BISD: Scan-Based Built-In Self-Diagnosis

Melanie Elm and Hans-Joachim Wunderlich
Institute for Computer Architecture and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47
70569 Stuttgart, Germany
{elm, wu}@informatik.uni-stuttgart.de

***Abstract*—Built-In Self-Test (BIST) is less often applied to random logic than to embedded memories due to the following reasons: Firstly, for a satisfiable fault coverage it may be necessary to apply additional deterministic patterns, which cause additional hardware costs. Secondly, the BIST-signature reveals only poor diagnostic information. Recently, the first issue has been addressed successfully. The paper at hand proposes a viable, effective and cost efficient solution for the second problem.**

The paper presents a new method for Built-In Self-Diagnosis (BISD). The core of the method is an extreme response compaction architecture, which for the first time enables an autonomous on-chip evaluation of test responses with negligible hardware overhead. The key advantage of this architecture is that all data, which is relevant for a subsequent diagnosis, is gathered during just one test session.

The BISD method comprises a hardware scheme, a test pattern generation approach and a diagnosis algorithm. Experiments conducted with industrial designs substantiate that the additional hardware overhead introduced by the BISD method is on average about 15% of the BIST area, and the same diagnostic resolution can be obtained as for external testing.

***Index Terms*—Logic BIST, Diagnosis**

I. Introduction

Built-In Self-Test (BIST) has several advantages over external test: Low cost testers can be employed to apply the test, the at-speed capability of BIST enables a high throughput, a high defect coverage is achievable even for defects not explicitly targeted by the test set, and the architecture can be reused in the field [1]. However, for random logic, BIST is not often employed as it entails some non-trivial problems:

1) The hardware overhead for the generation of deterministic test patterns can become prohibitively high.
2) The diagnostic resolution of compacted test responses is in many cases poor, and the overhead required for an acceptable resolution may become too high.

The non-shaded blocks in Fig. 1 show a generic BIST structure for pseudo-random BIST. During a BIST session a series of pseudo-random patterns is generated, applied and the corresponding test responses are compacted into a single signature, which is downloaded to the tester and compared with a reference signature.

For most designs, it is necessary to generate additional deterministic patterns in order to achieve a high defect coverage. In that case, a test pattern memory has to be implemented on chip. It corresponds to the horizontally shaded block in Fig. 1. In the last few years there has been continuous progress in encoding of deterministic pattern sets [2], [3], [4]. Today encoding methods are available, which reduce the pattern memory to an acceptable size even for large industrial designs [5].

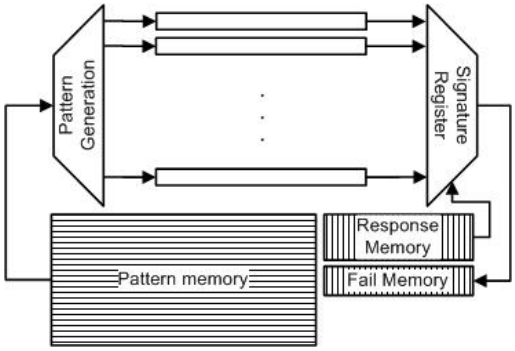

Fig. 1. Generic BIST structure extended by deterministic patterns and a BISD hardware.

In this paper, a modification of the BIST architecture is proposed, which for the first time allows to store the test responses for each test pattern on chip in a response and fail memory (vertically shaded blocks in Fig. 1). It thereby enables a *stop-on-n^{th}-fail* diagnosis for random logic in the same way as it is widely used for embedded memories. The response memory contains the expected signatures for each test pattern, the fail memory contains the first n test signatures deviating from the expected signatures. Core of the architecture is a compactor, which generates extremely short signatures for each test pattern. Consequently, fail and response memory only need negligible chip area compared to the pattern memory.

To circumvent aliasing and poor diagnostic resolution due to the short signatures, the BISD architecture is supplemented with a dedicated automatic test generation method (ATPG) and a statistical diagnosis algorithm, which is able to identify faults even if the test responses are compacted. As a result, the diagnostic success of the combined approach is even higher than that of external testing.

In this paper, X-values in the scan chains are not explicitly considered. Nonetheless, it is possible to combine well-known X-masking schemes with the proposed architecture [6], [7], [8].

