Efficient Space/Time Compression to Reduce Test Data Volume and Testing Time for IP Cores*

Lei Li[1], Krishnendu Chakrabarty[1], Seiji Kajihara[2] and Shivakumar Swaminathan[3]

[1]Dept. Electrical & Computer Engineering
Duke University, Durham, NC 27708
E-mail: {ll, krish}@ee.duke.edu

[2]Dept. Computer Sciences & Electronics
Kyushu Institute of Technology, Japan
E-mail: kajihara@cse.kyutech.ac.jp

[3]IBM Microelectronics
Research Triangle Park, NC
E-mail: s9s@us.ibm.com

***Abstract*— We present two-dimensional (space/time) compression techniques that reduce test data volume and test application time for scan testing of intellectual property (IP) cores. We start with a set of test cubes and use the well-known concept of scan chain compatibility to determine a small number c of tester channels that are needed to drive m scan chains ($c \ll m$). Next, we exploit logic dependencies between the test data for the scan chains to design a single-level decompression circuit based on two-input gates. We refer to these procedures collectively as width (space) compression. We then determine a small set of test patterns that can provide complete fault coverage when they are applied to the circuit under test using the c tester channels; this procedure is referred to as height (time) compression. In this way, structural information about the IP cores is not necessary for fault simulation, dynamic compaction, or test generation. The hardware overhead of the proposed approach is limited to the fan-out structure and a very small number of gates between the tester-driven external scan pins and the internal scan chains. Results are presented for the ISCAS-89 benchmarks and for four industrial circuits.**

I. Introduction

Recent advances in process technology have led to a rapid increase in the density of integrated circuits (ICs). The increased density and the need to test for new types of defects in nanometer technologies have resulted in a tremendous increase in test data volume and test application time. The test data volume for ICs in 2014 is projected to be as much as 150 times the test data volume in 1999 [10].

In addition to the increasing density of ICs, today's system-on-chip (SOC) designs also exacerbate the test data volume problem. The increase in test data volume not only leads to the increase of testing time, but the high test data volume may also exceed the limited memory depth of automatic test equipment (ATE). Multiple ATE reloads are time-consuming since data transfer from a workstation to the ATE hard disk or from the ATE hard disk to ATE channels are relatively slow; the upload time ranges from tens of minutes to hours [17]. While test application time for scan vectors can be reduced by using a large number of internal scan chains, the number of internal scan chains that can be driven by an ATE is limited in practice due to pin count constraints.

Test data volume for IP cores can be reduced by compressing the precomputed test set T_D provided by the core vendor to a much smaller data set T_E, which is stored in ATE memory. An on-chip decoder is used for pattern decompression to generate T_D from T_E during pattern application [3, 5, 19]. Such techniques are typically based on run-length codes and their variants, e.g., frequency-directed run-length (FDR) codes. However, most methods based on compression codes target single scan chains and they require complex synchronization between the ATE and the circuit under test. Test data volume reduction techniques based on on-chip linear decompression hardware [11, 14] and hybrid BIST [9] have also been presented in the literature. However, these techniques utilize structural information about the circuit under test, which prevents their applicability to IP cores.

In this paper, we present two-dimensional (space/time) compression techniques that reduce test data volume and testing time by allowing a large number of internal scan chains to be driven by a small number of tester channels. We start with a set of precomputed test cubes and use the well-known concept of scan chain compatibility to determine the number c of tester channels that are needed to drive m scan chains ($c \ll m$). Next, as sketched in Figure 1, we use the logic dependencies between the test data for the scan chains to design a decompression logic based on a single level of two-input gates. We refer to these procedures collectively as width (space) compression. We then determine a small set of test patterns that can provide complete fault coverage when they are applied to the circuit under test using the c tester channels; this procedure is referred to as height (time) compression. This approach does not require structural details of the IP core for fault simulation or test generation. It facilitates test pattern reuse because no additional test generation or fault simulation are required during system integration.

A number of related techniques for two-dimensional compression based on test cubes rely on the interleaving of the width and height compression procedures [1]. Such methods require tailored test development solutions that can support incremented test generation as the compression circuit is determined; thus even though these methods are non-intrusive in term of circuit redesign, they require structural information about the circuit under test. In contrast, height compression in this paper is carried out after width compression is accomplished, and without the use of any test generation. There is no loss in fault coverage because all the test patterns in T_D are applied to the circuit under test.

