Test Encoding for Extreme Response Compaction

Michael A. Kochte, Stefan Holst, Melanie Elm, Hans-Joachim Wunderlich
Institut für Technische Informatik
Universität Stuttgart
Pfaffenwaldring 47; D-70569 Stuttgart, Germany
email: {kochte, holst, elm, wu}@iti.uni-stuttgart.de

***Abstract*—Optimizing bandwidth by compression and compaction always has to solve the trade-off between input bandwidth reduction and output bandwidth reduction. Recently it has been shown that splitting scan chains into shorter segments and compacting the shift data outputs into a single parity bit reduces the test response data to one bit per cycle without affecting fault coverage and diagnostic resolution if the compactor's structure is included into the ATPG process.**

This test data reduction at the output side comes with challenges at the input side. The bandwidth requirement grows due to the increased number of chains and due to a drastically decreased amount of don't care values in the test patterns.

The paper at hand presents a new iterative approach to test set encoding which optimizes bandwidth on both input and output side while keeping the diagnostic resolution and fault coverage. Experiments with industrial designs demonstrate that test application time, test data volume and diagnostic resolution are improved at the same time and for most designs testing with a bandwidth of three bits per cycle is possible.

***Index Terms*—Embedded Diagnosis, Design for Test, Test Compression, Response Compaction**

I. Introduction

Bandwidth reduction is one of the major concerns for multi-site testing [1, 2]. Reducing bandwidth on the output side affects input side bandwidth, if fault coverage and diagnostic resolution should be maintained. In this work a new test set encoding approach is proposed, which solves this trade-off efficiently and enables multi-site testing with on average three pins for million gates industrial designs.

In [3] a test response compaction scheme is described that relies on scan chain splitting and subsequent compaction with a parity tree. Figure 1 shows the hardware structure of this approach. The existing scan chains are split up to form shorter sequences of scan elements. The data shifted out of the scan chains in one shift cycle is called *vector*. The scan chain splitting leads to fewer, but larger vectors, which are compacted with a parity tree to a single bit each. This results in an extreme bandwidth reduction on the output side.

If the compactor is included during the ATPG process, both fault coverage and diagnostic resolution improves [3]. In this case ATPG produces more patterns to overcome limited observability and fault masking effects due to the compactor. Because of the shorter chains the test time necessary to apply all the patterns is nevertheless smaller than for the original circuit. Yet, applying more patterns in shorter time raises the bandwidth requirement on the input side.

In addition, test set compression in this scheme is faced with overspecification of patterns and a high vector sensitivity:

Overspecification: If an unspecified value occurs in one of the detecting vectors, this vector becomes useless due to X-masking in the parity tree. In sophisticated compactors like [4–11] this is not the case as the detecting flip flop can be propagated to multiple compactor outputs and thereby X-masking can be circumvented. In the scheme at hand a single unspecified value in the detecting vector corrupts the whole vector. As the ATPG process is not aware of the decompressor logic that fills all unspecified bits pseudo-randomly during test application, it generates a large number of specified bits per pattern. In figure 2, only cone C1 contains an undetected fault. However, ATPG will additionally specify all input flip flops of cone C2 in order to avoid any unspecified value in vector v and to establish a defined propagation path to the parity output. As the vectors are large due to the chain splitting, the amount of specified bits per pattern is very high. Hence, test pattern compression techniques based on exploiting don't care values [12–17] fail for the generated test set.

Vector sensitivity: Segmenting the scan chains enlarges the size of the shift vectors. This increases the portion of information encoded in one vector, i.e. the importance of a single vector for high fault coverage and diagnostic resolution grows. Hence, dropping vectors for better input encoding is not an option. At the same time segmenting scan chains raises the probability of masking in the compactor, which also has to be avoided due to the grown importance of vectors.