The rest of the paper is organized as follows. The next section describes the previous work and state of the art. Section III

introduces the new BISD hardware architecture in detail and compares it to existing architectures. Section IV explains the algorithmic consequences for deterministic pattern generation and logic diagnosis. The experimental results in section V show that without any loss in fault coverage and diagnostic resolution, an autonomous built-in self-diagnosis is obtained by increasing the BIST area by less than 15% in average.

II. Previous Work and State of the Art

Groundbreaking progress in the field of test response modification and generation has been made by Agarwal, Aitken and Zorian [9], [10]. The underlying principle is depicted in Fig. 2. The responses of the device-under-diagnosis (DUD) are modified, i.e. compacted either combinationally [10] or sequentially [9]. The compacted, expected responses are generated on a test chip (e.g. by a ROM or a PLA). The test response is compared to the reference response from the test chip and as soon as the first fail is detected, the corresponding test response is stored in the fail memory. A dedicated diagnosis approach is able to identify the fault from this information.

The largest component in this scheme is the response generation unit which is consequently synthesized onto an external test chip. In addition, this scheme does neither support scan based designs nor deterministic test patterns.

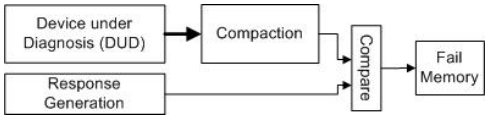

Fig. 2. Test response generation and modification as in [9], [10].

Today, BIST equipped designs are usually tested by compacting all the test responses into a single signature. This signature is downloaded to an external tester, where it is compared to a reference signature. This reveals enough information to separate good devices from faulty ones. But in general, the final signature does not provide sufficient information for diagnosis, and it is necessary to analyze intermediate results of the signature generation [11].

As storing those intermediate signatures on chip would require too much chip area, most BIST schemes are based on a multiphase test. In the first test session, good chips are separated from bad ones as described above. The bad ones are subject to several subsequent test sessions. Three different methods of those test session repetitions can be distinguished: Test sessions can be repeated on different subsets of scan elements or scan chains [12], [13], [14], [15]. They can be repeated on different subsets of test patterns [16] or as in [17], [18], [19] with a different compactor.

Diagnosis based on single or multiple signatures is also a challenge for the logic diagnosis algorithms. They can be classified into direct and indirect approaches. Indirect approaches compute the observing scan elements for each test pattern [20], [17], [21], [22], [23], [24]. This information is input to diagnosis approaches for combinational logic as e.g. [25], [11], [26], [27].

Direct approaches extract the faulty signal directly from the BIST signature without identifying observing scan elements. A direct diagnosis on MISR signatures was proposed in [28] and successfully applied in BIST environments. The method is based on two BIST sessions and the results demonstrate a high diagnostic resolution for random testable faults. Nonetheless, two test sessions are required, and during the second session a high data volume must be transferred between tester and DUD.

Hence, all of these approaches have in common, that several signatures must be evaluated externally and the test has to be repeated several times. A serious drawback is the consequence that the schemes are not directly applicable to the volume test in production. They require multiple test runs for defective chips and in some cases even adaptive testing based on the test outcome is necessary. They make the test flow more complex and increase test application time.

The new BISD scheme presented in the next section collects all the required diagnostic data during a single test run and stores it on chip. It has no drawback on the test flow and test time and allows complete analysis of responses during volume test, multi-site test or in-field test.

III. A New BISD Hardware Scheme

The proposed BISD architecture is depicted in Fig. 3. On the input side, an arbitrary mixed mode BIST scheme can be applied (e.g. [5]). The test set consisting of pseudo-random and deterministic patterns is denoted with T.

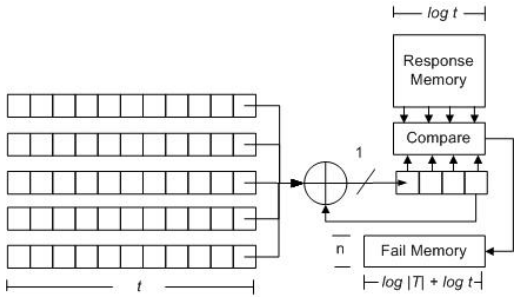

Fig. 3. BISD architecture.

