DATA COMPRESSION FOR MULTIPLE SCAN CHAINS USING DICTIONARIES WITH CORRECTIONS

Armin Würtenberger, Christofer S. Tautermann, Sybille Hellebrand
University of Innsbruck, Austria

Abstract

Reducing test application time and test data volume are major challenges in SoC design. In the case of IP cores, where no structural information is available, a common strategy is to compress the test data T_D provided by the core vendor into an encoded format T_E. Only the smaller set T_E is stored on the ATE, and during test the original test data T_D are regenerated by an on-chip decompressor. However, most of the encoding schemes suffer from two major drawbacks: Firstly, the irregularity of the encoded test data requires a complex test control including a handshake between the ATE and the system under test. Secondly, compression and decompression is very efficient for circuits with a single scan chain, however the extension to multiple scan chains requires either a separate decompressor for each chain or a serialization of the test data. So far, only a few approaches have been proposed trying to overcome these problems. Instead of dealing with the test vectors these approaches work with the slices to be fed into the scan chains, but they still allow a considerable degree of irregularity in the test application process. In this paper we propose a new dictionary based compression scheme which allows a fully regular test application while keeping the storage requirements low. Due to the regularity of the scheme the advantages of a multiple-scan architecture are preserved, and very low test times can be achieved.

1 Introduction

Test application time and test data volume are major cost factors in SoC design. To reduce the storage requirements for the ATE and to cope with a limited bandwidth between the ATE and the system under test, numerous approaches have been developed either integrating the complete test infrastructure on chip or partitioning the test resources between the ATE and the chip. Built-in self-test (BIST) with pseudo-random patterns is now a widely accepted solution [1]. For circuits with random-pattern resistant faults more advanced strategies have been proposed including deterministic or mixed-mode BIST, reduced pin count testing and test resource partitioning (TRP) for deterministic test [10, 13, 14, 16, 17, 24, 26, 29]. In the meantime even commercial support is available [3, 15, 19]. However, for an efficient implementation, most of the BIST approaches and also many strategies for TRP need to be combined with fault simulation or interleaved with ATPG. In the case of IP cores, where no structural information is available, a common strategy is to compress the test data T_D provided by the core vendor into an encoded format T_E and store this smaller set on the ATE. During test T_E is transmitted to the chip where the original test data T_D are regenerated by an on-chip decompressor. Recent coding strategies for test data compression are based on statistical coding, run-length coding, dictionary-based coding or hybrid schemes [6, 7, 9, 11, 12, 22, 25, 27, 28]. Most of these schemes suffer from two major drawbacks:

- The irregularity of the encoded test data (variable length blocks, etc.) requires a complex test control including a handshake between the ATE and the system under test.
- The decompression hardware is rather simple for circuits with a single scan chain, however the extension to multiple scan chains requires either a separate decompressor for each chain or a serialization of the test data resulting in increased test times.

So far, only a few approaches have been proposed trying to overcome the problems mentioned above [2, 18, 21, 23, 20]. Instead of dealing with the test vectors these approaches work with the slices to be fed into the scan chains as shown in Figure 1.

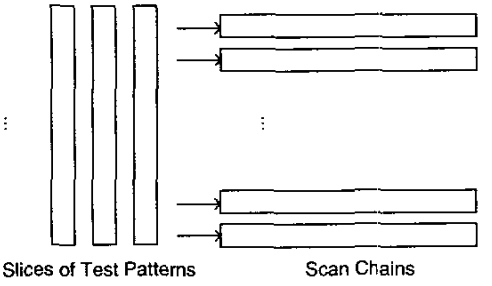

Figure 1: Decomposition of test patterns into slices.

ITC INTERNATIONAL TEST CONFERENCE

In [2, 20, 23] the test data volume to be stored is reduced by identifying compatible scan chains which can be fed with identical information or with linear combinations of the test data for other scan chains (similar to classical techniques for test width compression or for the generation of pseudo-exhaustive test sets [8]). This way, only k-bit slices have to be stored for n scan chains, $k < n$. To minimize the information to be stored for each slice Reda and Orailoglu propose to store the differences between slices (bit positions with conflicting specified bits) instead of the slices [21]. The problem with this approach is the varying number of conflicting bits and the resulting irregularity in the decompression process. In [18] a dictionary-based compression scheme is presented where the slices are encoded as fixed-length indices of a dictionary. However, the experimental results show, that in general it is not possible to encode all the slices of a test set as entries of a reasonably sized dictionary. Therefore the authors limit the size of the dictionary, and slices not in the dictionary are not encoded. An extra signal is used to distinguish between dictionary entries and explicit information.