The steady increase in clock frequencies over the recent past has led to designs with a small number of gates between latches, or between latches and I/O pins. As a result, current-generation logic circuits have very short combinational logic depth, and a

*This research was supported in part by the National Science Foundation under grants CCR-9875324, CCR-0204077, and OISE-0403217, and by a graduate fellowship from the IEEE/ACM Design Automation Conference.

Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design (VLSID'05)
1063-9667/05 $20.00 © 2005 **IEEE**

large number of logic cones with very little overlap. This is in contrast to older circuits such as the ISCAS-85 benchmarks that tend to have a smaller number of overlapping logic cones. A consequence of the shallow logic depth is that test patterns in present-day circuits contain many don't-care bits. A commercial test pattern generator typically uses random fill to increase the likelihood of surreptitious fault detection. However, if X-fill is used during test generation and the test sets for the cores are delivered with the don't-care bits to the system integrator, an appropriate compression method can be used at the system level to reduce test data volume and testing time. This imposes no additional burden on the core vendor.

Experimental results for the ISCAS-89 circuits demonstrate that despite the inherent simplicity and low hardware overhead, the proposed technique provide significant improvement over other recent methods. We also present results for four production circuits from IBM, and show that compared to the methods employed for production testing of these circuits, the proposed technique leads to 10X reduction over compacted test sets in both test data volume and testing time.

The rest of the paper is organized as follows. In Section II, we present the proposed compression techniques. Section III discuss the impact of the number of ATE channels on the test application time. Experimental results and a comparison with related recent work for IP cores are presented in Section IV. Finally, Section V concludes the paper.

II. Proposed Approach

The proposed two-dimensional test data compression techniques consist of two steps, namely width compression and height compression. Consider a test set of p test cubes for m internal scan chains. The "width" of the test data is the number of internal scan chains m, the "height" of a test pattern is the scan chain length l, and the "height" of the test set is the product of l and the number of test patterns p. Here we assume that the scan chains are balanced. For unbalanced scan chains, don't-care bits can be used to pad the shorter scan chains. The first step in our approach is referred to as width compression, in which we use c ATE channels to drive the m scan chains, where $c \ll m$. Width compression is achieved by exploiting the compatibilities and the logic dependencies between the scan chains for the precomputed test cubes. In the second step, we determine a set of test patterns that can detect all the irredundant faults in the circuit; the test data is compressed now in the height dimension. Height compression is achieved through static compaction on the set of precomputed test cubes.

A. Width Compression

In the width compression step, c ATE channels are used to drive m scan chains, where c is much less than m. In this way, the test patterns are compressed in the width dimension. The parameter c is determined based on the compatibilities and the logic dependencies between the scan chains for different test cubes. The concept of scan chain compatibility and the associated fan-out structure have been utilized in a number of papers [1, 7] since the broadcast scan architecture was presented in [12]. In addition to the compatibilities between the scan chains, we also exploit the logic dependencies between the scan chains to reduce c.

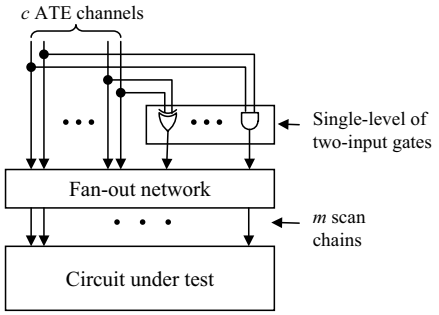
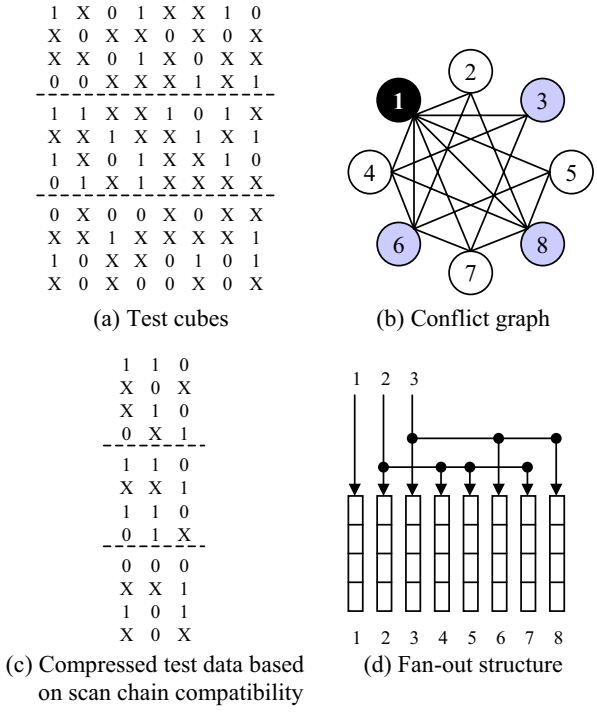


Fig. 1. Fan-out structure.