The response compaction consists of four components:

1) an extreme space compaction as in [29],
2) a subsequent time compaction with a Single Input Signature Register (SISR) and
3) a compare unit, which is fed by the SISR and compares its outputs with the reference signature in the on-chip response memory.
4) a fail memory, which stores the first n failing patterns and the corresponding pattern index.

Let k be the amount of scan chains and t the length of the longest scan chain. The k bits shifted out of the scan chains in the same shift cycle are a *shift vector*. The space compactor compacts all bits of a shift vector into a single parity bit. Hence, the space compaction results in a bit sequence of

length t for each test pattern. This bit sequence is fed into a SISR of length $\lceil \log t \rceil$, which generates a unique signature for every single bit failure in the parity bit sequence. Thus, the signature of a test pattern consists of $\lceil \log t \rceil$ bits. We call this compaction scheme XP-SISR.

The compare unit compares the test response to the expected reference response, which is held in the response memory for every (also the random) pattern. Afterwards it resets the SISR, and if a failure is detected, it stores the failing signature in the fail memory together with the corresponding test pattern index. Only the first n failing signatures need to be stored for diagnosis with the diagnosis algorithm described in section IV. Thus, the fail memory only needs to store $n \cdot (\lceil \log |T| \rceil + \lceil \log t \rceil)$ bits. In section V it can be seen that $n = 50$ failing signatures are sufficient for excellent diagnostic resolution.

After the first and only test session, the fail memory is downloaded to the external tester. If it is empty, the chip will be regarded as defect free, otherwise the content of the fail memory is subject to a subsequent diagnosis.

IV. Algorithmic Aspects of BISD

The hardware architecture presented above is based on extreme on-chip test response compaction, and it is highly challenging for both, automatic test pattern generation (ATPG) and diagnosis algorithms. ATPG for mixed-mode BIST as described in [30], [31], e.g., has to support the on-chip storage of deterministic test sets with minimum hardware costs, and tries to minimize the care bits of the test set. This is not directly applicable to the BISD architecture presented above due to the involved extreme space compaction. The required modification of the ATPG algorithms are described below. The combination of extreme space and time compaction puts additional challenges to the diagnosis algorithms. An appropriate algorithm is presented in the second part of this section.

A. ATPG for BISD

If an even number of flip flops observe a fault in each shift vector, all the fault effects will cancel out each other, and the resulting parity bit sequence will appear to be fault free. Thus, ATPG has to make sure, that for each fault f there is at least one test pattern p for which at least one shift vector contains an odd number of faulty scan elements. This can automatically be achieved by any ATPG approach, if the parity compactor is attached to the circuit as in [29]. The circuit, which is input to ATPG, is depicted in Fig. 4. The scan chains are disconnected and an XOR-tree is attached to all those circuit outputs, which correspond to one shift vector. This is logically equivalent to a circuit with scan chains feeding a parity compactor. ATPG automatically avoids aliasing effects in the space compactor by trying to achieve a high fault coverage.

While any ATPG tool could be used, the additional constraints involved make the ATPG problem harder. In general, the harder problem does not lead to an increased number of aborted faults, but to a large number of unnecessarily specified bits (compare [32]).

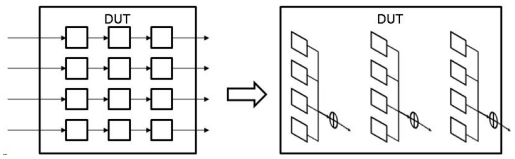

Fig. 4. Circuit model, input to ATPG.

Most mixed-mode logic BIST schemes exploit the fact that the majority of bits in deterministic patterns is not specified, and the hardware overhead introduced by deterministic patterns depends on their amount of care bits [33], [34], [2], [4]. In order to encode the deterministic patterns described above efficiently, additional steps are required:

1) Pattern Stripping: Pattern stripping is the task of removing specified bits from a test pattern without sacrificing fault detection in a combinational circuit. Efficient algorithms are found in [35], [36], e.g. We take each pattern generated above, and do pattern stripping with respect to the original circuit without compactors attached in order to remove overspecified bits. The new pattern contains a small number of specified bits and is suited for deterministic BIST. However, now it is not any more guaranteed that there is at least one response vector with an odd number of erroneous bits. This problem is now tackled by the repeated encoding.

