FCSCAN: An Efficient Multiscan-based Test Compression Technique for Test Cost Reduction

Youhua Shi, Nozomu Togawa, Shinji Kimura*, Masao Yanagisawa, and Tatsuo Ohtsuki

Dept. of Computer Science
Waseda University, Japan

*Grad. School of IPS
Waseda University, Japan

e-mail: shi@yanagi.comm.waseda.ac.jp

Abstract— This paper proposes a new multiscan-based test input data compression technique by employing a Fan-out Compression Scan Architecture (FCSCAN) for test cost reduction. The basic idea of FCSCAN is to target the minority specified 1 or 0 bits (either 1 or 0) in scan slices for compression. Due to the low specified bit density in test cube set, FCSCAN can significantly reduce input test data volume and the number of required test channels so as to reduce test cost. The FCSCAN technique is easy to be implemented with small hardware overhead and does not need any special ATPG for test generation. In addition, based on the theoretical compression efficiency analysis, improved procedures are also proposed for the FCSCAN to achieve further compression. Experimental results on both benchmark circuits and one real industrial design indicate that drastic reduction in test cost can be indeed achieved.

I. Introduction

The central issue in manufacturing test has always been how to apply sufficient test data to a design to ensure that the highest quality is reached, while at the same time to minimize test cost. Nowadays, almost all of the current design-for-test (DFT) techniques start with a baseline of scan technology. However, the chip complexity continuous increasing, which results in excessive test data volume even for single-stuck-at faults with single-detection [1]. In conventional external testing, this huge amount of test data must be stored on the external automatic test equipment (ATE) and be transferred to and from the circuit-under-test (CUT) through the limited test channels. This poses a serious problem on manufacturing test because, as test data volume increases, it takes more tester buffer space to hold the complete test set, and longer to deliver the test set through limited test channels, both leading to higher test cost. Therefore, to reduce tester storage and tester channel bandwidth requirements for million-gates designs is recognized as an extremely important problem and, consequently, is receiving a lot of attention in the past five years.

In the literature numerous papers have been published on test cost reduction using test data compression, which involves employing on-chip DFT structures to reduce the amount of data stored on the ATE and thereby shorten the time it takes to load or observe the scan chains. An important class of test data compression (TDC) techniques involves using an on-chip linear decompressor to decompress test vectors. This includes techniques based on linear feedback shift register (LFSR) reseeding [2, 3] and combinational linear expansion circuits [4]. There are also some commercial tools based on LFSR reseeding combined with on-chip decompression developed recently including Mentor Graphics' TestKompress [5], SmartBIST from IBM/Cadence [6] and Synopsys' DBIST [7]. Unfortunately, although these techniques can achieve high compression ratios, for an efficient implementation, most of them need to be combined with fault simulation or interleaved with automatic test pattern generation (ATPG). Another group of compression techniques uses lossless source coding for test data reduction, such as selective Huffman coding [8], run-length coding [9], and dictionary coding [10]. However, when applied to multiscan-based designs, the effectiveness of these methods is limited either by the additional hardware overhead or or the increased test time.

In addition, there are some methods focused on reducing external test channels to achieve great compression for designs with multiple scan chains. A technique using a single input supporting multiple scan chains was proposed in [11], but its application is limited to test multiple independent full scan circuits in parallel. The Illinois scan architecture [12] overcomes this limitation by using two modes of scan operation, parallel scan and serial scan. CircularScan [13] configures the scan chains in a circular form, enabling the generation of the next pattern from the captured response. CircularScan efficiently overcomes the tester channel bandwidth limitation, however it introduces a new problem on the test diagnosis process due to the undeterministic property of the test response. The approaches proposed in [14] and [15] explored the logic dependencies of the internal scan chains to construct a simple logic gates based decompression network, so that a great number of scan chains could be driven by a limited number of external scan channels and test cost is reduced.

In this paper we propose a new DFT technique – Fan-out Compression Scan Architecture (FCSCAN), to drastically reduce test cost for multiscan-based designs. The basic idea of the proposed FCSCAN technique is to only encode the minority specified 1 or 0 bits (either 1 or 0) (to be referred as *coded bits* in this paper) in scan slices for compression. While retaining the original number of scan chains, it drastically reduces input test data volume as well as the number of required test channels for precomputed test sets of multiscan-based designs so as to reduce test cost. To be mentioned that we assume

0-7803-9451-8/06/$20.00 ©2006 IEEE.