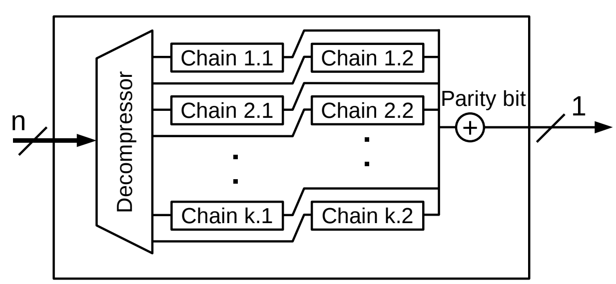

Fig. 1. Extreme response compaction architecture with n inputs and 1 output.

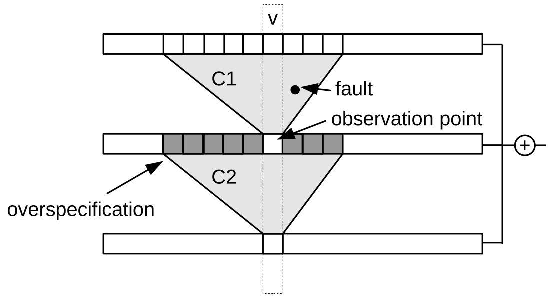

Fig. 2. Overspecification of cone C2.

These effects are illustrated in figure 3. The fault in the circuit can be detected in 9 flip flops. If the scan chains are organized in the standard way, the number of shift vectors carrying information about the fault is 9. If the scan chains are organized as proposed in [3], the number of observing vectors is only 5. If fault masking effects are taken into account, just a single vector will detect the fault.

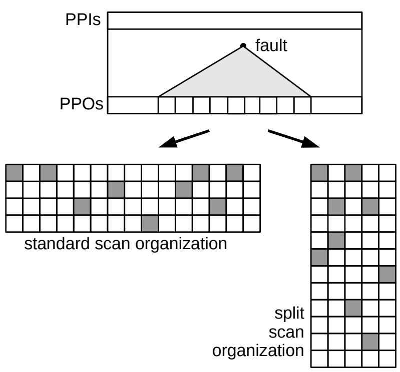

Fig. 3. Masking effects due to fewer but larger shift vectors.

In this paper a new iterative approach for Masking Aware Test set Encoding (MATE) is presented, which fulfills the following requirements:

1) No increase in test time
2) Low constant bandwidth
3) No decrease in fault coverage and diagnostic resolution.

The remainder of this paper is organized as follows. In section II an overview of MATE is given. In the following section the methodology is exemplarily applied to a compression scheme based on partial seeding. In section IV the approach is evaluated in terms of bandwidth requirements, test time, fault coverage and diagnostic resolution.

II. Masking-Aware Test Set Encoding

There are mainly two ways to obtain a compressed test set with maximum fault coverage.

Compressed test patterns can be generated directly by the ATPG by adding a combinatorial representation of the decompressor hardware to the circuit model [18]. For maximum coverage, the compactor at the output side has to be added to the circuit model as well. The efficiency of this approach strongly depends on the type of compression and compaction logic. For the extreme response compaction scheme of figure 1, global reconvergences spanning from the inputs to the outputs are generated. For instance, a 100,000 gate circuit with 28,000 pseudo primary inputs and outputs (PPIs / PPOs) will be mapped to a circuit with just 100 PPIs and PPOs. This results in a large increase in ATPG run time, an increase in test pattern count due to inefficient compaction and a loss of fault coverage due to aborted faults.

The alternative is a two-step process, which first generates an uncompressed test set with maximum coverage and a small number of specified bits, and then encodes the patterns in an efficient way. This also adds flexibility in the choice of the encoding technique. The two-step process is applicable to both encoding techniques which work pattern-by-pattern as well as continuous techniques like partial seeding.

The test set encoding approach proposed here is based on the two-step approach, but overcomes overspecification and vector sensitivity as described above by an iterative encoding process: Test patterns are generated by an ATPG tool for the circuit with attached parity compactor. The overspecification resulting from ATPG is removed by test set stripping considering the circuit without compactor and identifying those bits in the patterns that are not required for fault detection.

The resulting test set contains a significantly smaller number of specified bits and can then be efficiently encoded by known test compression techniques.