2) Repeated Encoding: Per construction, the stripped pattern can be further specified in such a way that aliasing is avoided. Encoding a stripped pattern by LFSR reseeding leads to a completely specified one, and there is a high chance that aliasing is now excluded. In some few cases, a different encoding may be required to avoid aliasing. In very rare cases, linear dependencies may not allow an aliasing free encoding at all. However, in [32] it was shown that the loss of fault coverage due to this is negligible.

The complete ATPG and encoding flow is now:

1) Generate a pattern for the circuit with parity trees attached.
2) Strip this pattern for the circuit without parity trees.
3) Encode this pattern for reseeding and generate a completely specified pattern this way.
4) If this new pattern detects the fault with parity trees attached, store the seed, else try a different pattern.

As all the seeds stem from stripped patterns with a minimized number of specified bits, the approach has nearly no impact on the size of the seed memory for deterministic logic BIST.

B. Logic Diagnosis for BISD

The direct diagnosis approach employed here is able to identify the underlying fault despite the compact signatures generated by the BISD architecture from section III. It is a modification of the fault model independent method of [37], in order to handle extremely short signatures.

The method is based on the conditional stuck-at line model. A conditional stuck-at line changes the value of a signal under

certain conditions, which can describe values of neighboring signals, temporal or even indeterministic properties. In this way, a large variety of fault models can be represented.

For each stuck-at fault f and every test pattern $p \in T$, the diagnosis starts with generating a reference signature $s_{ref}(f,p)$. $S_{ref}(f) = \bigcup_{p:T}\{s_{ref}(f,p)\}$ refers to the signature of fault f.

The faulty signatures, which were downloaded from the DUD, can easily be adapted to form a similar signature S_{obs} by simply inserting the correct XP-SISR signatures for those test patterns, which did not fail. The signatures $S_{ref}(f)$ and S_{obs} will then be compared and sorted due to the following metrics:

- σ^f: Amount of bits faulty in S_{obs} as well as in $S_{ref}(f)$.
- ι^f: Amount of bits faulty in $S_{ref}(f)$ but correct in S_{obs}.
- τ^f: Amount of bits correct in $S_{ref}(f)$ but faulty in S_{obs}.

The faults will be ordered due to the metrics in such a way, that the first fault most likely corresponds to the actual defect in the DUD. Such an order is

$$r(f) > r(f') \Leftrightarrow \begin{cases} \sigma^f > \sigma^{f'} \text{ or} \\ \sigma^f = \sigma^{f'} \wedge \iota^f > \iota^{f'} \text{ or} \\ \sigma^f = \sigma^{f'} \wedge \iota^f = \iota^{f'} \wedge \tau^f < \tau^{f'} \end{cases}$$

First, faults are sorted due to σ, then ties are broken by sorting due to ι and finally due to τ. The order defined above can be motivated by considering different fault classes. If the defect corresponds to a stuck-at fault, there will be one $S_{ref}(f)$ for which $\iota^f = 0$, $\tau^f = 0$ and $\sigma^f > 0$ is maximum among all faults f. If σ^f is maximum, but $\iota^f > 0$ and $\tau^f = 0$, not all test patterns detecting f also activated f in the DUD. An example is a transition fault on a line, whose value depends on the value from the previous cycle. If $\tau^f > 0$ for all faults f, there is more than one signal involved in the defect. An example for this are 4-way bridges. Also in this case σ^f will be maximum for those stuck-at faults, best explaining the defect. Hence, the signal corresponding to the fault f with highest rank is most likely culprit of the faulty behavior. In this way it is possible to diagnose the faulty signals directly from the signatures in the fail memory.

V. Experimental Results

To validate the BISD architecture, experiments with industrial designs kindly provided by NXP have been conducted. Their characteristics are given in table I. In column *design* the design name is given, in column *# gates* the amount of gates. *# PPO* corresponds to the number of pseudo primary outputs, *k* to the amount of scan chains and *t* is the length of the longest scan chain.