In this paper we propose a dictionary based compression scheme which allows a fully regular test application and preserves the advantages of a multiple-scan architecture. The basic principle of the proposed compression strategy is described in Section 2. Subsequently in Section 3 a heuristic procedure for constructing the dictionary and encoding the test data is presented. Finally, in Section 4 experimental data are discussed. It will be shown that the compression ratio improves with the number of scan chains, and that in the best configurations between 7 % and 20 % of the original test data volume needs to be stored on the ATE only. Due to the regularity of the scheme one tester cycle is actually sufficient to load all scan chains simultaneously with a new slice, which results in very low test times.

2 Proposed Compression Strategy

As pointed out in the introduction, it cannot be expected that all the slices in a test can be encoded as entries of a reasonably small dictionary. To circumvent this problem without disturbing the regularity of the scheme, we allow a fixed number of corrections before the slices are delivered to the scan chains. Our experimental data in Section 4 will show that allowing only one correction per slice already guarantees very small dictionaries. As shown in Figure 2 a slice is then encoded as a dictionary entry (stored in a RAM) and a bit position indicating which bit of the dictionary entry must be flipped to get the desired slice. If a slice can be taken from the dictionary without correction, this is indicated as bit position 0. Therefore, one slice is encoded with $log_2 d + log_2(n+1)$ bits, where d denotes the number of dictionary entries and n the number of scan chains.

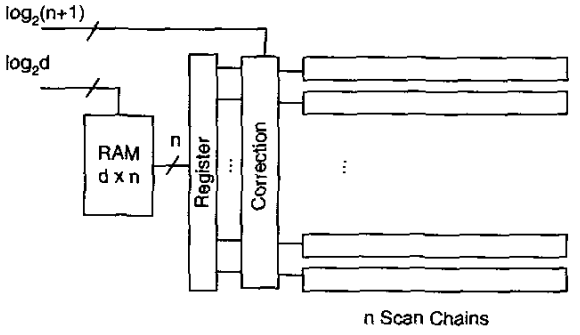

Figure 2: Basic architecture for dictionary-based encoding with correction.

Remaining don't cares can be set, such that the number of independent input signals, k, needed to feed the scan chains are minimized similarly as in [2, 8, 20, 23]. In our approach we consider scan chains as compatible, if they can be fed with identical data or with the complementary data (bitwise inversion). Further optimization is possible, if for slices which do not need correction, the extra bits are used to indicate a correction yielding more compatible scan chains. The resulting architecture is shown in Figure 3.

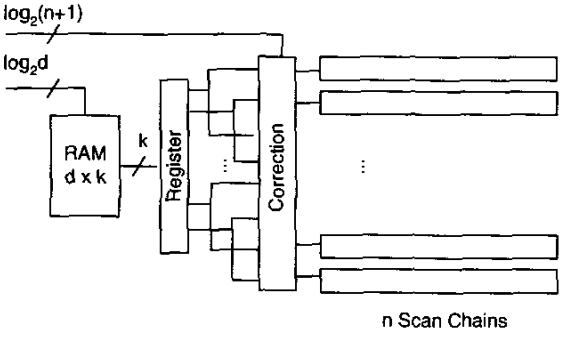

Figure 3: Architecture with a reduced number of independent scan inputs.

The complete encoding and decoding procedure is illustrated for the small example of Figure 4. For a circuit with 12 scan elements and 4 scan inputs the original 12-bit pattern is split into 4 patterns of length 3 for the individual scan chains. Accordingly the 3 depicted 4-bit slices have to be applied to the scan inputs during test. To construct a dictionary for the basic architecture of Figure 2, the slices are analyzed. If only specified bits are considered, then the first and the third slice differ only in bit position 3 and can be represented by the same dictionary entry (0,1,1,X). The second slice differs from this dictionary entry in 2 bit

positions and must be represented by an extra dictionary entry (1,X,0,0). As a result the slices are encoded as pairs (0,0), (1,0), (0,3), where the first component indicates the dictionary entry and the second component indicates the bit position to be flipped ('0' means that no correction is necessary).