Once this test set is encoded, a decompressor model is used to generate the corresponding completely specified test set as applied to the circuit. This fully specified test set is input to fault simulation of the circuit with the parity compactor attached. In some cases, the encoding leads to fault masking, and the process must be repeated with a different pattern or a different encoding. Iteration stops when all targeted faults are detected or saturation is reached.

III. Test Compression Architecture and Test Set Stripping

This section explains the application of the MATE approach to a partial seeding based decompression architecture which is described in the next section. Then, the test set stripping technique is shortly outlined, followed by the detailed flow of the test set encoding procedure.

A. Partial Seeding

The exemplary decompressor architecture used here is based on partial seeding, which injects a fixed amount of free variables into the LFSR in each shift cycle [19, 20]. Figure 4 depicts the decompressor structure with two free variables fed into it in each cycle.

This seeding scheme is similar to [21]. It was chosen as it is often applied in embedded deterministic test environments and allows to constrain the input bandwidth to a fixed value.

The specified bits are encoded vector-by-vector in a continuous way. For each scan cycle, an equation system is built with w free variables. These variables are placeholders for the injects in the current shift cycle and some future shift cycles, the so called *encoding window*. For example in figure 4, six free variables are considered in each encoding step. Here, the encoding window spans three shift cycles. To determine the values of $X0$ and $X1$, the equation system of the first window is solved, and the state of the LFSR is updated accordingly before the next window is processed.

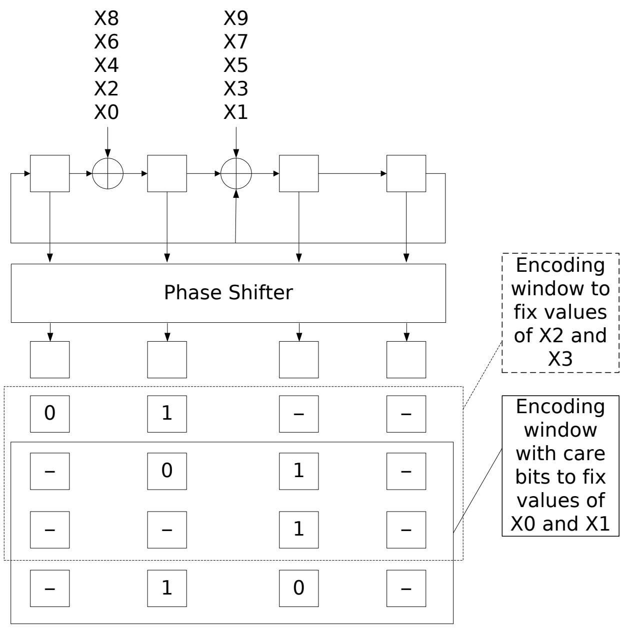

Fig. 4. Example of two encoding windows for an LFSR with $n = 2$ injects per shift cycle.

This encoding procedure is able to balance fluctuations in the density of specified bits because each injected variable contributes to the solution for future specified bits in the corresponding encoding window. If the encoding window is too small, the ability of the LFSR to balance bursts of specified bits is impaired as free variables are fixed too soon and cannot contribute to later bursts of specified bits. If the size of the window is chosen very large, the number of free variables increases and the run time required to solve the set of equations grows quickly. As the distribution of specified bits in a test set may vary strongly both between vectors as well as between patterns, the size of the encoding window is chosen to span a few patterns.

To keep the LFSR size and the amount of input seed bits per cycle as low as possible, it is necessary to start the encoding on a test set which is not too unbalanced and contains as few specified bits as possible. Therefore, the test set generated by ATPG is transformed as described in the next section.

B. Test Set Stripping

Test set stripping or relaxation methods [22–24] allow to uncover a high number of overspecified bits in completely or partially specified test sets without compromising the fault coverage. The stripping method used here allows in addition to bound the number of specified bits in a test pattern to a given limit l [24].

Let P_{ATPG} be the pattern set generated by ATPG and $F_{ATPG-detected}$ the set of faults detected by it. The result after stripping is a pattern set $P_{stripped}$ detecting the same set of faults, where for each pattern $p'_i \in P_{stripped}$ the number of specified bits does not exceed the limit l.

