoded symbols is provided in Fig. 7.

The decompress architecture of the proposed dictionary-based compression scheme is consisted of the control logic and the dictionary logic. The control logic is comprised of a small finite-state machine (FSM), a register and counter. The control logic checks the prefix of the first bits of the codeword which is shifted from ATE, and decides the following tail is the dictionary index or uncompressed test data. If the tail is the dictionary index, the  $m$ -bits code is read from the dictionary and driven into the multiple scan chains. On the other hands, if the tail is the uncompressed test data, the intact  $m$ -bits tail is driven into the multiple scan chains. The dictionary logic is implemented by using the combination logic circuit. The size of dictionary depends on the length of

dictionary index. The hardware overhead for the dictionary logic is a weak point of the proposed scheme like other dictionary-based schemes. However, the hardware overhead can be kept reasonable size because we use a small amount of the test data vector which is already compressed by the scan chain compaction and the dictionary has fixed-length indices. Moreover, the proposed dictionary-based compression scheme does not require multiple clock cycles and additional synchronization and handshaking between the CUT and the ATE.

## V. EXPERIMENTAL RESULTS

In this section, to verify the efficiency of the proposed method, we have implemented the proposed compression method in the C language on a Linux system. We generated the test cubes for the five largest ISCAS-89 fully scanned test bench circuits. These test data cubes were generated by TetraMax [13].

The results of the proposed compacted scan chain network scheme are presented in Table 1. In Table 1, we provide the number of vectors, the scan depth, and the ratio of don't care bits, and total bits of test data cube for the traditional single scan chain architecture. The results of the proposed compacted scan chain network methodology are presented in next columns. The number of test vectors, provided in column 6, is reduced in several circuits. If the compacted test vector of scan chain is exactly equal to other compacted test vectors, the test vector can be rejected from the test cube. The scan depth equals the number of the scan slices in the scan chain network, which is reduced by the proposed methodology. The total bits of compressed test bits are also reduced by the decrement of test vectors and scan depth. The area overhead denotes the percentage ratio of the number of 2-input compaction XOR gates to the total number of gates in the circuit.

According to these results, the scan chain compaction scheme provides the reduction of test data volume and test application time simultaneously with small hardware

Table. 1 Comparison of the proposed reconfigured scan chain network method with the traditional single chain architecture

Circuit	Single scan chain				Proposed reconfigured scan chain network				
	Test vectors	Scan depth	Don't care ( % )	Total bits	Test vectors	Scan depth	Comp. bits	Run-time ( sec )	Area ( % )
s13207	111	638	86.70	70818	108	256	27648	0.32	2.5
s15850	112	534	81.26	59808	95	300	28880	0.28	1.6
s35932	28	1728	61.85	48384	28	804	22624	4.86	1.1
s38417	412	1636	93.56	674032	142	830	118144	6.61	2.3
s38584	140	1426	86.64	199640	140	850	119840	3.70	0.3

Table. 2 Comparison of the compression ratio with the previously proposed techniques

Circuit	FDR [2]	VIHC [3]	RL-HC [4]	SHC [5]	OSHC [6]	SN [11]	Proposed	# of SC
s13207	69.59	71.15	69.51	66.95	72.84	59.87	<b>97.56</b>	128
s15850	56.82	58.99	57.83	52.61	62.39	51.71	<b>88.33</b>	101
s35932	44.07	51.08	55.08	50.71	52.22	53.24	<b>93.86</b>	115
s38417	85.17	86.90	89.44	79.87	82.81	82.47	<b>90.68</b>	115
s38584	60.84	62.33	61.52	57.80	66.66	39.97	<b>71.33</b>	32

Table 3 Comparison of the application time (clock cycle)

<i>Circuit</i>	<i>VIHC [3]</i>	<i>SN [11]</i>	<i>Proposed</i>
s13207	70818	27648	<b>1728</b>
s15850	59808	28880	<b>6980</b>
s35932	49384	22624	<b>2972</b>
s38417	674032	118144	<b>62832</b>
s38584	199640	119840	<b>57240</b>

overhead. However, in the certain circuits, such as s38584, the decrement of the test data is not sufficient. Then we combine the dictionary-based compression to achieve the more high compression ratio and faster test speed.

Table 2 presents a comparison of the compression ratio with FDR coding [2], variable-length input Huffman coding [3], RL-Huffman coding [4], selective Huffman coding [5], optimal selective Huffman coding [6], and an adaptive scan network [11]. For each test bench circuit, we determine the size of dictionary that is 128 entries, because which provides good tradeoff between compression ratio and hardware overhead. The length of dictionary index is 7 bits. The column 8 shows the best compression ratio among the proposed compression schemes which provided by the number of multiple internal scan chains in the last column. The proposed compression scheme outperforms every previous compression schemes for all test bench circuits. The proposed scheme compress the test data twice with the scan chain compaction and the dictionary-based compression, then they can be achieved the highest test data compression ratio.