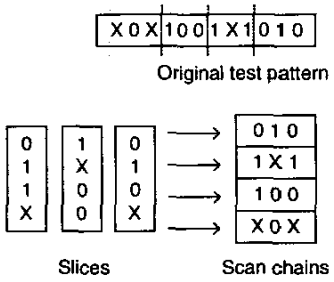

Figure 4: Example with 4 scan chains and 3 slices.

To implement the dictionary the don't cares can be fixed to arbitrary values. Figure 5 shows the resulting architecture.

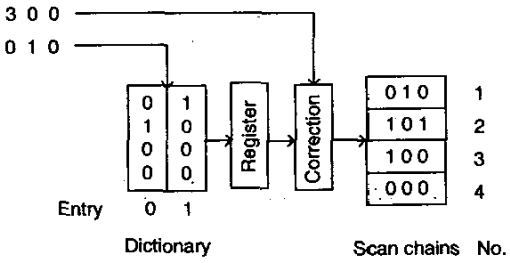

Figure 5: Basic architecture for the example of Figure 4.

If the don't cares in the dictionary are both mapped to 0 as in Figure 5, then the first two rows of the dictionary are complementary and the last two rows are identical. This can be exploited to reduce the dictionary as illustrated in Figure 6.

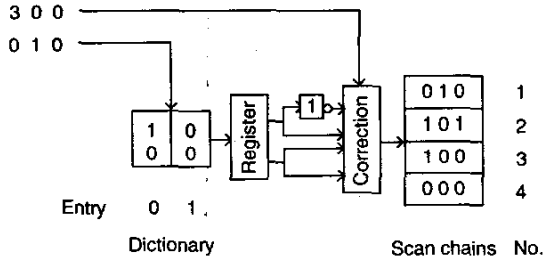

Figure 6: Example of Figure 4 with a reduced number of independent scan chains.

With the proposed architecture a throughput of one slice per tester cycle can be achieved using $log_2 d$ + $log_2(n+1)$ tester channels. Because of the regularity of the scheme, the same throughput is also possible using even less tester channels and pipelining the decompression inside the chip. In this case some extra registers and control have to be added.

As shown by our experimental data, in general, a large subset of bit positions never needs correction, and the correction circuit has to distribute possible corrections to $m < n$ bit positions only. The number of tester channels can be reduced to $log_2 d + log_2(m+1)$ this way, but then the correction circuit must be specifically tuned to the test set. Thus, there is a trade-off between an improved compression and a general, test set independent decompression hardware.

3 Encoding Procedure

The compression ratio is determined by several parameters. Depending on the implementation of the correction circuit as a test set independent or a tailored circuit the number of scan chains n or the number of correctable bit positions m, respectively, is one of these parameters. Furthermore, the number of dictionary entries d, and the size of the RAM $d \cdot k$, where k is the number of bits in the reduced slices, directly influence the compression ratio. If s slices have to applied to the scan inputs, then the compression ratio is determined by the formula

$$\frac{size(T_E(n)) + size(RAM)}{size(T_D)} = \frac{s \cdot (\lceil \log_2 d \rceil + \lceil \log_2(n+1) \rceil) + d \cdot k}{s \cdot n}$$

or

$$\frac{size(T_E(m)) + size(RAM)}{size(T_D)} = \frac{s \cdot (\lceil \log_2 d \rceil + \lceil \log_2(m+1) \rceil) + d \cdot k}{s \cdot n},$$

respectively. To optimize the compression ratio, the parameters d, m, and k must be minimized. The problem of finding the best dictionary entries and the corresponding information for correction can be viewed as a clique partitioning problem in a similar way as in [18]. But whereas in [18] only slices with hamming distance 0 are compatible, in our approach also slices with hamming distance 1 must be considered (the hamming distance between two slices only takes into account the bit positions which are specified in both slices). From each clique one base vector is selected as a dictionary entry. Because of the NP completeness of the clique covering problem finding an optimal solution might require exponential time (the solution is also dependent on the filling of don't cares). Therefore the simple heuristic summarized in Figure 7 has been developed as a first solution.