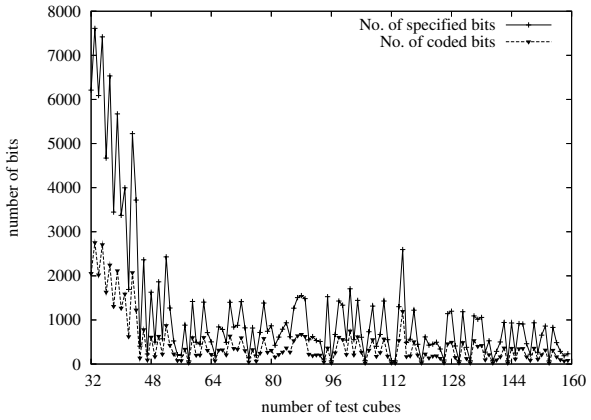


Fig. 1. Test cube profile for one industrial circuit - ASIC 1

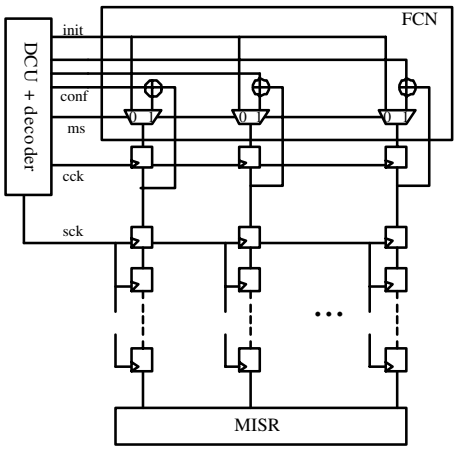


Fig. 2. The proposed fan-out compression scan (FCSCAN) architecture

the test response could be compacted using a Multiple Input Signature Register (MISR) or some other techniques [16], and we only consider the input test data reduction in this work. The proposed FCSCAN technique is very easy to implement with small hardware overhead and it does not need any special ATPG for test generation. In addition, starting with the analysis on the compression efficiency of FCSCAN, improved procedures by exploring the linear dependencies of the internal scan chains are also proposed for further compression. Unlike previous works, the computation complexity is greatly reduced with better compression. Experimental results indicate that drastic reduction in test cost can be indeed achieved using the proposed FCSCAN technique.

The rest of this paper is organized as follows: Section 2 presents the FCSCAN technique. Section 3 makes an analysis on the compression efficiency of the FCSCAN scheme and then describes the procedures for improvement. Finally experimental results and conclusions are given in Section 4 and 5, respectively.

II. FCSCAN Technique

In practice, in an ATPG-based test set even with the application of state-of-the-art dynamic and static test pattern compaction techniques, only 1%-10% of the total test data is specified with logic values for fault detection, and the other larger part of bits are don't cares. One example is shown in Figure 1, where ASIC 1 is a real industrial circuit used in our work. Due to the large fraction of don't-cares in test set, most of the test input data compression techniques as referred above focus on the don't-cares and utilize them for test data compression.

In our work, in addition to low specified bit densities, we have further studied the care bits information and found some other useful scan test properties. Specifically, in each scan slice, the number of the minority specified 1 or 0 bits (either 1 or 0) is definitely to be less than half of the total number of specified bits. In this paper we call the specified bits with the minority specified density in the current slice as **coded bits** and the coded bit density in the current slice, P_{ci}, is equal to $min(P_i(0), P_i(1))$. Figure 1 also illustrates this observation for ASIC 1, where there are 128 scan chains and the specified bit density is 12%. It can be seen that the total number of coded bits is much less than that of specified bits. With this important observation greater compression can be expected by loading only the coded bits information other than loading all the specified bits. Details will be illustrated as follows.

Figure 2 shows the proposed FCSCAN architecture, which enables the scan chains to be assigned proper data through one of the two loading mode: broadcast mode and configuration mode.

- Broadcast mode – all the internal scan chains are loaded the same value, which is similar to the parallel scan mode in Illinois Scan Architecture [12].
- Configuration mode – in this mode according to the values supplied from the test channels, an individual scan chain is selected to be inverted on the pre-loaded values in Broadcast mode.