Let $b_{k,i,j}$ denote the data bit that is shifted into the jth flip-flop of ith scan chain for the kth test pattern. The compatibility of a pair of scan chains is defined as follows. The i_1th scan chain and the i_2th scan chain are mutually compatible if for any k $(1 \leq k \leq p)$ and j $(1 \leq j \leq l)$, $b_{k,i_1,j}$ and $b_{k,i_2,j}$ are either equal to each other or at least one of them is a don't-care. We next form a conflict graph from the test data and use graph coloring solution to obtain the number of ATE channels needed. The conflict graph G contains a vertex for each scan chain. If two scan chains are not compatible, they are connected by an edge in G. A vertex coloring of G yields the minimum number of ATE channels required to apply the test cubes with no loss of fault coverage (In the graph coloring solution, a minimum number of colors is determined to color the vertices such that no two adjacent vertices are assigned the same color.) Vertices in G that are assigned the same color represent scan chains that can be fed by the same ATE channel via a fan-out structure. This procedure is illustrated using Figures 2.

In the above example, the number of scan chains $m = 8$, and the scan chain length $l = 4$. A conflict graph consisting 8 vertices is first determined. The coloring solution for the conflict graph is also shown; three colors are used to color three sets of vertices: {2, 4, 5, 7}, {3, 6, 8} and {1}. Consequently, three ATE channels are needed to feed the test data for the 8 scan chains, and the fan-out structure is shown in the figure. Since the graph coloring problem is known to be NP-complete, we use the simple heuristic described in [13].

While a significant amount of space compression can be achieved by exploiting the compatibilities between the scan chains and using a fan-out structure with c' inputs to drive m internal scan chains, we can achieve additional space compression by investigating the logic dependencies between the c' data streams from the ATE. In this way, the c' channels from the ATE can be reduced to $c < c'$ channels. The following discussion generalizes the design technique introduced in [8] for BIST of combinational logic circuits. Let $b_{k,i,j}$ denote the jth data bit that is shifted to the ith input of the fan-out structure for the kth test pattern. The ith input is logically dependent on a set of inputs $\{u_1, u_2, \ldots, u_s\}$ if there exists an s-input logic function F such that for any k $(1 \leq k \leq p)$ and j $(1 \leq j \leq l)$,



(a) Test cubes (b) Conflict graph

(c) Compressed test data based on scan chain compatibility (d) Fan-out structure

Fig. 2. Width compression based on scan chain compatibilities.

$b_{k,i,j} = F(b_{k,u_1,j}, b_{k,u_2,j}, \ldots, b_{k,u_s,j})$. The don't-care bits among $b_{k,i,j}, b_{k,u_1,j}, b_{k,u_2,j}, \ldots, b_{k,u_s,j}$ can be set appropriately to satisfy the above equation. If the ith input is logically dependent on an input set $\{u_1, u_2, \ldots, u_s\}$, then a s-to-1 combinational circuit can be used to generate the test data for the ith input from the inputs $\{u_1, u_2, \ldots, u_s\}$ and the number c of ATE channels needed to drive the fan-out structure is reduced by one. For c inputs, the complexity for checking all possible logic dependencies with a specific s-input logic function is $O(plc\binom{c}{s})$. Since there are a total of 2^{2^s} different s-input logic functions, the complexity for checking all the possible logic dependencies with any s-input logic function is $O(plc\binom{c}{s}2^{2^s})$. While in many cases, the logic dependency check can terminate before going though all the pl bits, the computational complexity is still prohibitive for large s. In this paper, we only examine the logic dependencies for the six 2-input functions: AND, NAND, OR, NOR, XOR, and XNOR. This restriction also reduces the hardware overhead required to generate the data for an input channel from the data for other input channels.