Table 3 shows the comparison of the application time in clock cycle of the compression schemes. The clock cycle is defined the number of the clock count that drive the test data pattern to the scan chain. The VIHC coding and the other code-based commonly use the original full-scan chain, therefore these schemes need more clock cycle for whole test. As several schemes of the code-based compression use multiple clocks to reduce this application time, but they impose additional synchronization and handshaking between the SoC and ATE. The adaptive scan network scheme [11] reconstructs the scan chain and reduces the scan depth, which can reduce the clock cycle of the test application time. The proposed compression scheme provides best application test time by using the scan chain compaction and dictionary-based compression.

The hardware overhead of the proposed compression scheme is slightly bigger than other compression schemes because of the dictionary logic. However, the hardware overhead can be kept reasonable size because we use the proposed dictionary which has fixed-length indices and the small amount of the test data vector which is already compressed by the scan chain compaction. Moreover, the proposed compression architecture can be used for the multiple scan chain architecture with only one decompressor, on the other hands; other code-based compression schemes need more decompressors according to the number of scan

chains.

## VI. CONCLUSION

In this paper, we proposed an effective hybrid test data compression method using the scan chain compaction and the dictionary-based compression scheme. We have shown how proposed test data compression method can be used to reduce the test data volume and test application time for SoC. The proposed compression method delivers compression patterns from the ATE to the chip and drives a large number of multiple internal scan chains using only a single ATE input and output. The test data volume is compressed twice with the scan chain compaction and the dictionary-based compression, then they can be achieved the highest test data compression ratio and the fastest application time. The proposed compression scheme does not require multiple clock cycles, additional synchronization, and handshaking between the circuit and the ATE. Experimental results for the ISCAS-89 test benches show that the test data volume and testing time for the proposed method are less than those for a number of recently-proposed test data compression methods.

## REFERENCES

- [1] R. Chandramouli and S. Pateras, "Testing Systems on a Chip," *IEEE Spectrum*, pp. 42-47, Nov. 1996.
- [2] A. Chandra and K. Chakrabarty, "Frequency-directed run-length (FDR) codes with application to system-on-a-chip test data compression," *Proc. VLSIT est Symp.*, pp. 42-47, 2001.
- [3] P. T. Gonciari, B. Al-Hashimi and N. Nicolici, "Improving compression ratio, area overhead, and test application time for system-on-a-chip test data compression/decompression", *Proc. Design, Automation and Test in Europe Conf.*, pp. 604-611, 2002.
- [4] M. Nourani, and M. Tehranipour, "RL-Huffman encoding for test compression and power reduction in scan application," *ACM Trans. Design Automat. Electron. Syst.*, vol. 10, no. 1, pp. 91-115, Jan. 2005.
- [5] A. Jas, J. Ghosh-Dastidar and N. A. Touba, "An efficient test vector compression scheme using selective Huffman coding", *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2003, 22, pp. 797-806
- [6] X. Kavousianos, E. Kalligeros, and D. Nikolos, "Optimal Selective Huffman Coding for Test-Data Compression," *IEEE Trans. Computers*, vol. 56, pp. 1146-1152, 2007
- [7] A. Jas, J. Ghosh-Dastidar and N. A. Touba, "Scan vector compression/decompression using statistical coding," *Proc. VLSIT est Symp.*, pp. 114-120, 1999.
- [8] S. M. Reddy, K. Miyase, S. Kajihara, and I. Pomeranz, "On test data volume reduction for multiple scan chain design", *Proc. VLSIT est Symp.*, pp. 103-108, 2002.
- [9] L. Lei and K. Chakrabarty, "Test data compression using dictionaries with fixed-length indices [SOC testing]," in *VLSI Test Symposium, 2003. Proceedings. 21st*, 2003, pp. 219-224.
- [10] F.F. Hsu, K. M. Butler and J. H. Patel, "A case study on the implementation of Illinois scan architecture", *Proc. Int. Test Conf.*, pp. 538-547, 2001.
- [11] O. Sinanoglu., "Construction of an adaptive scan network for test time and data volume reduction." *IET. Computers & Digital Tech.*, Vol. 2, no. 1, pp. 12-22, 2008.
- [12] M. Garey, and D.S. Johnson, "Computers and intractability: a guide to the theory of NP-completeness", *Freeman, 1979, 1st edn.*
- [13] TetraMax Reference Manual. Release 2004. 12, Synopsys Inc., Mountain View, CA, 2001.