Initially the dictionary D is empty, and the first slice s^* in the set of all slices S is selected as first entry of D and

removed from S. Then the remaining slices are analyzed, and step by step slices with hamming distance 0 or 1 to s^* are removed from S. If necessary, don't cares are fixed in s*. For slices with hamming distance 1, the conflicting bit position is remembered as second component of the encoding. This process is repeated for the remaining slices until the set S is empty.

```
D = {}; // dictionary D is initially empty
S = {all slices};
while S != {}
{
   select the first slice s* ∈ S;
   D = D ∪{s*}; S = S \ {s*};
   while possible
   {
        select s ∈ S with d(s, s*) = 0
        or d(s, s*) = 1;
        // d: hamming distance
        fix all don't cares in s* where
        corresponding bits in s are specified;
        if d(s, s*) = 1 store the bit
        position where s must be corrected;
        S = S \ {s};
   }
}
```

Figure 7: Heuristic Encoding Procedure.

After the dictionary and the correction information has been determined using the heuristic of Figure 7, the size of the dictionary entries is minimized by appropriately specifying the remaining don't cares and exploiting unused correction bits.

4 Experimental Results

The encoding procedure of Section 3 has been applied to the ISCAS'85 (considering the circuit inputs as scan elements) and the combinational parts of the ISCAS'89 circuits [4, 5]. Here only results for the larger random pattern resistant circuits are reported. Table 1 shows the results for a test set independent correction circuit and the same test sets as used in [28]. The number of pseudo-primary inputs, the number of test vectors, and the size of the original test set are shown in columns 2 to 4. For each circuit different numbers of scan chains have been investigated. Depending on the number of scan elements the maximum number of scan chains has been set to 64 or 128, respectively. The number of scan chains, the corresponding lengths of the chains, and the resulting number of slices are listed in columns 5 to 7. The following 3 columns show the number of dictionary entries d, the size of the reduced slices k, and the resulting RAM size in bits ($k \cdot d$). Please note that in some cases, there are only 2 dictionary entries of size 1. In thes cases the information can be sent directly instead of specifying a dictionary index of the same width. The RAM size is therefore listed as zero. Finally, the last 3 columns show the size of the compressed test set $T_E(n)$, the number of bits required to store both $T_E(n)$ and the dictionary (RAM), and the compression ratio which is given by the ratio of the number of bits in T_D and the number of bits in $T_E(n) + RAM$. It can be observed that the compression ratio depends on the number of scan chains and improves with a larger number of shorter scan chains. In the best cases a compression down to 8 % - 26 % of the original test data volume can be achieved.

As explained before, the test data storage can be further reduced, if a specifically tailored correction circuit for $m < n$ bit positions is implemented. Table 2 presents results for this approach. For convenience the number of pseudo-primary inputs, the number of test vectors, the size of the original test set, and the number of scan chains are again shown in columns 2 to 5. The next two columns repeat the results from Table 1 for a test set independent solution, and the following four columns show the experimental data for a tailored correction circuit. In general, the number of bit positions m which actually need correction is considerably smaller than the number of scan chains. Consequently, in most of the cases a further improvement of the compression ratio can be observed. In the best cases the storage requirements are now between 7 % and 24 % of the original test data volume.

To analyze the impact of the number of scan chains on the quality of the results in more detail, the experiments were repeated for all possible configurations from 4 to 128 scan chains for the larger examples, and from 4 to 64 scan chains for the smaller examples. The diagram in Figure 8 summarizes the results for some of the circuits and clearly confirms the trends of Tables 1 and 2. If the number of scan chains is small, then only little compression or no compression at all is possible. This is due to the fact that the number of bits to represent a slice logarithmically depends on the number of scan chains n and the number of dictionary entries d, and for small n the difference between n and $log_2(n+1) + log_2 d$ is small. However, with an increasing number of scan chains the size of the encoded slices relative to the size of the original slices decreases rapidly.