The proposed FCSCAN architecture mainly consists of a decompression control unit (DCU), a flip configuration network (FCN) and a $\log_2 N$-to-N decoder. The DCU is a finite-state machine, which is responsible for controlling the decompression process, generating the mode selection signal *ms* and controlling the scan clock *sck* for the CUT. The flip configuration network is a combinational block composed of one XOR gate and one MUX for each internal scan chain. The multiplexers are controlled by the *ms* signal provided from DCU. When *ms* is 0, the FCSCAN is operated in broadcast mode, otherwise it runs in configuration mode. The i^{th} bit of the $\log_2 N$-to-N decoder is XORed with the feedback of the first scan cell in the i^{th} scan chain that allows for the pre-existing values to be kept or inverted in the scan cells of the associated chain.

Based on the FCSCAN architecture, a compressed form of a scan slice contains the following:
(1) an initial vector: to indicate the initial value for the scan cells (1 bit) and the number of coded bits in the current slice (M-1 bits). Since the number of internal scan chains is N_{sc} and the number of coded bits in any scan slice would be less than $0.5 * N_{sc}$, when $M = \lceil \log_2 N_{sc} \rceil$, it can guarantee a regular test application.
(2) configuration vectors: to indicate the positions of the coded bits needed by the decompressor to specify an inversion on any bits in the current slice.

TABLE I
FCSCAN COMPRESSION EXAMPLE

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
s1:	0	0	0	1	0	0	X	X	X	X
s2:	1	0	X	0	0	1	1	X	X	X
s3:	1	X	0	1	X	X	X	X	X	X
s4:	0	0	1	1	1	X	1	X	0	X
s5:	X	0	X	1	X	X	1	1	X	1
s6:	0	X	0	X	X	0	X	X	0	X
s7:	X	1	1	X	X	X	X	X	X	X
s8:	1	X	X	0	X	1	1	X	1	X

(a)

	init vector	conf. vector	final vec
s1	0 0 0 1	0 1 0 0	0 0 0 1 0 0 0 0 0 0
s2	1 0 1 1	0 0 1 0	
		0 1 0 0	
		0 1 0 1	1 0 1 0 0 1 1 1 1 1
s3	1 0 0 1	0 0 1 1	1 1 0 1 1 1 1 1 1 1
s4	1 0 1 1	0 0 0 1	
		0 0 1 0	
		1 0 0 1	0 0 1 1 1 1 1 1 0 1
s5	1 0 0 1	0 0 1 0	1 0 1 1 1 1 1 1 1 1
s6	0 0 0 0	- - - -	0 0 0 0 0 0 0 0 0 0
s7	1 0 0 0	- - - -	1 1 1 1 1 1 1 1 1 1
s8	1 0 0 1	0 1 0 0	1 1 1 0 1 1 1 1 1 1

(b)

Example I - Table I (a) shows an example of test data for multiple scan chains. There are 10 scan chains ($c_1, c_2, \ldots, c_{10}$) and 8 scan slices ($s_1, s_2, \ldots, s_8$). In the proposed method, instead of using 10 external test channels, only 4 are used. Look at the second slice (s_2), in the first stage, we load an initial vector $i_2 = 1011$ to the decompressor. In this paper we assume the leftmost bit of the initial vector is the initial value for the current slice, and the other three bits are used to indicate the number of coded bits, which is 011 (i.e. there are three bits to be flipped). Thus by clocking *cck* as shown in Figure 2, the scan cells are filled with the same logic values as the initial data (i.e. the current content in the scan cells is $v = 1111111111$). In the second stage, we load the configuration vector $vc_1 = 0010$ into the decompressor, and clock *cck* again. Thus the *conf* signal is 0100000000, and an XOR operation is performed on the *conf* signal with the previously loaded scan slice v. Then the second bit position of the v is flipped while at the same time the counter is decremented. The configuration process is repeated until the counter reaches zero. Then in the next cycle, shift the current slice into the next scan cells by enabling the clock *sck*; while at the same time comes the next initial block $i_3 = 1001$. The compressed test data is shown in Table I (b). In our work the decompression logic is implemented in VHDL and synthesized using Synopsys' design compiler to access the hardware overhead of the decompressor. The synthesized circuit (DCU) for the above example only contains two flip flops and 19 combinational gates, which is very small.