For each pattern $p_i \in P_{ATPG}$, $F_i \subseteq F_{ATPG-detected}$ is the set of faults it detects. The used stripping algorithm first identifies the bits required to detect them. If the number of specified bits does not exceed the limit l, they are written to a new pattern p'_i, which is added to $P_{stripped}$. Otherwise, pattern splitting is employed, as proposed in [25]. It duplicates patterns and distributes target faults to the duplicates. Thus, F_i is split into two disjoint subsets F_{i1} and F_{i2}. Then p_i is stripped twice, once targeting F_{i1} and then targeting F_{i2}. This step is repeated until the limit is enforced.

C. Iterative Test Set Encoding

Figure 5 depicts the overall flow of MATE. The initial test set P_{ATPG} is obtained by a commercial ATPG tool on the circuit with compactor. P_{ATPG} is completely specified and detects the set of target faults $F_{ATPG-detected}$. Test set stripping is

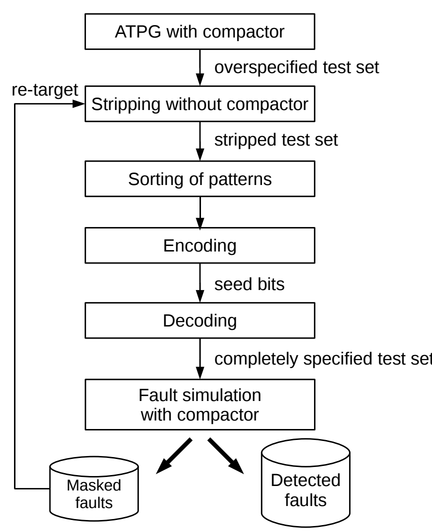

Fig. 5. Overview of the test set encoding flow.

applied to the patterns with a given limit and the resulting set of partially specified patterns $P_{stripped}$ is sorted such that the pattern sequence shows a balanced distribution of specified bits. In the next step, the seed bits for the partial seeding scheme are computed for the ordered patterns. The seed bits are decoded with the LFSR and the resulting fully specified patterns $P_{decoded}$ are subject to fault simulation.

Some of the targeted faults may have been missed because of masking or the rare case of a failed encoding. These faults are targeted again in the next iteration of the flow. Once all target faults F have been detected or saturation is reached, i.e. no targeted faults have been detected in one iteration, the process stops.

The average number of specified bits l_{avg} in a test pattern that can be encoded by the partial seeding scheme for a particular LFSR depends on the number of injects n to the partial seeder, its coding efficiency and the maximum length t of the scan chains in the circuit. Assuming a coding efficiency of 0.9, l_{avg} evaluates to $0.9 \cdot n \cdot t$. The number of injects n to the partial seeding scheme is set to the minimum value that results in no or only a negligible number of failed encodings.

For some faults there exists no pattern in the test set that detects it with at most l_{avg} specified bits. MATE allows that the number of specified bits in a single pattern exceeds l_{avg} as long as the limit is kept in average over multiple patterns. The resulting imbalance of specified bits between patterns is reduced by subsequent reordering of patterns. This guarantees that the average number of specified bits in each encoding window is theoretically encodable for the LFSR.

Nevertheless, even if the number of specified bits theoretically allows encoding, some parts of the vectors or sequences of vectors may exhibit a conflicting combination of specified bits. Here, the encoding might fail in rare cases.

IV. Experimental Results

In this section, we first present the basic characteristics of the considered full-scan designs. Then, we show the bandwidth requirements of the proposed encoding procedure and its impact on pattern count and fault coverage. Finally we discuss the influence of the proposed approach on diagnostic resolution.