Circuit	PPI	Vectors	T_D (bits)	Scan chains (n)	Lengths of scan chains	Slices (s)	d	k	RAM (bits)	$T_E(n)$ (bits)	$T_E(n)$ + RAM	$\frac{T_E(n)+RAM}{T_D}$
c2670	233	112	26096	16	15	1680	4	6	24	11760	11784	0,45
				32	8	896	7	9	63	8064	8127	0,31
				64	4	448	12	15	180	4928	5108	0,2
c7552	207	129	26703	16	13	1677	17	16	272	16770	17042	0,64
				32	7	903	18	27	486	9933	10419	0,39
				64	4	516	21	33	693	6192	6885	0,26
s5378	214	28	5992	16	14	392	2	1	0	2352	2352	0,39
				32	7	196	2	2	4	1372	1376	0,23
				64	4	112	5	5	25	1120	1145	0,19
s9234	247	296	73112	16	16	4736	7	9	63	37888	37951	0,52
				32	8	2368	10	19	190	23680	23870	0,33
				64	4	1184	19	36	684	14208	14892	0,2
s13207	700	315	220500	16	44	13860	2	1	0	83160	83160	0,38
				32	22	6930	2	1	0	48510	48510	0,22
				64	11	3465	3	3	9	31185	31194	0,14
				128	6	1890	4	5	20	18900	18920	0,09
s15850	611	268	163748	16	39	10452	2	2	4	62712	62716	0,38
				32	20	5360	3	3	9	42880	42889	0,26
				64	10	2680	6	8	48	26800	26848	0,16
				128	5	1340	8	20	160	14740	14900	0,09
s38417	1664	1323	2201472	16	104	137592	2	1	0	825552	825552	0,38
				32	52	68796	3	3	9	550368	550377	0,25
				64	26	34398	4	3	12	309582	309594	0,14
				128	13	17199	7	22	154	189189	189343	0,09
s38584	1464	312	456768	16	92	28704	2	1	0	172224	172224	0,38
				32	46	14352	2	2	4	100464	100468	0,22
				64	23	7176	3	3	9	64584	64593	0,14
				128	12	3744	4	3	12	37440	37452	0,08

Table 1: Experimental results for a test set independent correction circuit.

Circuit	PPI	Vectors	T_D (bits)	Scan chains (n)	$T_E(n)$ + RAM	$\frac{T_E(n) + RAM}{T_D}$	m	$T_E(m)$ (bits)	$T_E(m) + RAM$	$\frac{T_E(m) + RAM}{T_D}$
c2670	233	112	26096	16	11784	0,45	9	10080	10104	0,39
				32	8127	0,31	12	6272	6335	0,24
				64	5108	0,2	20	4032	4212	0,16
c7552	207	129	26703	16	17042	0,64	16	16770	17042	0,64
				32	10419	0,39	26	9030	9516	0,36
				64	6885	0,26	37	5676	6369	0,24
s5378	214	28	5992	16	2352	0,39	7	1568	1568	0,26
				32	1376	0,23	7	784	788	0,13
				64	1145	0,19	7	672	697	0,12
s9234	247	296	73112	16	37951	0,52	16	37888	37951	0,52
				32	23870	0,33	29	21312	21502	0,29
				64	14892	0,2	44	13024	13708	0,19
s13207	700	315	220500	16	83160	0,38	16	83160	83160	0,38
				32	48510	0,22	29	41580	41580	0,19
				64	31194	0,14	47	27720	27729	0,13
				128	18920	0,09	54	15120	15140	0,07
s15850	611	268	163748	16	62716	0,38	14	52260	52264	0,32
				32	42889	0,26	22	37520	37529	0,23
				64	26848	0,16	35	24120	24168	0,15
				128	14900	0,09	56	12060	12220	0,07
s38417	1664	1323	2201472	16	825552	0,38	16	825552	825552	0,38
				32	550377	0,25	32	550368	550377	0,25
				64	309594	0,14	59	275184	275196	0,13
				128	189343	0,09	99	171990	172144	0,08
s38584	1464	312	456768	16	172224	0,38	16	172224	172224	0,38
				32	100468	0,22	26	86112	86116	0,19
				64	64593	0,14	39	57408	57417	0,13
				128	37452	0,08	51	29952	29964	0,07

Table 2: Experimental results for a tailored correction circuit.

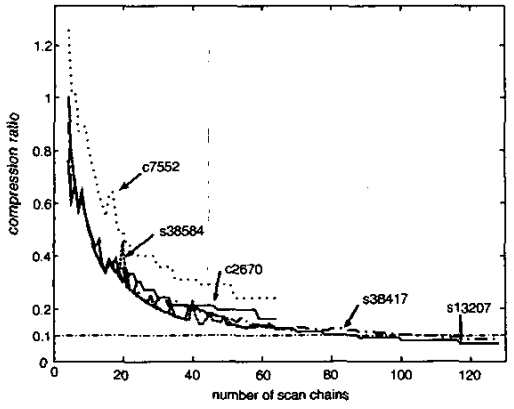

Figure 8: Compression as a function of the number of scan chains.