III. Optimization for Further Compression

In this section we first make an analysis on the compression efficiency of the proposed FCSCAN technique, from which we derive that the compression efficiency is affected by two parameters, such as the number of test channels and the total number of coded bits. Consequently, an optimized solution is proposed to further the compression through reducing the two parameters.

A. Compression Analysis

In our work, we use the compression ratio γ to analyze the compression efficiency, which is defined as the ratio of the compressed data volume ($|T_E|$) to the original uncompressed data size ($|T_D|$) (i.e. $\gamma = |T_E|/|T_D|$), where γ must always be less than unity for the compression method to be effective.

Equation 1 computes the number of bits in the compressed test set ($|T_E|$) using the proposed FCSCAN technique.

$$|T_E| = \sum_{i=1}^{N_v} \sum_{j=1}^{N_{sl}} M * (1 + n_{i,j}) \quad (1)$$

Where M is the number of external scan channels, N_v is the total number of test patterns, N_{sl} represents the maximum scan length, and $n_{i,j}$ is equal to the number of coded bits in slice $s_{i,j}$. The compression ratio γ therefore can be calculated as following.

$$\begin{aligned} \gamma &= \sum_{i=1}^{N_v} \sum_{j=1}^{N_{sl}} \frac{M * (1 + n_{i,j})}{N_v * N_{sl} * N_{sc}} \\ &= \frac{M}{N_{sc}} + M * P_c \end{aligned} \quad (2)$$

Where the product of $N_v * N_{sl} * N_{sc}$ amounts to the uncompressed test data volume $|T_D|$, while P_c is the coded bit density that is equal to the total number of coded bits divided by $|T_D|$.

Look at the example shown in Table I again, where $N_{sc} = 10$, $M = 4$ and for the 8 test cubes the number of coded bits is $(1, 3, 1, 3, 1, 0, 0, 1)$ respectively. Thus the coded bit density is $10/80 = 12\%$, the total number of test data after compression is 72 bits and for this example we can achieve 10% savings in the test data volume. Since this test data example is very small and the filling rate ($P_s = 48\%$) is relatively high when compared with typical industry designs ($P_s \approx 1\% - 10\%$), greater compression could be expected for the larger circuits. For example, if we have a CUT with 1000 scan chains and the average filling rate is 10%, using the proposed FCSCAN scheme only 10 external test channels are required for a regular test application. Furthermore, in the worst case (i.e. $P_c \leq 0.5 * P_s = 5\%$) our method can still achieve 49% reduction in test input data volume no matter whether the specified bit distribution is uniform or non-uniform.

As can be seen from Equation 2, the compression efficiency of FCSCAN depends on two parameters, such as the number of test channels (M) and the total number of coded bits ($N_c = \sum_{i=1}^{N_v} \sum_{j=1}^{N_{sl}} n_{i,j}$). If the required test channels M and/or the total number of coded bits decreases, then the compression level will increase provided further reduction in test cost. Reducing M can help to reduce the number of bits for both initial vectors and configuration vectors, while reducing total number of coded bits could be used to decrease the number of configuration vectors. An improved solutions based on the FCSCAN scheme is presented in the following subsections, while keep both of these parameters intact.

1. For a precomputed test set, descendingly sort the scan chains according to the number of specified bits $S = (c'_1, c'_2, \ldots, c'_n)$ and $SC = \emptyset$;
2. Loop until no entry left in S
   i) select the first entry c_k in S;
   ii) for each c_i in SC
   iii) if ($d(c_k, k_i) = 0$ or $d(c_k, k_i) = N_{sc}$)
   // d: hamming distance
   update k_i and delete c_k from S;
   iv) else
   copy c_k to SC and delete it from S;
3. According to the logic dependencies between scan chain inputs, build the fan-out network only with inverters.