A. Circuits characteristics

Table I shows the characteristics of the designs under consideration. These industrial designs are kindly provided by NXP. The name in the first column roughly reflects the number of logic gates in the circuit. Column t shows the maximum scan chain length and column k shows the number of scan chains of the original scan configuration. The testability of these circuits is determined by a commercial ATPG tool generating a fully specified and compacted test set for stuck-at faults. The number of patterns in the resulting test sets are given in column p. The number of shift cycles is given by $t \cdot p$. The absolute numbers of detected, untestable and aborted faults, reported by the ATPG are given in the three last columns. These figures are based on a structurally collapsed stuck-at fault set.

design	t	k	p	$t \cdot p$	det.	untest.	abort
p100k	792	18	2055	1,627,560	166,212	568	180
p141k	486	24	1618	786,348	284,275	3,255	22
p239k	541	40	692	374,372	450,699	5,287	6
p259k	541	40	846	457,686	602,074	5,457	5
p267k	494	45	1139	562,666	370,636	1,504	0
p269k	494	45	1119	552,786	372,792	1,504	0
p279k	409	55	1287	526,383	483,321	10,409	14
p286k	416	55	2149	893,984	637,297	10,731	16
p295k	1852	11	3873	7,172,796	474,942	4,036	18
p330k	317	64	5134	1,627,478	542,054	5,627	127
p378k	64	325	84	5,376	816,274	0	0
p388k	525	50	1007	528,675	852,033	4,610	35
p418k	830	64	1391	1,154,530	678,029	10,715	64
p469k	706	1	317	223,802	167,468	1,755	141
p483k	900	71	493	443,700	912,106	10,748	96

TABLE I
Original design characteristics, ATPG w/o compactor.

B. Pattern generation for extreme compaction

Now, the chains are split to obtain a ratio of about $t \sim \frac{k}{5}$ except for p469k, which has very few scan cells. Parity compactors are attached to all the outputs corresponding to a single vector. These parity compactors compact each vector into one single response bit and thus have a depth of $\log k$. This operation reduces test time and massively reduces the observable response data. For the new designs, the ATPG tool generates a stuck-at fault pattern set, and the outcome is shown in table II.

design	t	k	p	$t \cdot p$	det.	untest.	abort
p100k	53	270	2345	124,285	166,183	526	251
p141k	45	264	2725	122,625	284,202	3,312	38
p239k	61	360	3239	197,579	450,684	5,277	31
p259k	61	360	3777	230,397	602,062	5,446	28
p267k	62	360	4782	296,484	370,507	1,500	133
p269k	62	360	4827	299,274	372,664	1,501	131
p279k	59	385	5248	309,632	483,276	10,424	44
p286k	60	385	6567	394,020	637,294	10,722	28
p295k	62	330	8331	516,522	474,937	4,038	21
p330k	64	320	8849	566,336	541,939	5,619	250
p378k	64	325	169	10,816	816,274	0	0
p388k	66	400	3753	247,698	852,024	4,610	44
p418k	93	576	6742	627,006	677,880	10,767	161
p469k	89	8	319	28,391	167,439	1,755	170
p483k	113	568	4369	493,697	911,828	10,742	380

TABLE II
Results for ATPG with parity compactor.

We observe an increase in pattern count for almost every circuit. However, due to the chain segmentation, the number of vectors $t \cdot p$ (i.e. the test time) is still reduced. Only in one case the ATPG tool chosen here is generating ten times more patterns than in the original circuit configuration (p483k). This is an irregularity of the ATPG tool as a different ATPG tool was able to find a test pattern set with 488 patterns only. In order to present consistent results and for reasons of comparability we nevertheless proceed with the test set generated by the first tool.

The original fault coverage is almost maintained. For some circuits with parity compactor the amount of untestable faults is lower than in the case without compactor. The reason for this is that now the ATPG tool is not able to prove the untestability for some untestable faults. They are just aborted.

The slightly increased number of aborted faults can be avoided by some more sophisticated ATPG heuristics or scan chain organizations [26], but as ATPG is not the subject of this paper, additional means for improving the fault coverage are not discussed here.