Comparing the proposed technique to other approaches has to cope with the fact that previous proposals either use different test sets or require a much more irregular test scheme. Since the experimental data in [18] show that the dictionary based scheme presented there favorably compares to other approaches published so far, we compare the proposed technique only to the best results reported in [18].

To ensure a fair comparison, we ran the same experiments as described above also for the test sets used in [18]. The results for a test set independent correction circuit are summarized in Table 3. It can be observed that, in general, the compression is not as effective as in our initial experiments. This is mainly due to the fact that the test sets contain less don't cares and thus provide less potential for compression.

Circuit	PPI	Vectors	T_D (bits)	Scan chains (n)	Lengths of scan chains	Slices (s)	d	k	RAM (bits)	$T_E(n)$ (bits)	$T_E(n)$ + RAM	$\frac{T_E(n)+RAM}{T_D}$
s5378	214	111	23754	16	13	14	96	16	1536	17094	18630	0.78
				32	7	7	142	31	4402	10101	14503	0.61
				64	3	4	116	54	6264	5328	11592	0.49
s9234	247	159	39273	16	15	16	89	16	1424	30528	31952	0.81
				32	8	8	134	31	4154	16536	20690	0.53
				64	4	4	164	61	10004	8904	18908	0.48
				128	2	2	155	110	17050	4452	21502	0.55
s13207	700	236	165200	16	44	44	105	16	1680	124608	126288	0.76
				32	22	22	128	32	4096	67496	71592	0.43
				64	11	11	117	64	7488	33748	41236	0.25
				128	5	6	103	116	11948	19824	31772	0.19
s15850	611	126	76986	16	38	39	103	16	1648	58968	60616	0.79
				32	19	20	172	31	5332	32760	38092	0.46
				64	10	10	175	61	10675	17640	28315	0.37
				128	5	5	151	121	18271	9450	27721	0.36
s38417	1664	99	164736	16	104	104	273	16	4368	144144	148512	0.90
				32	52	52	510	32	16320	77220	93540	0.57
				64	26	26	683	64	43712	41184	84896	0.52
				128	13	13	656	125	82000	21879	103879	0.63
s38584	1464	136	199104	16	92	92	342	16	5472	175168	180640	0.91
				32	46	46	472	32	15104	93840	108944	0.55
				64	23	23	410	64	26240	50048	76288	0.38
				128	11	12	322	122	39284	26112	65396	0.33

Table 3: Results for a test set independent correction circuit and the test sets used in [18].

Considering again the compression ratio as a function of the number of scan chains we obtain the curves sketched in Figure 9.

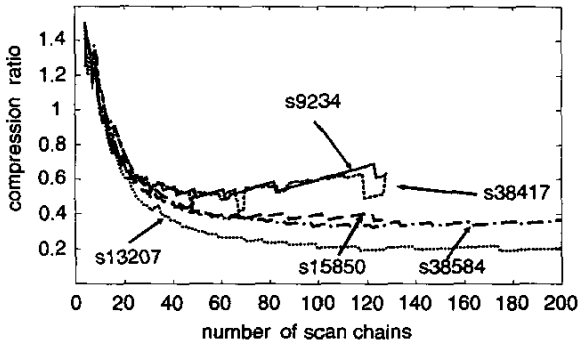

Figure 9: Compression ration as a function of the number of scan chains for the test sets used in [18].

It is interesting to see that for the test sets of [18] an increased number of scan chains cannot guarantee an improved compression ratio any longer. For two circuits, s9243 and s38417, the compression ratio is best for a medium number of scan chains.

A comparison of these data to the best results reported in [18] is provided in Table 4, which contains two lines for each circuit. In the first line the best results for the proposed method are extracted from Table 3, and in the second line the best results from [18] are listed. It can be observed, that for all circuits except s38417, the proposed method provides better compression ratios. For example for s9234 the compression ratio is improved from 71 % down to 48 %. Having a closer look at the results of s38417, it should be noted that in Table 3, which was the basis for our comparison, only powers of 2 are considered as possible numbers of scan chains. According to Figure 9, however, for this circuit the best values are reached for about 50 or 70 scan chains and are in the range of 40 %. Therefore also for circuit s38417 result comparable to the best results in [18] can be achieved.