Fig. 3. Scan chain clustering algorithm

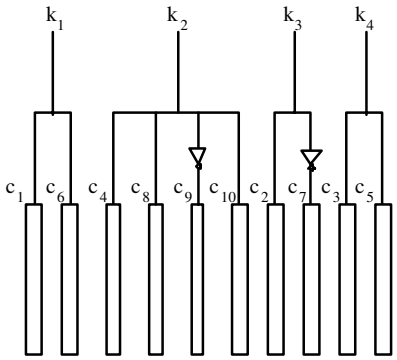
B. Minimizing Required Test Channels

The first step is to minimize the required test channels (M). This is achieved by exploiting the compatibilities among the internal scan chains for a given test set to construct a simple fan-out structure between the decompressor output and the scan chains.

In the literature, there are a number of published papers based on constructing such a simple logic fan-out network structure through compatibility analysis for test cost reduction. Our method generalizes the design techniques introduced in [17, 14] by constructing a single-level fan-out structure with inverters. This structure is chosen because it is very low cost, but still capable of achieving significant compression for the proposed FCSCAN technique. Figure 4(a) shows an example of such a network through exploiting the compatibility and inverse compatibility of the internal scan chains.

To build such a fan-out structure needs to divide the internal scan chains into scan clusters, which can be viewed as a clique-partitioning problem such as illustrated in [15, 14]. However due to the $NP-$ completeness of the clique covering problem, finding an optimal solution might require exponential time. Therefore a simple heuristic summarized in Figure 3 has been developed.

The proposed scan chain clustering (SCC) algorithm has two steps. In the first step, it descendly sorts the scan input of each internal scan chain into S according to the corresponding number of specified bits. The computation complexity of this step is $O(n^2)$, where n is the number of scan chains. In the second step, initially the scan cluster set SC is empty, and the first scan chain input c'_1 in S is moved from S to SC as the first entry (k_1). Then the remaining scan chain inputs (c'_i) are analyzed one by one according to their corresponding hamming distance with the entries in SC. If the hamming distance between c'_i and k_k is 0 or N_{sc}, i.e. c'_i and k_k are directly/inversely compatible, then remove c'_i from SC and properly fill the don't-cares in k_k; otherwise copy c'_i to SC and delete it from S. This process is repeated until the set S is empty. In addition, we also set some constraints in our algorithm because our goal is to divide the N_{sc} internal scan chains into m scan clusters, where $\lceil \log_2 m \rceil < \lceil \log_2 Nsc \rceil$, so as to reduce M. Thus when the number of entries in SC is larger than $2^{\lfloor \log_2 N_{sc} \rfloor}$, the process is stopped. To do this is very useful, because it can not only help to reduce the useless computation but also preserve the compression space. The computation complexity of this step is $O(n \log_2 n)$, where $n = N_{sc}$, therefore the total computation complexity of the proposed SCC algorithm is: $O(n^2) + O(n \log_2 n) \approx O(n^2)$. While for the previous approaches based on scan chains clustering such as [14, 15], the computation complexity is $O(n^4)$ and $O(n^3)$ respecively.



(a)

	init vec	conf. vec	scan in k_i 1 2 3 4	final vec c_i 1 2 3 4 5 6 7 8 9 10
s1	0 0 1	0 1 0	0 1 0 0	0 0 0 1 0 0 1 1 0 1
s2	0 0 1	0 0 1	1 0 0 0	1 0 0 0 0 1 1 0 1 0
s3	1 0 1	1 0 0	1 1 1 0	1 1 0 1 0 1 0 1 0 1
s4	0 1 0	0 1 0		
		1 0 0	0 1 0 1	0 0 1 1 1 0 1 1 0 1
s5	0 0 1	0 1 0	0 1 0 0	0 0 0 1 0 0 1 1 0 1
s6	0 0 1	0 1 0	0 1 0 0	0 0 0 1 0 0 1 1 0 1
s7	1 0 0	- - -	1 1 1 1	1 1 1 1 1 1 0 1 0 1
s8	0 0 1	0 0 1	1 0 0 0	1 0 0 0 0 1 1 0 1 0

(b)

Fig. 4. Compression example using scan chain clustering (a)single-level fan-out structure with inverters and (b) the compressed data with final test set

Example II - We use the same test data as in Table I. Initially, N_{sc} is 10, so we set the constraint to the number of scan clusters, when $m > 7$ the procedure is terminated. Then the SCC algorithm is applied, which divides the 10 scan chains into 4 scan clusters $k_1 = \{c_1, c_6\}$, $k_2 = \{c_4, c_8, c_9, c_{10}\}$, $k_3 = \{c_2, c_7\}$ and $k_4 = \{c_3, c_5\}$, as shown in Figure 4(a). Three scan inputs can fan out to 10 scan chains, thus M is reduced from 4 to 3. Figure 4(b) shows the corresponding compressed data with the final fully specified test vectors. In this example, the total number of test data after compression is 48 bits, and we can achieve 40% savings in the test data volume. When compared with that in Example I, 24 bits are reduced. In addition, the number of coded bit is reduced from 10 to 8. This is an additional benefit of the scan chain clustering algorithm.