For these circuits three questions have been investigated:

1) Input side: What is the influence of the compaction scheme on the length of the ATPG test set and what is its fault coverage?
2) Output side: What are the hardware costs of the presented BISD scheme in terms of memory compared with the costs for a standard mixed mode logic BIST?

design	# gates	# PPO	k	t
p100k	84356	5829	270	53
p141k	152808	10502	264	45
p239k	224597	18495	360	61
p259k	298796	18495	360	61
p267k	238697	16621	360	62
p269k	239771	16621	360	62
p279k	257736	17827	385	59
p286k	332726	17835	385	60
p295k	249747	18521	330	62
p330k	312666	17468	320	64
p378k	341315	17420	325	64

TABLE I. Circuit characteristics.

3) Diagnosability: What is the diagnostic resolution obtainable from the on-chip fail memory compared with the diagnostic resolution provided by external testing without any compaction?

A. Fault Coverage and Test Length

An arbitrary reseeding-based mixed-mode BIST approach has been emulated, which first applies 4096 pseudo-random patterns generated by an LFSR. For the circuit with attached parity tree those faults are determined, which are detected by the random patterns. To minimize the test time, the LFSR is stopped after the last random detectable fault has been detected. For the remaining faults deterministic patterns are generated.

The results of the test pattern generation are given in table II. Column *Original* gives the results for the original design without any compaction, column *XP-SISR* gives the results for the circuit with attached XP-SISR. The amount of applied random patterns can be found in subcolumn *# rp*. Their resulting fault coverage is given in subcolumn *fc r*. Only one circuit is completely random testable. Subcolumns *# p* and *fc* give the size and fault coverage of the final test set containing random and deterministic patterns.

Most mixed-mode BIST schemes require that the number of bits per pattern is restricted, for the BISD scheme here this is ensured by the pattern stripping step explained above. The reduced number of care bits often prevents pattern compaction and may lead to larger test sets. For some circuits a growth of the test set can also be observed here. In general the pattern set size increases at most by a factor of two. As described in section IV it is still efficiently encodable, so that the size of the pattern memory does not increase. For most circuits the fault coverage can be maintained, in some few cases there is a negligible loss due to the reasons explained above.

		Original			XP-SISR		
design	# rp	fc r	# p	fc	fc r	# p	fc
p100k	4088	90.46%	5397	99.56%	90.45%	5726	99.56%
p141k	4091	89.80%	5642	98.86%	89.79%	6712	98.86%
p239k	4096	94.22%	4778	98.84%	94.21%	7170	98.84%
p259k	4095	94.72%	4919	99.10%	94.72%	7702	99.10%
p267k	4096	88.52%	5191	99.60%	88.48%	8505	99.56%
p269k	4095	88.45%	5164	99.60%	88.40%	8550	99.56%
p279k	4096	87.63%	5360	97.89%	87.51%	9212	97.78%
p286k	4094	87.72%	6224	98.34%	87.71%	10524	98.34%
p295k	4096	73.32%	7916	99.15%	73.31%	12317	99.15%
p330k	4096	84.31%	9165	98.95%	84.27%	12927	98.91%
p378k	664	100.00%	664	100.00%	100.00%	664	100.00%

TABLE II. Test pattern sets and fault coverage.

B. Hardware Cost for BISD

The reduction of test and diagnostic data volume is the backbone of this work. In contrast to formerly proposed BIST compaction schemes, the XP-SISR for the first time fulfills all necessary conditions to store the reference signatures and the failing signatures on chip. Table III reports the combined size of the fail memory and the response memory for the XP-SISR and compares it to the memory size necessary for generating deterministic test patterns. The memory sizes on the input side have been taken from the BIST scheme presented in [5] and were chosen for comparison, as this scheme is one of the most efficient schemes published so far. It shows that diagnosability can be added to BIST at negligible costs in many cases.