Consider two v-bit data streams $X = x_1, x_2, \ldots, x_v$ and $Y = y_1, y_2, \ldots, y_v$, respectively, where $x_i, y_i \in \{0, 1, x\}, 1 \leq i \leq v$. Suppose that the two data streams are combined using a two-input gate, denoted by $\bigotimes$, to generate a third data stream $Z = z_1, z_2, \ldots, z_v$, where $z_i = x_i \bigotimes y_i, 1 \leq i \leq v$. Table I shows the necessary and sufficient conditions under which gate type $\bigotimes$ can be used to generate z_i from x_i and y_i, $1 \leq i \leq v$, for three different gate types. Note that Z can be generated from X and Y using gate type $\bigotimes$ if and only if z_i can be generated from x_i and y_i for all $i, 1 \leq i \leq v$.

Let $\mathcal{G}$ be the set of available two-input gates, i.e., $\mathcal{G}$ = {AND, NAND, OR, NOR, XOR, XNOR}, and $\mathcal{M}$ be the compressed test data in matrix form (with c' columns and pl rows) after exploiting the compatibilities between scan chains. The procedure used in our experiments to find the logic dependencies between scan chains is described in Figure 3. The output of the procedure is a set $\mathcal{Q}$ in which each element is a 4-tuple (o, g, i_1, i_2). The 4-tuple (o, g, i_1, i_2) indicates that the data for the input channel o can be generated from the data for the input channels i_1 and i_2 using a two-input gate type g. Some of the don't-care bits in the compressed test data matrix $\mathcal{M}$ are set to appropriate values in accordance with logic dependencies identified by the procedure. The function *IsIn*$(\mathcal{Q}, i)$ is used to find out if the input channel i is the input of any of the 4-tuples in $\mathcal{Q}$. Similarly, the function *IsOut*$(\mathcal{Q}, i)$ is used to find out if the input channel i is the output of any of the 4-tuples in $\mathcal{Q}$. Applying the procedure to the compressed test data in Figure 2(c), we can obtain the output $\mathcal{Q} = \{(1, \text{XOR}, 2, 3)\}$. The resulting compressed test data and gated fan-out structure are shown in Figure 4.

TABLE I. NECESSARY AND SUFFICIENT CONDITIONS TO GENERATE z_i FROM x_i AND y_i USING GATE TYPE $\bigotimes$.

$\bigotimes$	z_i	x_i	y_i
AND	0	0	X[1]
		X	0
		X	X (map to 0)
	1	1	1
		1	X (map to 1)
		X (map to 1)	1
		X (map to 1)	X (map to 1)
	X	X	X
OR	0	0	0
		0	X (map to 0)
		X (map to 0)	0
		X (map to 0)	X (map to 0)
	1	1	X
		X	1
		X	X (map to 1)
	X	X	X

$\bigotimes$	z_i	x_i	y_i
XOR	0	0	0
		0	X (map to 0)
		X (map to 0)	0
		X (map to 0)	X (map to 0)
	1	0	1
		0	X (map to 1)
		X (map to 1)	0
		X (map to 1)	1
		1	0
		1	X (map to 0)
		X (map to 0)	1
		X (map to 0)	X (map to 1)
	X	X	X

[1] The X can be arbitrarily mapped to 1 or 0 unless indicated otherwise.

Procedure *Logic_dependency* ($\mathcal{G}$, $\mathcal{M}$, $\mathcal{Q}$)

```
1 /* Find the logic dependencies between scan chains. */
2 c = Width (M);
3 for (o = 1 to c)
4     if ( IsIn (o, Q) ) continue; end if
5     for (i1 = 1 to c)
6         if ( IsOut (i1, Q) or i1 = o ) continue; end if
7         for (i2 = 1 to c)
8             if ( IsOut (i2, Q) or i2 = o or i2 = i1 ) continue; end if
9             G1 = G;
10            /* Check the logic dependency of input o on inputs i1 and i2. */
11            for each (g in G1)
12                if ( Not_valid (M, o, g, i1, i2) )
13                    G1 = G1 - g;
14                end if
15            end for each
16            if ( G1 ≠ Ø)
17                g = First_element (G1);
18                q = (o, g, i1, i2);
19                Insert (Q, q);
20                /* Set the unspecified bits to appropriate values to
21                   validate the logic dependency. */
22                Validate (M, o, g, i1, i2);
23            end if
24        end for  end for
25 end for
```

Fig. 3. Pseudocode determine the logic dependencies between scan chains.

As a result of the width compression procedure, constraints are introduced for the test patterns, i.e., the scan chains fed by the same ATE channel are filled with the same test data for any test pattern. However, as long as all the irredundant faults can be detected by the original test set that is used to obtain the fan-out structure, they can also be detected by the test patterns arising from the constraints that are introduced by the fan-out structure. It is easy to show that if a fault is detectable in the original circuit with m external channels, it is also detectable in the circuit with c external channels obtained after width compression. In contrast to [1], no attempt is made here to make c any smaller by relaxing care bits or by interleaving width compression with test generation.