C. Reducing Coded Bits

While dividing scan chains into directly/inversely compatible scan clusters provides a lot of benefits for test cost reduction, we can achieve additional compression by reducing the number of coded bits for the FCSCAN scheme. Since the basic idea of the FCSCAN is to encode the minority specified 1 or 0 bits (either 1 or 0) in scan slices for compression, we can alter and reshape the space of the scan inputs (i.e. the output space of the decompressor) by inserting some appropriate inverters.

Figure 5 shows the procedure how to reshape the scan inputs to reduce the total number of coded bits. Our heuristic seeks to

1. Counting the number of coded bits for each scan inputs (k_i);
2. Loop until for each scan input, the number of coded bits is less than half of the number of specified bits ($k_i < 0.5 * s_i$)
   i) select k_j with the maximum coded bits;
   ii) invert k_j;
   iii) add an inverter to the fan-out structure;
   iv) re-counting k_i for each scan inputs;
3. Optimization for reducing the number of inverters.

Fig. 5. Heuristic for coded bits reduction

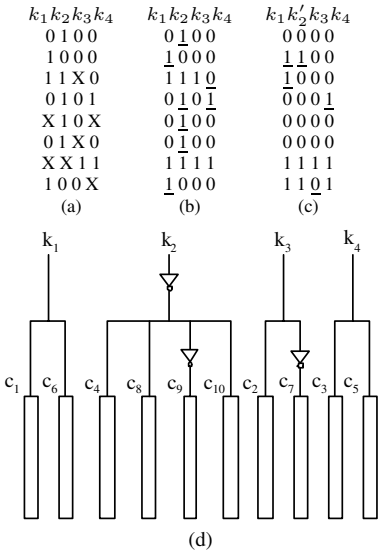


Fig. 6. Example for reducing coded bits

reduce the number of coded bits in the scan clusters extracted from Section 3.2 by adding appropriate inverters between the decompressor and the fan-out structure.

Example III - Figure 6 illustrates the example how to reduce coded bits for the previous test data after scan clustering. Figure 6 (a) shows the test data after scan chain clustering, in which there are some don'-cares left. After carefully filling the don't-cares, the number of coded bits is 7 as shown in Figure 6 (b). Since k_2 has 4 coded bits, which is larger than half of its specified bits (i.e. 7 bits). Then k_2 is inverted, and now the total number of coded bits for the whole test set is reduced to 5 when carefully fill the left don't-cares as shown in Figure 6 (c). When compared with Example II, three coded bits are reduced with only one additional inverter.

IV. Experimental Results

In this section, we present experimental results for the five largest ISCAS'89 benchmark circuits and one real industry design (ASIC 1) to validate the effectiveness of the proposed FCSCAN technique. For each circuit, a commercial ATPG tool is used to generate test cubes with dynamic and static compaction providing 100% coverage of detectable faults. In our work, We first apply the scan clustering technique to minimize the number of required channels based on the test cube set, and then use the method proposed in Section 3.3 to reduce the number of coded bits and construct the fan-out structure using inverters. Finally the basic FCSCAN technique is applied.

Table II presents the results on the compression efficiency of the proposed FCSCAN technique for the six circuits with varying number of scan chains. In the table, the name of circuits, the internal scan cells, and the number of test vectors are shown in the first three columns, respectively. Because we focus on test cost reduction for large designs with multiple scan chains, thus we set the number of scan chains to be 50, 100 and 200 as representatives for the benchmark circuits and 100,200 and 400 for ASIC 1. The size of uncompressed test data ($|T_D|$) is shown followed by the average specified bit density (P_s) and the varying number of scan chains (N_{sc}). The compressed results for the basic FCSCAN scheme and the improved technique proposed in Section 3 are also listed, where $|T_E|$ and M are the compressed data size and the number of external test channels, respectively. In the improved scheme column, *cluster* indicates the number of scan clusters after applying the heuristic of Figure 3, and the number of added inverters is also summarized.