C. Time and bandwidth reduction with MATE

Table III shows the results after encoding the ATPG patterns with MATE. For on-chip test set decoding, a 128-bit LFSR is used together with a randomly generated phase shifter with 10 taps per scan chain in average. The number of seed bits that are injected into the LFSR at each scan clock cycle (column *injects*) is determined as described in III. All faults detected by the given ATPG pattern set are targeted by MATE. After MATE saturates, only very few faults remain undetected (column *missed faults*). In some cases, MATE leads to additional detects (column *add. detects*). These faults were previously aborted by the ATPG and are now detected because the pattern set is filled with different bits after encoding. The increase in the pattern count p stems from splitting patterns with too many specified bits to be encoded with the limited number of injects. Still, the resulting test time $t \cdot p$ is in all but two cases better than the time needed for the original designs. The reasons for the exceptions lie in the original circuits (see below). The required bandwidth is at most 3 bits per shift cycle for most of the circuits. For circuit p141k, the amount of specified bits in some patterns is extremely high increasing the number of injects.

To observe the parity bits of the response, one additional bit is needed for each shift cycle, which leads to a total bandwidth of at most 4 bits per cycle. Table IV lists the improvements in bandwidth (column Δbw) and test time with respect to the original circuit (column Δt). We also compare the presented architecture (columns bw_m and t_m) to an architecture without scan chain splitting and test decompressor and an X-compactor attached to the outputs of the scan chains (columns bw_o and t_o). The X-compactor can be parametrized to handle D error bits and U unknowns in an output vector. For $U = 0$ the authors propose a signature of 10 bits, if the vector length is between 257 and 512 [4].

In most cases the proposed approach results in a significant improvement in both bandwidth reduction and test time, but there are some rare cases in which only one parameter is optimized. The original circuit p469k contains just a single scan chain and therefore already has the lowest possible bandwidth requirement. By a small increase in bandwidth, the method presented here reduces the test time by 7.5X. The scan configuration of p378k was already optimal and not subject to splitting. Hence test time could not be improved but increased by a factor of 3.3X. However, at the same time the required bandwidth was reduced by a factor of 167X! Finally,

design	injects	miss. flt	add. det.	p	$t \cdot p$
p100k	1	0	81	3761	199,333
p141k	6	35	8	3491	157,095
p239k	1	0	6	4859	296,399
p259k	1	3	0	6153	375,333
p267k	2	12	16	6949	430,838
p269k	2	15	14	5652	350,424
p279k	2	0	111	8458	499,022
p286k	2	2	12	8543	512,580
p295k	2	1	18	9734	603,508
p330k	3	87	13	9533	610,112
p378k	1	0	1	256	16,384
p388k	2	0	155	5063	334,158
p418k	2	1	96	8010	744,930
p469k	2	0	0	335	29,815
p483k	2	1	902	5601	632,913

TABLE III
ENCODING RESULTS OF MATE.

design	bw_o	bw_m	Δbw	t_o	t_m	Δt
p100k	24	2	12.0X	1,627,560	199,333	8.2X
p141k	30	7	4.3X	786,348	157,095	5.0X
p239k	47	2	23.5X	374,372	296,399	1.3X
p259k	47	2	23.5X	457,686	375,333	1.2X
p267k	52	3	17.3X	562,666	430,838	1.3X
p269k	52	3	17.3X	552,786	350,424	1.6X
p279k	62	3	20.7X	526,383	499,022	1.1X
p286k	62	3	20.7X	893,984	512,580	1.7X
p295k	16	3	5.3X	7,172,796	603,508	11.9X
p330k	71	4	17.8X	1,627,478	610,112	2.7X
p378k	335	2	167.5X	5,376	16,384	0.3X
p388k	57	3	19.0X	528,675	334,158	1.6X
p418k	71	3	23.7X	1,154,530	744,930	1.5X
p469k	2	3	0.7X	223,802	29,815	7.5X
p483k	79	3	26.3X	443,700	632,913	0.7X
avg.			31.2X			2.4X

TABLE IV
BANDWIDTH AND TEST TIME IMPROVEMENTS.

the increase in test time for circuit p483k results from the huge set of test patterns generated by the commercial ATPG tool as described above. If we weight the improvement for each circuit with the size of the circuit, the weighted average of the bandwidth reduction is 31.2X the weighted average of test time improvement is 2.4X.