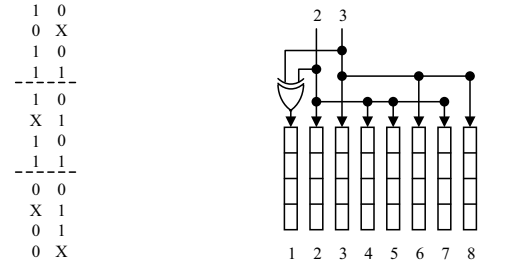
B. Height Compression

In this subsection, we describe the procedure for height compression. Assuming that structural information about the IP core is not available or test reuse if preferred (test generation is deemed to be too expensive), we first merge the test data of the compatible scan chains for the original test set after the width compression procedure. Next, static compression is carried out with the width-compressed test data to reduce the number of test patterns. The fault coverage achieved with the fan-out structure and with the test patterns obtained after static compaction is the same as the fault coverage obtained with the original set of p test cubes and m external channels. The static compaction algorithm is similar to the width compression of the previous subsection in that it also relies on graph coloring. The difference lies in that fact that each vertex corresponds to a test pattern instead of a scan chain. This approach does not require any test generation or fault simulation, hence it is especially attractive for hard cores in SOCs.

III. Impact of the number of ATE channels

In conventional testing using multiple scan chains, the number of ATE channels is equal to the number of scan chains. An increase in the number of ATE channels leads to a reduction in testing time because more data can be shifted into the scan chains in a single scan clock cycle. However, in the fan-out approach, the number of internal scan chains m can be very large, even for a small number of ATE channels. In practice, it may not be possible to increase m beyond a certain limit due to routing constraints. In such situations, an increase in the number of ATE channels only provides the benefit of more freedom for height compression; the number of data bits that are shifted into the scan chains in a single cycle is not increased. Thus there is only limited benefit in increasing the number of ATE channels any further. This issue is highlighted in the experimental results obtained for two production circuits from IBM, namely CKT1 and CKT2.

CKT1 consists of 51082 gates and 10788 latches and its test set provides 99.80% fault coverage. CKT2 consists of 94340 gates and 17915 latches and its test set provides 99.76% fault coverage. The detailed experimental results for these two circuits are listed in Section IV. Here we only show the impact of the number of ATE channels on the test application time. We limit the number of scan chains to 600 for CKT1 and 1000 for CKT2, respectively. For CKT1, we vary the number of scan chains from 128 to 600, and apply the width compression procedure to determine the minimum number of ATE channels that are needed to drive the scan chains. For 600 scan chains, we find that the minimum number of ATE channels needed is 32. Next, we fix the number of scan chains at 600 while continuing to increase the number of ATE channels. The relationship of the number of internal scan chains to the number of ATE channels is shown in Figure 5(a). For each pair of (c, m) values shown in Figure 5(a) for CKT1, we perform static compaction to reduce the number of test patterns, and derive the test application time. Figure 5(b) shows the testing time versus the number of ATE channels. After the number of ATE channels reaches 32, the increase in the number of ATE channels does not lead to significant reduction of testing time for the proposed method. In Figure 5(c), we also show the testing time for the ATPG-compacted test set versus the number of ATE channels. The figure shows that even though the testing time for the latter case continues to decrease with an increase in c, it is only for very large values of c that the testing time for this case becomes comparable to that for the proposed methods. Similar results are obtained for CKT2, as shown in Figure 5.



(a) Compressed test data based on the compatibilities and logic dependencies (b) Gated fan-out structure

Fig. 4. Width compression based on scan chain compatibilities and logic dependencies.

IV. Experimental results

In this section, we first apply the proposed approach to the seven largest ISCAS-89 benchmark circuits. For each circuit, we start with a test set obtained from the Mintest ATPG program [6] without dynamic compaction. We first use the width compression technique described in Section II-A to obtain the minimum number of ATE channels that are needed to drive the scan chains, and we determine the corresponding fan-out structure. Since these test sets provide 100% fault coverage for the irredundant single stuck-at faults, the width compression procedure guarantees that all the irredundant faults can be detected by test data transferred through the fan-out structure. After width compression, we apply height compression techniques to reduce the number of test patterns. Table II shows the results for height compression based on static compaction. Static compaction took less then 70 seconds for each of the ISCAS-89 circuits.