Because we assume that the scan chains already exist in the CUT and scan chain reordering is prohibited, the data shown for FCSCAN do not use the scan chain reordering methods. From the table, we can observe that as the number of scan chains increases, FCSCAN leads to greatly reduced input test data volume with small hardware overhead. It is notable that for the four largest benchmark circuits (s15850, s35932, s38417 and s38584), the volume of compressed test data is less than the number of specified bits in the original test set.

Table 3 shows a comparison of the results for the proposed method with CircularScan [13], 9C coding [19], Dictionary Coding with Correction (DCC) [20] and Frugal Linear network (FLN) [14], which are the representatives of the recently introduced test data compression schemes. The result of the Mintest ATPG-compacted test sets [18] is also listed for the sake of comparison. The result listed for the proposed method is the minimum size shown in boldface in Table II. Since different ATPG tools may be used, Table 3 shows both the number of vectors and the total number of compressed bits for each case. As can be seen from the table, the results obtained using the proposed FCSCAN scheme are better than those of CircularScan [13] and DCC [20] for all the circuits. When compared with 9C coding [19] and FLN [19], our method achieves higher compression for all but one of the circuits (s35932). We should note that when compared with industrial designs, the used benchmark circuits are very smaller in size and the specified bit density is relatively higher, which will limit the compression efficiency of the proposed FCSCAN technique.

Finally, it must be mentioned that although the proposed FCSCAN is very easy to implement, it really leads to drastic reduction in test cost, reducing both test data volume and the test channel requirement, which is especially effective for the designs with a great number of scan chains.

V. Conclusions

We have presented a FCSCAN technique that can reduce both input test data volume and external test channel requirement with small hardware overhead for large designs. Improved procedures have also been proposed to help the FCSCAN to further the compression. Experimental results for the

TABLE II
COMPRESSION RESULTS OF FCSCAN

Circuits	No. of FFs	No. of vectors	Size of original bits	P_s (%)	N_{sc}	FCSCAN M	FCSCAN T_E	Improved FCSCAN cluster	M	T_E	P_c	No. of inverters
s13207	700	251	175,700	4.36%	50	6	30,564	50	6	29,520	0.8%	14
					100	7	25,830	61	6	18,132	0.72%	21
					200	8	26,304	30	5	**10,290**	0.6%	46
s15850	611	148	90.428	12.40%	50	6	24,024	49	6	20,766	1.7%	11
					100	7	26,873	28	5	12,415	1.6%	39
					200	8	30,056	14	4	**7,072**	1.3%	48
s35932	1763	35	61,635	14.10%	50	6	22,722	50	6	17,916	2.8%	10
					100	7	23,828	58	6	13,794	2.7%	20
					200	8	27,176	27	5	**8,045**	2.1%	35
s38417	1664	183	304,512	13.40%	50	6	88,488	31	5	58,515	1.8%	20
					100	7	92,120	60	6	53,382	1.9%	22
					200	8	10,5752	22	5	**29,550**	1.4%	29
s38584	1464	288	421,632	6.01%	50	6	84,726	29	5	53,740	0.5%	17
					100	7	80,416	22	5	32,190	0.5%	24
					200	8	85,896	13	4	**21,020**	0.7%	47
ASIC 1	8017	246	1,972,182	12%	100	7	733,110	60	6	510,048	3.3%	21
					200	8	806,448	59	6	391,842	2.8%	35
					400	9	951,723	27	5	**292,075**	2.7%	58

TABLE III
COMPARISON WITH PREVIOUS WORKS

Circuits	Mintest [18] vectors	Mintest [18] T_E	Circular Scan [13] vectors	Circular Scan [13] T_E	9C [19] vectors	9C [19] T_E	DCC [20] vectors	DCC [20] T_E	FLN [14] vectors	FLN [14] T_E	Proposed vectors	Proposed T_E
s13207	233	163,100	299	62,415	236	28,893	236	31,772	317	13,948	251	10,290
s15850	94	57,434	186	62,408	126	25,143	126	27,721	309	13,596	148	7,072
s35932	11	19,393	34	46,593	16	3,029	N/A	N/A	38	836	35	8,045
s38417	68	113,152	270	250,016	99	59,024	99	84,896	678	63,732	183	29,550
s38584	110	161,040	251	162,909	136	74,863	136	65,396	477	25,758	288	21,020

larger benchmark circuits and a real industrial ASIC demonstrate that FCSCAN provides significant improvement over other recent works. One notable result is that the test data volume could be reduced to less than or close to the number of specified bits in the original test set.