D. Diagnostic resolution

As shown in [3], diagnostic resolution is at least maintained by extreme response compaction. These results are also applicable here. In order to verify that the test set encoding at the input side did not influence the diagnostic outcome, we performed a short diagnosis experiment on a sample of 400 faults. The sample was picked randomly and contains four fault models (stuck-at, crosstalk, transition, single-victim bridge) with 100 instances each. A diagnosis is considered a success, if there is a single top-candidate which correctly localizes the injected fault.

Table V compares the diagnostic success rate for diagnosis on the full response data of the original circuits without any compaction to the success rate for diagnosis on the extremely compacted response of the compressed test data. We observe a slightly higher fluctuation in the diagnostic success rates for

design	orig.	mate	diff
p100k	84%	80%	–4%
p141k	85%	85%	+0%
p239k	87%	88%	+1%
p259k	80%	82%	+2%
p267k	81%	82%	+1%
p269k	86%	85%	–1%
p279k	77%	82%	+5%
p286k	73%	79%	+6%
p295k	77%	73%	–5%
p330k	81%	79%	–2%
p378k	81%	88%	+7%
p388k	86%	86%	+0%
p418k	82%	83%	+1%
p469k	54%	55%	+1%
p483k	84%	87%	+3%
avg.	80.9%	82.5%	+1.6%

TABLE V
DIAGNOSTIC SUCCESS BEFORE AND AFTER TEST SET ENCODING WITH MATE.

the individual circuits due to the smaller fault samples used in this experiment. Still, the weighted average in the last row again shows that diagnostic resolution is maintained by the proposed iterative test set encoding method.

V. CONCLUSION

ATPG on circuits with extreme response compaction leads to overspecified test patterns, high bandwidth requirements and increased vector sensitivity. This poses a great challenge for test encoding at the input side. The new, iterative encoding method presented in this paper effectively deals with these challenges by employing a combination of test set stripping, pattern splitting, fault simulation and partial reseeding.

For most of the considered industrial designs, MATE achieves a maximum bandwidth of only 3 bits per shift cycle to perform the test while keeping test application time low. For some circuits, the bandwidth requirement can be lowered to only a single input and one output pin. Furthermore, the encoded test set preserves both fault coverage and diagnostic properties of the original test.

ACKNOWLEDGEMENT

This work has been funded by the DFG under contract WU 245/4-1.

REFERENCES

[1] F. Poehl, J. Rzeha, M. Beck, M. Gössel, R. Arnold, and P. Ossimitz, "On-chip evaluation, compensation, and storage of scan diagnosis data - a test time efficient scan diagnosis architecture," in *11th European Test Symposium (ETS 2006), 21-24 May 2006, Southhampton, UK*. IEEE Computer Society, May 2006, pp. 239–246.

[2] A. Kinsman, S. Ollivierre, and N. Nicolici, "Diagnosis of logic circuits using compressed deterministic data and on-chip response comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 537–548, May 2006.

[3] S. Holst and H.-J. Wunderlich, "A diagnosis algorithm for extreme space compaction," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE'09)*, 2009.

[4] S. Mitra and K. S. Kim, "X-compact: an efficient response compaction technique for test cost reduction," in *Proceedings of the International Test Conference*, 2002, pp. 311–320.

[5] J. Rajski, J. Tyszer, C. Wang, and S. Reddy, "Finite memory test response compactors for embedded test applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 4, pp. 622–634, April 2005.

[6] A. Leininger, M. Goessel, and P. Muhmenthaler, "Diagnosis of scan-chains by use of a configurable signature register and error-correcting codes," in *Proceedings of the Design, Automation and Test in Europe Conference*, vol. 2, 2004, pp. 1302–1307.

[7] K. K. Saluja and M. Karpovsky, "Testing computer hardware through data compression in space and time," in *Proceedings of the IEEE International Test Conference*, 1983, p. 8389.