However, these comparisons are still very coarse, because the method in [18] doesn't require a correction circuit. On the other hand, the proposed approach is fully regular and produces one slice per tester cycle. In contrast to that, the method in [18] cannot accommodate all slices in the dictionary and requires extra cycles when explicit information has to be transmitted for slices not in the dictionary.

Circuit	T_D	Method	Scan chains	d	k	RAM	T_E	$T_E + RAM$	$\frac{T_E + RAM}{T_D}$
s5378	23754	proposed	64	116	54	6264	5328	11592	0.48
		[18]	64	128	64	8192	6345	14537	0,61
s9234	39273	proposed	64	164	61	10004	8904	18908	0.48
		[18]	128	128	128	16384	11498	27882	0,71
s13207	165200	proposed	128	103	116	11948	19824	31772	0.19
		[18]	200	128	200	25600	8517	34117	0,21
s15850	76986	proposed	128	151	121	18271	9450	27721	0.36
		[18]	128	128	128	16384	13873	30257	0,39
s38417	164736	proposed	64	683	64	43712	41184	84896	0.52
		[18]	48	128	48	6144	62939	69083	0,42
s38584	199104	proposed	128	322	122	39284	26112	65396	0.33
		[18]	200	128	200	25600	53287	78887	0,4

Table 3: Comparison to the dictionary based approach in [18].

5 Conclusions

A technique for test data compression for multiple-scan architectures has been presented. The method combines dictionary-based encoding with possible corrections for a fixed number of bit positions. Remaining don't cares are exploited to reduce the number of scan chains to be fed with independent test data. The method doesn't require any structural information about the circuit and fully pre-

serves the advantages of a multiple-scan architecture. Due to the regularity of the proposed scheme each tester cycle provides all the information for a complete slice to be delivered to the scan chains. This way a high throughput is achieved, and the test time is minimized. As the experimental data show, the proposed approach also guarantees an excellent data compression. The compression ratios improve with the number of scan chains yielding a reduction down to 7 %.- 22 % for the best scan configurations. The reported experiments also show that the presented approach favorably compares to previously published work.

6 Acknowledgement

The author would like thank Krish Chakrabarty for providing the test sets used in [18].

7 References

1 M. Abramovici, M. Breuer, A. Friedman: Digital Systems Testing and Testable Design; New York: Computer Science Press (W. H. Freeman and Co.), 1990

2 I. Bayraktaroglou, A. Orailoglu: Test Volume and Application Time Reduction Through Scan Chain Concealment; Proceedings ACM/IEEE Design Automation Conference, Las Vegas, Nevada, USA, 2001, pp. 151-155

3 C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, B. Koenemann: OPMISR: The Foundation for Compressed ATPG Vectors; Proceedings IEEE International Test Conference, Baltimore, MD, USA, November 2001, pp. 748-757

4 F. Brglez et al.: Accelerated ATPG and fault grading via testability analysis; Proc. IEEE International Symposium on Circuits and Systems, Kyoto, 1985

5 F. Brglez, D. Bryan and K. Kozminski: Combinational Profiles of Sequential Benchmark Circuits; Proc. IEEE International Symposium on Circuits and Systems, 1989, pp. 1929-1934

6 A. Chandra, K. Chakrabarty: System-on-a-chip test data compression and decompression architectures based on Golomb codes; IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, No. 3, March 2001, pp. 355-368

7 A. Chandra, K. Chakrabarty : Frequency-Directed Run-Length (FDR) Codes with Application to System-on-a-Chip Test Data Compression ; Proceedings 19th VLSI Test Symposium, Marina Del Rey, Ca., USA, 2001, pp. 42-43

8 C.-A. Chen, S. K. Gupta: Efficient BIST TPG Design and Test Set Compaction via Input Reduction; IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 17, No. 8, August 1998, pp. 692-705

9 P. T. Gonciari, B. M. Al-Hashimi, N. Nicolici: Improving Compression Ratio, Area Overhead, and Test Application Time for System-on-a-Chip Test Data Compression / Decompression; Proceedings 2002 Design and Test in Europe, Paris, France, March 4-8, 2002, pp. 604-611