Column *Input* gives the size of the pattern memory in kilo byte. Column *Responses* gives the size of the response and fail memory. The last column shows the increase in memory introduced by the proposed BISD method with respect to the pattern memory of the BIST scheme. For circuit p100k the test set is extraordinarily well compressible. Thus, the overhead for the BISD scheme is with 60% the largest of all circuits. In most cases, the overhead is less than 15% of the total BIST costs. Additionally, it is obvious that with growing circuit size the overhead is shrinking. The reason is roughly that the response volume grows logarithmically with the circuit size while the care bits in the test set grow linearly. Thus, with bigger circuits, the increase becomes relatively smaller.

design	Input ([5]) [KB]	Responses [KB]	Increase (%)
p100k	7.25	4.41	60.83%
p141k	36.18	5.15	14.24%
p239k	17.97	5.50	30.58%
p259k	23.54	5.89	25.04%
p267k	47.95	6.50	13.55%
p269k	47.44	6.53	13.77%
p279k	48.37	7.03	14.53%
p286k	63.69	8.01	12.58%
p295k	not available	9.36	not available
p330k	76.56	9.82	12.82%
p378k	not available	0.59	not available

TABLE III. Overhead for fail and response memory.

C. Diagnostic Resolution of BISD

This subsection investigates whether the extreme compaction can maintain diagnostic resolution and whether diagnosis on the restricted number of signatures in the fail memory works reliably. For diagnostic validation 200 stuck-at, 200 delay, 200 crosstalk faults and 200 wired-and bridges have randomly and uniformly been injected into the circuits an the test sets of table II were applied. Diagnosis has been conducted on the fail memory containing the first $n = 50$ faulty signatures. The results are reported in table IV. Diagnosability is measured in percentage of perfect matches, where a perfect match means, that the first candidate in the ranked list of faults can indeed explain the defect.

The column *Orig.* gives the diagnosability which can be obtained by external testing without any compaction. The column XP-SISR reports the diagnosability which can be obtained by the BISD strategy presented above, Δ is the difference compared to the original circuit and column *CR* denotes the compaction ratio. The last row reports the average of all these numbers weighted by the circuit sizes. Amazingly, for most of the circuits the diagnostic resolution on a limited fail memory outperforms the diagnosability obtained by external testing, and in average diagnosability is 2.6% better. The reason for this is twofold. First, the BISD test set created by the method of section IV is larger than the original one in general. The less compacted patterns are able to distinguish more faults. Second, the ATPG tool tries to propagate the fault to as many PPOs as possible to avoid aliasing for the circuit with parity tree. The consequence is, that at the cost of slightly more ATPG and encoding effort diagnostic resolution and the compaction ratio are enhanced at the same time.

design	Orig.	XP-SISR (Δ)	CR
p100k	72.4%	71.9% (–0.5)	971
p141k	73.2%	73.5% (+0.2)	1750
p239k	81.4%	82.1% (+0.8)	3082
p259k	76.2%	77.9% (+1.6)	3082
p267k	67.6%	74.8% (+7.1)	2770
p269k	67.4%	73.6% (+6.2)	2770
p279k	68.5%	71.1% (+2.6)	2971
p286k	65.4%	71.2% (+5.9)	2972
p295k	55.8%	55.6% (–0.1)	3086
p330k	67.1%	71.9% (+4.7)	2911
p378k	86.6%	86.1% (–0.5)	2903
avg.	71.7%	74.3% (+2.6)	

TABLE IV. Diagnostic resolution after 800 fault injections.

The compaction ratio *CR* denotes the factor by which the response information is reduced. By applying the proposed BISD strategy, the response data volume is reduced by a factor of several 1000X, the diagnostic resolution is increased significantly within a single test run, and the costs are just a 15% larger BIST memory.

Diagnosis strategies mentioned in section II which use the original test set in repeated BIST sessions cannot exceed the results of external testing for the original circuit and there is no need for comparison.