[8] J. Patel, S. Lumetta, and S. Reddy, "Application of Saluja-Karpovsky compactors to test responses with many unknowns," in *Proceedings of the 21st VLSI Test Symposium (VTS)*, 2003, pp. 107–112.

[9] I. Bayraktaroglu and A. Orailoglu, "Deterministic partitioning techniques for fault diagnosis in scan-based BIST." in *Proceedings of the International Test Conference*, 2000, pp. 273–282.

[10] G. Mrugalski, A. Pogiel, J. Rajski, J. Tyszer, and C. Wang, "Fault diagnosis with convolutional compactors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 8, pp. 1478–1494, Aug. 2007.

[11] H. Tang, C. Wang, J. Rajski, S. Reddy, J. Tyszer, and I. Pomeranz, "On efficient X-handling using a selective compaction scheme to achieve high test response compaction ratios," in *Proceedings of the 18th International Conference on VLSI Design*, Jan. 2005, pp. 59–64.

[12] 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 Transactions on Computers*, vol. 44, no. 2, pp. 223–233, 1995.

[13] B. Koenemann, "LFSR-coded test patterns for scan designs," in *Proceedings of the European Test Conference*, 1991, pp. 237–242.

[14] E. H. Volkerink and S. Mitra, "Efficient seed utilization for reseeding based compression," in *Proceedings of the 21st IEEE VLSI Test Symposium.* Washington, DC, USA: IEEE Computer Society, 2003, p. 232.

[15] A.-W. Hakmi, H.-J. Wunderlich, C. G. Zoellin, A. Glowatz, F. Hapke, J. Schloeffel, and L. Souef, "Programmable deterministic built-in self-test," in *Proceedings of the IEEE International Test Conference*, 2007.

[16] H.-J. Wunderlich and G. Kiefer, "Bit-flipping BIST," in *Proceeding of the International Conference on Computer-Aided Design (ICCAD)*, 1996, pp. 337–343.

[17] N. A. Touba and E. J. McCluskey, "Altering a pseudo-random bit sequence for scan-based BIST," in *Proceedings of the IEEE International Test Conference*, 1996, pp. 167–175.

[18] L.-T. Wang, X. Wen, S. Wu, Z. Wang, Z. Jiang, B. Sheu, and X. Gu, "Virtualscan: Test compression technology using combinational logic and one-pass ATPG," *IEEE Design & Test of Computers*, vol. 25, no. 2, pp. 122–130, March-April 2008.

[19] Y.-H. Fu and S.-J. Wang, "Test data compression with partial LFSR-reseeding," in *14th Asian Test Symposium (ATS 2005), 18-21 December 2005, Calcutta, India*. IEEE Computer Society, Dec. 2005, pp. 343–347.

[20] C. V. Krishna, A. Jas, and N. A. Touba, "Achieving high encoding efficiency with partial dynamic LFSR reseeding," *ACM Trans. Design Autom. Electr. Syst.*, vol. 9, no. 4, pp. 500–516, 2004.

[21] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 776–792, May 2004.

[22] K. Miyase and S. Kajihara, "Xid: Don't care identification of test patterns for combinational circuits." *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 23, no. 2, pp. 321–326, 2004.

[23] A. El-Maleh and A. Al-Suwaiyan, "An efficient test relaxation technique for combinational & full-scan sequential circuits," in *Proceedings of the 20th IEEE VLSI Test Symposium*, 2002, pp. 53–59.

[24] M. A. Kochte, C. G. Zoellin, M. E. Imhof, and H.-J. Wunderlich, "Test set stripping limiting the maximum number of specified bits," in *4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA 2008)*, 2008, pp. 581–586.

[25] I. Pomeranz and S. Reddy, "Reducing the number of specified values per test vector by increasing the test set size," *IEE Proceedings of Computers and Digital Techniques*, vol. 153, no. 1, pp. 39–46, Jan. 2006.

[26] M. Elm and H.-J. Wunderlich, "Scan chain organization for embedded diagnosis." in *Proceedings of Design, Automation and Test in Europe (DATE'08)*, 2008.