REFERENCES

[1] H. Furukawa F. Hsu S. Lin S. Tsai K. Abdel-hafez L. Wang, X. Wen and S. Wu. VirtualScan: A New Compressed Scan Technology for Test Cost Reduction. In *Proceedings IEEE International Test Conference (ITC)*, pages 916–925, October 2004.

[2] J. Rajski S. Hellebrand and el. Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers. *IEEE Transactions on Computers*, 44(22):223–233, February 1995.

[3] B. Pouya A. Jas and N. A. Touba. Virtual scan chains: a means for reducing scan length in cores. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 73–78, May 2000.

[4] I. Bayraktaroglu and A. Orailoglu. Test volume and application time reduction through scan chain concealment. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 151–155, June 2001.

[5] M. Kassab J. Rajski, J. Tyszer and N. Mukherjee. Embedded Deterministic Test. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(5):776–792, May 2004.

[6] B. Koenemann et al. A smartBIST variant with guaranteed encoding. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 325–330, November 2001.

[7] S.Patel P.Wohl, J.Waicukauski and M.Amin. Efficient Compression and Application of Deterministic Patterns in a Logic BIST architecture. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 566–569, June 2003.

[8] M.-E. Ng A. Jas, J. Ghosh-Dastidar and N. A. Touba. An Efficient Test Vector Compression Scheme Using Selective Huffman Coding. *IEEE Transactions on Computer-Aided Design*, 22(6):797–806, November 2003.

[9] A. Chandra and K. Chakrabarty. Frequency-directed run-length (FDR) codes with application to system-on-a-chip test data compression. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 42–47, May 2001.

[10] K. Chakrabarty L. Li and N. Touba. Test data compression using dictionaries with selective entries and fixed-length indices. *ACM Transactions on Design Automation of Electronic Systems*, 8(4):470–490, October 2003.

[11] J.-J. Chen K.-J. Lee and C.-H. Huang. Using a single input to support multiple scan chains. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 74–78, November 1998.

[12] K. Butler F. Hsu and J. Patel. A case study on the implementation of the Illinois scan architecture. In *Proceedings IEEE International Test Conference (ITC)*, pages 538–547, October 2001.

[13] B. Arslan and A. Orailoglu. CircularScan A Scan Architecture for Test Cost Reduction. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1290–1295, March 2004.

[14] A. Orailoglu W. Rao and G. Su. Frugal Linear Network-Based Test Decompression for Drastic Test Cost Reduction. In *Proceedings International Conference on Computer-Aided Design (ICCAD)*, pages 721–725, November 2004.

[15] S. Kajihara L. Li, K. Chakrabarty and S. Swaminathan. Efficient Space/Time Compression to Reduce Test Data Volume and Testing Time for IP Cores. In *Proceedings IEEE VLSI Design (VLSID)*, pages 53–58, January 2005.

[16] S. Mitra and K. S. Kim. X-Compact: An efficient response compaction technique for test cost reduction. In *Proceedings IEEE International Test Conference (ITC)*, pages 311–320, October 2002.

[17] C. Chen and S. Gupta. A methodology to design efficient BIST test pattern generators. In *Proceedings IEEE International Test Conference (ITC)*, pages 814–823, October 1995.

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

[19] M. Nourani M. Tehranipour and K. Chakrabarty. Nine-coded Compression Technique with Application to Reduced Pin-count Testing and Flexible On-chip Decompression. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1284–1289, March 2004.

[20] C. S. Tautermann A. Wurtenberger and S. Hellebrand. Data Compression for Multiple Scan Chains Using Dictionaries with Corrections. In *Proceedings IEEE International Test Conference (ITC)*, pages 926–935, October 2004.