As a side effect of the diagnosis experiments, we also get an estimate of the defect coverage for non-target defects (here delay faults, wired-and bridges and crosstalk faults). These estimates are based on the coverage of the injected faults and are given in table V for all the three fault models. Again the BISD outperforms external testing, which can be explained by the same reasons as the improvement in diagnosability.

design	Delay Faults		Crosstalk Faults		Wired-And Faults	
	Orig.	BISD	Orig	BISD	Orig.	BISD
p100k	96.5%	98.5%	90.0%	91.5%	96.5%	97.0%
p141k	93.5%	94.5%	89.5%	92.0%	97.0%	97.5%
p239k	96.5%	98.0%	89.5%	92.5%	96.0%	96.0%
p259k	95.5%	98.0%	89.0%	95.5%	97.0%	97.5%
p267k	94.0%	95.5%	88.5%	93.0%	94.5%	97.0%
p269k	94.5%	96.5%	82.5%	91.0%	94.0%	96.5%
p279k	93.5%	94.5%	82.5%	88.0%	94.0%	96.5%
p286k	91.0%	93.0%	81.5%	92.0%	93.0%	96.5%
p295k	85.0%	86.5%	71.0%	79.0%	84.0%	87.5%
p330k	93.5%	95.0%	87.0%	92.0%	95.0%	96.5%
p378k	100.0%	100.0%	97.5%	97.5%	98.5%	98.5%

TABLE V. Estimate of the defect coverage for delay, bridging and crosstalk faults.

VI. Conclusion

A Built-In Self-Diagnosis architecture was presented, which enables the collection of sufficient diagnostic data during a single BIST session. The backbone of the architecture is an extreme compaction, which allows to store expected and failing responses on chip with negligible hardware overhead. Hence, volume diagnosis is enabled in BIST scenarios as volume, multi-site or in-field test without the necessity of additional test sessions for defective devices. Experiments with large industrial designs reveal that a high defect coverage can be achieved, diagnostic resolution can even be improved and the overhead for storing the expected responses on chip is negligible.

Acknowledgment

This work has been funded by the DFG under contract WU 245/4-1. We would like to thank NXP for providing the circuits and Stefan Holst for hints on the implementation.

References

[1] H. Wunderlich, "Bist for systems-on-a-chip," *INTEGRATION, the VLSI journal*, vol. 26, no. 1-2, pp. 55–78, 1998.

[2] 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.*, pp. 167–175, 1996.

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

[4] G. Kiefer, H. Vranken, E. J. Marinissen, and H.-J. Wunderlich, "Application of deterministic logic bist on industrial circuits," in *Proceedings of the 31st IEEE International Test Conference.*, pp. 105–114, 2000.

[5] A.-W. Hakmi, S. Holst, H.-J. Wunderlich, J. Schloeffel, F. Hapke, and A. Glowatz, "Restrict encoding for mixed–mode bist," in *Proceedings of the 27th IEEE VLSI Test Symposium (VTS '09)*, pp. 179–184, 2009. accepted for IEEE VLSI Test Symposium (VTS'09).

[6] M. Naruse, I. Porneranz, S. Reddy, and S. Kundu, "On-chip compression of output responses with unknown values using lfsr reseeding," in *Proceedings of the International Test Conference.*, vol. 1, pp. 1060–1068, 2003.

[7] Y. Tang, H.-J. Wunderlich, H. P. E. Vranken, F. Hapke, M. Wittke, P. Engelke, I. Polian, and B. Becker, "X-masking during logic BIST and its impact on defect coverage," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 442–451, February 2006.

[8] P. Wohl, J. Waicukauski, and S. Patel, "Scalable selector architecture for x-tolerant deterministic bist," in *41st Design Automation Conference.*, pp. 934–939, 2004.

[9] R. Aitken and V. Agarwal, "A diagnosis method using pseudo-random vectors without intermediate signatures," in *Digest of Technical Papers of the IEEE International Conference on Computer-Aided Design.*, pp. 574–577, 1989.

[10] Y. Zorian and V. Agarwal, "A general scheme to optimize error masking in built-in self-testing," *Digest of Papers*, p. 410, 1986.

[11] J. Waicukauski and E. Lindbloom, "Failure diagnosis of structured vlsi," *Design & Test of Computers, IEEE*, vol. 6, pp. 49–60, Aug 1989.

[12] I. Bayraktaroglu and A. Orailoglu, "Cost-effective deterministic partitioning for rapid diagnosis in scan-based bist," *IEEE Design & Test of Computers.*, vol. 19, pp. 42–53, Jan/Feb 2002.

[13] Y. Nakamura, T. Clouqueur, K. Saluja, and H. Fujiwara, "Diagnosing at-speed scan bist circuits using a low speed and low memory tester," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.*, vol. 15, pp. 790–800, July 2007.