In Table II, we also compare the test data volume and testing

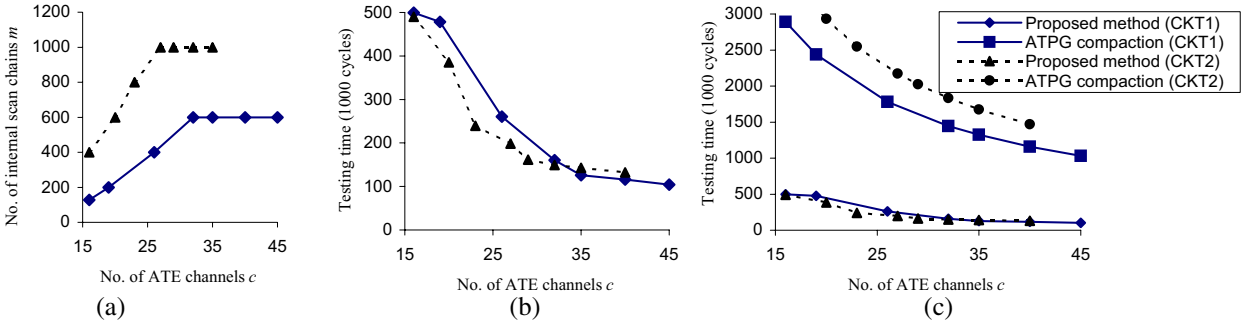


Fig. 5. The impact of the number of ATE channels on testing time for CKT1 and CKT2.

TABLE II. COMPRESSION RESULTS FOR THE ISCAS-89 BENCHMARK CIRCUITS.

Circuit	No. of test cubes	No. of scan chains	No. of ATE channels	No. of two-input gates required	No. of patterns for T_E	$\|T_E\|$ (bits)	TAT_E (cycles)	$\|T_{\rm M}\|$ (bits)	$TAT_{\rm M}$ (cycles)	$\triangle V$ (%) (Mintest)	$\triangle TAT$ (%) (Mintest)	$\|T_{\rm A}\|$ (bits)	$TAT_{\rm A}$ (cycles)	$\triangle V$ (%) (Atalanta)	$\triangle TAT$ (%) (Atalanta)
s5378	1458	64	9	4	395	14220	1975	23754	2775	40.14	28.83	52858	6175	73.10	68.02
s9234	1928	64	16	7	471	30144	2355	39273	2703	23.24	12.87	88426	6086	65.91	61.30
s13207	3237	200	11	4	477	20988	2385	165200	15340	87.30	84.45	323400	30030	93.51	92.06
s15850	3920	200	15	2	422	25320	2110	76986	5292	67.11	60.13	265785	18270	90.47	88.45
	3920	128	12	2	419	25140	2514	76986	6552	67.34	61.63	265785	22620	90.54	88.89
s35932	10810	200	3	1	147	3969	1470	28208	9424	85.93	84.40	111069	37107	96.43	96.04
s38417	10771	200	14	0	1129	142254	11290	164736	11880	13.65	4.97	1560832	112560	90.89	89.97
	10771	48	5	0	487	85225	17532	164736	33066	48.27	46.98	1560832	313292	94.54	94.40
s38584	13468	200	14	2	510	57120	4590	199104	14416	71.31	68.16	900360	65190	93.66	92.96

time to the case when a set of ATPG-compacted test patterns $T_{\rm M}$ ($T_{\rm A}$) obtained using Mintest (Atalanta) is applied through the same number of external ATE channels. The parameters $\triangle V$ and $\triangle TAT$ refer to the reduction in test data volume and and test application time, respectively, over ATPG compaction. While we achieve lower test data volume and testing time in all cases with negligible hardware overhead, the reduction is especially striking for s13207 and s35932.

In Table III, we compare our results to some recently-published test data compression methods that do not utilize structural information of the circuit under test. We report the smallest test data volume obtained over several values of m. Compared to FDR and 9C (EFDR and VIHC), the test data volume of our proposed approach is higher in three (four) out of the seven cases. The proposed method outperforms LZ77 in five out seven cases. We are unable to directly compare our results with [19] because of the different test sets used in [19]. Note however that coding methods such as [3, 4, 5, 19] are targeted towards single scan chains; for multiple scan chains, these methods either require considerable hardware overhead (decoder per scan chain) or higher testing time. In 9C coding [16], an m-bit shifter is required for the decompression architecture if it drives m scan chains. In [18], the boundary scan cells need to be modified and used for decompression. Unlike FDR [3], EFDR [4], VIHC [5] and 9C coding [16], the proposed method does not require synchronization with the ATE.