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

11 A. Jas, N. A. Touba: Test Vector Decompression via Cyclical Scan Chains and Its Application to Testing Core-Based Designs; Proceedings IEEE International Test Conference, Washington, DC, USA, 1998, pp. 458-464

12 A. Jas, J. Gosh-Dastidar, N. A. Touba: Scan Vector Compression/Decompression Using Statistical Coding; Proceedings 17th IEEE VLSI Test Symposium, Dana Point, Ca., USA, 1999, pp. 114-120

13 G. Kiefer, H. Vranken, E. J. Marinissen, H.-J. Wunderlich: Application of Deterministic Logic BIST on Industrial Circuits; Proc. IEEE International Test Conference, ITC 2000, Atlantic City, NJ, October 3 - 5, 2000

14 B. Koenemann: LFSR-Coded Test Patterns for Scan Designs; Proc. Eur. Test Conf., Munich 1991, pp. 237-242

15 B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, D. Wheater: A SmartBIST variant with guaranteed encoding. Proceedings Asian Test Symposium, 2001, pp. 325-330

16 C. V. Krishna, N.A. Touba: Adjustable Width Linear Combinational Scan Vector Decompression; Proc. ACM/IEEE International Conference on Computer-Aided Design (ICCAD), pp. 863-866, 2003.

17 C. V. Krishna, N. A. Touba: 3-Stage Variable Length Continuous-Flow Scan Vector Decompression Scheme; Proc. IEEE VLSI Test Symposium, 2004.

18 L. Li, K. Chakrabarty, N. A. Touba: Test Data Compression Using Dictionaries with Selective Entries and Fixed-Length Indices; ACM Transactions on Design Automation of Electronic Systems, Vol. 8, No. 4, October 2003, pp. 470-490

19 J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K.-H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Qian: Embedded Deterministic Test for Low Cost Manufacturing Test; Proceedings IEEE International Test Conference, Baltimore, MD, October 2002, pp. 301-310

20 S. M. Reddy, K. Miyase, S. Kajihara, I. Pomeranz: On Test Data Volume Reduction for Multiple Scan Chain Designs; ACM Transactions on Design Automation of Electronic Systems, Vol. 8, No. 4, October 2003, pp. 460-469

21 S. Reda, A. Orailoglu: Reducing Test Application Time Through Test Data Mutation Encoding; Proceedings of the 2002 Design and Test in Europe Conference and Exhibition (DATE'02), pp. 387-393

22 K. Sayood: Introduction to Data Compression; London, San Francisco: Academic Press – The Morgan Kaufmann Series in Multimedia Information and Systems, 2nd Edition, 2000

23 H. Tang, S. M. Reddy, I. Pomeranz: On Reducing Test Data Volume and Test Application Time Multiple Scan Chain Designs; Proceedings IEEE International Test Conference, Charlotte, NC, USA, 2003, pp. 1079-1088

24 N. A. Touba and E. J. McCluskey: Altering a Pseudo-Random Bit Sequence for Scan-Based BIST; Proc. IEEE International Test Conference, Washington, DC, 1996, pp. 167-175

25 E. H. Volkerink, A. Khoche, S. Mitra: Packet-based Input Test Data Compression Techniques; Proceedings IEEE International Test Conference; Baltimore, MD, October 2002, pp. 154-163

26 H. Vranken, T. Waayers, H. Fleury, D. Lelouvier: Enhanced Reduced Pin-Count Test for Full-Scan Design; Proceedings IEEE International Test Conference, Baltimore, MD,November 2001, pp. 738-747

27 F. G. Wolff, C. Papachristou: Multiscan-based Test Compression and Hardware Decompression Using LZ77; Proceedings IEEE International Test Conference, Baltimore, MD, October 2002, pp. 331-339

28 A. Wuertenberger, C. S. Tautermann, S. Hellebrand: A Hybrid Coding Strategy for Optimized Test Data Compression; Proceedings IEEE International Test Conference, Charlotte, NC, USA, 2003, pp. 451-459

29 H.-J. Wunderlich, G. Kiefer: Bit-Flipping BIST; Proc. ACM/IEEE Int. Conf. on CAD-96 (ICCAD96), San Jose, CA, November 1996, pp. 337-343