[14] J. Ghosh-Dastidar and N. Touba, "A rapid and scalable diagnosis scheme for bist environments with a large number of scan chains," in *Proceedings of the IEEE VLSI Test Symposium.*, pp. 79–85, 2000.

[15] J. Rajski and J. Tyszer, "Diagnosis of scan cells in bist environment," *IEEE Transactions on Computers.*, vol. 48, pp. 724–731, Jul 1999.

[16] P. Wohl, J. Waicukauski, S. Patel, and G. Maston, "Effective diagnostics through interval unloads in a bist environment," in *Proceedings of the Design Automation Conference.*, pp. 249–254, 2002.

[17] 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, pp. 1302–1307, 2004.

[18] C. Liu, K. Chakrabarty, and M. Goessel, "An interval-based diagnosis scheme for identifying failing vectors in a scan-bist environment," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition.*, pp. 382–386, 2002.

[19] Y. Wu and S. Adham, "Scan-based bist fault diagnosis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*, vol. 18, pp. 203–211, Feb 1999.

[20] 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, pp. 622–634, April 2005.

[21] N. Touba, "X-canceling misr - an x-tolerant methodology for compacting output responses with unknowns using a misr," in *IEEE International Test Conference.*, pp. 1–10, 2007.

[22] S. Mitra and K. S. Kim, "X-compact: an efficient response compaction technique," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 23, pp. 421–432, March 2004.

[23] K. K. Saluja and M. Karpovsky, "Testing computer hardware through data compression in space and time," in *Proceedings of the IEEE International Test Conference.*, pp. 83–89, 1983.

[24] 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.*, pp. 107–112, 2003.

[25] M. Abramovici and M. Breuer, "Fault diagnosis based on effect-cause analysis: An introduction," in *17th Conference on Design Automation.*, pp. 69–76, 1980.

[26] R. Desineni, O. Poku, and R. D. Blanton, "A logic diagnosis methodology for improved localization and extraction of accurate defect behavior," in *IEEE International Test Conference.*, pp. 1–10, 2006.

[27] M. E. Amyeen, D. Nayak, and S. Venkataraman, "Improving precision using mixed-level fault diagnosis," in *IEEE International Test Conference.*, pp. 1–10, 2006.

[28] W.-T. Cheng, M. Sharma, T. Rinderknecht, L. Lai, and C. Hill, "Signature based diagnosis for logic bist," in *Proceedings of the IEEE International Test Conference (ITC '06).*, pp. 1–9, 2006.

[29] S. Holst and H.-J. Wunderlich, "A diagnosis algorithm for extreme space compaction," in *Proceedings of the Design Automation and Test in Europe Conference and Exhibition (DATE '09)*, pp. 1355–1360, 2009. accepted for Date'09.

[30] S. Hellebrand, B. Reeb, S. Tarnick, and H.-J. Wunderlich, "Pattern generation for a deterministic bist scheme," in *Digest of Technical Papers of the IEEE/ACM International Conference on Computer-Aided Design.*, pp. 88–94, 1995.

[31] M. Karkala, N. Touba, and H.-J. Wunderlich, "Special atpg to correlate test patterns for low-overhead mixed-mode bist," in *Proceedings of the Seventh Asian Test Symposium.*, pp. 492–499, 1998.

[32] M. A. Kochte, S. Holst, M. Elm, and H.-J. Wunderlich, "Test encoding for extreme response compaction," in *Proceedings of the 14th IEEE European Test Symposium*, pp. 155–160, 2009.

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

[34] 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.

[35] K. Miyase and S. Kajihara, "Xid: Don't care identification of test patterns for combinational circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*, vol. 23, pp. 321–326, Feb. 2004.

[36] M. Kochte, C. Zoellin, M. Imhof, and H.-J. Wunderlich, "Test set stripping limiting the maximum number of specified bits," in *Proceedings of the 4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA '08)*, pp. 581–586, 2008.

[37] S. Holst and H.-J. Wunderlich, "Adaptive debug and diagnosis without fault dictionaries," in *Proceedings of the 12th IEEE European Test Symposium (ETS '07)*, pp. 7–12, 2007.