In order to demonstrate the effectiveness of the proposed two-dimensional test data compression techniques for industrial circuits, we present results for four production circuits from IBM. CKT1 and CKT 2 have been described in Section III. CKT3 consists of 3.6 million gates and 726000 latches, and

TABLE III. COMPARISON OF COMPRESSION RESULTS

Circuit	Size of T_E (bits)					
	Proposed approach	FDR [3]	EFDR [4]	VIHC [5]	9C [16]	LZ77 [18]
s5378	14220	12346	11419	11516	11487	12769
s9234	30144	22152	21250	17736	19279	19429
s13207	20988	30880	29992	27737	29224	N/A
s15850	25140	26000	24643	30271	25883	37980
s35932	3969	22744	5554	9458	N/A	22496
s38417	85225	93466	64962	74938	64857	N/A
s38584	57120	77812	73853	85674	68631	N/A

CKT4 consists 1.2 million gates and 32200 latches. We were only provided with partial test sets for these circuits, thereby we are unable to report fault coverage numbers. We carry out static compaction on the test cubes after width compression for CKT1 and CKT2. For CKT3 and CKT4, the number of test patterns is small, hence we do not perform height compression for these test sets. While XOR is the predominant gate type for the smaller circuits, NAND and OR gates were also used in the decompression logic for CKT3 and CKT4. Table IV shows the test data volume and the testing time for our approach, and compares these to that obtained with the maximum compaction provided by the same ATPG tool. Compared to the ATPG-compacted test set T_C, the proposed method achieves up to 91% reduction in test data volume. With the same number of ATE channels, the test application time TAT_E for the proposed method is over 90% less than that for the compacted test set. On a laptop with a 1.8 GHz CPU and 512 MB memory, the computation time to obtain a complete set of results for CKT2 in Table IV is approximately 3 hours. The CPU times are less for the other circuits.

Table V compares the proposed approach with the FDR code [3] in terms of the reduction in test data volume and test

TABLE IV. COMPRESSION RESULTS FOR THE INDUSTRIAL CIRCUITS.

Circuit	No. of test cubes	No. of scan chains	Scan chain length	No. of ATE channels	No. of two-input gates required	No. of test patterns for T_E	Size of T_E (bits)	TAT_E (cycles)	No. of test vectors for T_C^*	Size of T_C (bits)	TAT_C (cycles)	Compression percentage relative to T_C	Percentage reduction in TAT
CKT1	17176	128	96	16	8	4150	6374400	402550	3768	46180608	2890056	86.20	86.07
		200	62	19	12	7593	8944554	478359	3768	46180608	2437896	80.63	80.38
		400	31	26	17	8140	6560840	260480	3768	46180608	1782264	85.79	85.38
		600	21	32	15	7301	4906272	160622	3768	46180608	1446912	89.38	88.90
CKT2	43079	400	56	16	5	8600	7705600	490200	2636	58561376	3664040	86.84	86.62
		600	38	20	4	9876	7505760	385164	2636	58561376	2931232	87.18	86.86
		800	28	23	2	8252	5314288	239308	2636	58561376	2549012	90.93	90.61
		1000	23	27	2	8265	5132565	198360	2636	58561376	2172064	91.24	90.87
CKT3	32	600	605	50	41	32	968000	19392	32	11613472	232320	91.66	91.65
		800	454	58	34	32	842624	14560	32	11613472	200288	92.74	92.73
		1000	363	68	38	32	789888	11648	32	11613472	170848	93.20	93.18
CKT4	4	600	1719	73	43	4	501948	6880	4	4124288	56504	87.83	87.82
		800	1289	86	51	4	443416	5160	4	4124288	47964	89.25	89.24
		1000	1032	93	48	4	383904	4132	4	4124288	44352	90.69	90.68

*For CKT3 and CKT4, no compacted test sets are available. The data listed in this table corresponds to the original test set T_D.

TABLE V. COMPARISON WITH FDR CODE [3].

		Proposed approach		FDR [3]			
	No. of ATE	Size of	TAT_E (ATE clock	Size of		TAT_{FDR} (ATE clock cycles)	
Circuit	channels	T_E (bits)	cycles)	T_E (bits)	α	Upper bound	Lower bound
CKT1	32	4906272	160622	3047472	4	1739836	1692219
CKT2	27	5132565	198360	5425352	4	9062449	8961979
CKT3	68	789888	11648	243998	4	46285	44491
CKT4	93	383904	4132	201490	4	13253	12170

application time. The comparison is carried out for the four industrial circuits. Using the method described in [2], we calculate upper and lower bounds on the testing time for the FDR code. In [3], the scan clock can run at a higher frequency $f_{scan} = \alpha f_{ATE}$ than the ATE clock frequency f_{ATE}, where $\alpha > 1$ and is normally set to the power of 2. We calculate the upper and lower bounds for $\alpha = 4$. We also assume that the compressed data is transferred from the ATE using the same number of channels for the two methods. The results show that although the test data volume obtained using the FDR code is slightly in three out of four cases, the test application time of FDR code is at least at an order of magnitude higher than that for the proposed approach. This difference is even higher for the practical case of $\alpha = 1$.

V. CONCLUSION

We have presented a two-step compression technique that can reduce test data volume and test application time with negligible hardware overhead for on-chip decompression. The hardware overhead is largely limited to the fan-out between the tester-driven external scan pins and the internal scan chains. Experimental results for the ISCAS-89 circuits demonstrate that the proposed technique provides significant improvement over other recent methods. Results for four industrial circuits show that compared to compacted test sets, up to 10X reduction in scan test data volume and testing time can be obtained.

REFERENCES

[1] I. Bayraktaroglu and A. Orailoglu, "Decompression hardware determination for test volume and time reduction through unified test pattern compaction and compression," *Proc. VLSI Test Symp.*, pp. 113–118, 2003.

[2] A. Chandra and K. Chakrabarty, "Test resource partitioning and reduced pin-count testing based on test data compression", *Proc. DATE Conf.*, pp. 598–603, 2002.

[3] A. Chandra and K. Chakrabarty, "Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes," *IEEE Trans. Computers*, vol. 52, pp. 1076-1088, August 2003.

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

[5] 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. DATE Conf.*, pp. 604–611, 2002.

[6] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits", *Proc. Int. Conf. CAD*, pp. 283-289, 1998.

[7] I. Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores," *Proc. IEEE Int. Symp. Fault Tolerant Computing*, pp. 260–267, 1999.

[8] I. Hamzaoglu and J. H. Patel, " Reducing test application time for built-in-self-test test pattern generators", Proc. IEEE VLSI Test Symposium, pp. 369–375, 2000.

[9] A. Jas, C. V. Krishna and N. A. Touba, "Hybrid BIST based on weighted pseudo-random testing: a new test resource partitioning scheme," *Proc. VLSI Test Symp.*, pp. 2–8, 2001.

[10] A. Khoche and J. Rivoir, "I/O bandwidth bottleneck for test: is it real?" *Test Resource Partitioning Workshop*, 2002.

[11] B. Koenemann et al., "A SmartBIST variant with guaranteed encoding," *Proc. Asian Test Symp.*, pp. 325–330, 2001.

[12] K.-J. Lee, J.-J. Chen, C.-H. Huang, "Using a single input to support multiple scan chains," *Proc. Int. Conf. CAD*, pp. 74–78, 1998.

[13] L. Li and K. Chakrabarty, "Test data compression using dictionaries with fixed-length indices," *Proc. VLSI Test Symp.*, pp. 219–224, 2003.

[14] J. Rajski et al., "Embedded deterministic test for low-cost manufacturing test," *Proc. Int. Test Conf.*, pp. 301–310, 2002.

[15] H. Tang, S. M. Reddy and I. Pomeranz, "On reducing test data volume and test application time for multiple scan chain designs," *Proc. Intl. Test Conf.*, pp. 1079–1088, 2003.

[16] M. Tehranipour, M. Nourani and K. Chakrabarty, "Nine-coded compression technique with application to reduced pin-count testing and flexible on-chip decompression," *Proc. DATE Conf.*, pp. 1284–1289, 2004.

[17] H. Vranken et al., "ATPG padding and ATE vector repeat per port for reducing test data volume," *Proc. Intl. Test Conf.*, pp. 1069–1076, 2003.

[18] F. G. Wolff and C. Papachristou, "Multiscan-based test compression and hardware decompression using LZ77", *Proc. Int. Test Conf.*, pp. 331–339, 2002.

[19] A. Wurtenberger, C. S. Tautermann and S. Hellebrand, "A hybrid coding strategy for optimized test data compression", *Proc. Int. Test Conf.*, pp. 451–459, 2003.