Structural In-Field Diagnosis for Random Logic Circuits

Cook, Alejandro; Elm, Melanie; Wunderlich, Hans-Joachim; Abelein, Ulrich

Proceedings of the 16th IEEE European Test Symposium (ETS'11) Trondheim, Norway, 23-27 May 2011

doi: http://dx.doi.org/10.1109/ETS.2011.25

Abstract: In-field diagnosability of electronic components in larger systems such as automobiles becomes a necessity for both customers and system integrators. Traditionally, functional diagnosis is applied during integration and in workshops for infield failures or break-downs. However, functional diagnosis does not yield sufficient coverage to allow for short repair times and fast reaction on systematic failures in the production. Structural diagnosis could yield the desired coverage, yet recent builtin architectures which could be reused in the field either do not reveal diagnostic information or necessitate dedicated test schemes. The paper at hand closes this gap with a new built-in test method for autonomous in-field testing and in-field diagnostic data collection. The proposed Built-In Self-Diagnosis method (BISD) is based on the standard BIST architecture and can seamlessly be integrated with recent, commercial DfT techniques. Experiments with industrial designs show that its overhead is marginal and its structural diagnostic capabilities are comparable to those of external diagnosis on high-end test equipment.

Structural In-Field Diagnosis for Random Logic Circuits

Alejandro Cook, Melanie Elm, Hans-Joachim Wunderlich
University of Stuttgart
Institute of Computer Architecture and Computer Engineering
D-70569 Stuttgart, Germany
email: {cook, elm, wu}@iti.uni-stuttgart.de

Ulrich Abelein
AUDI AG
D-85045 Ingolstadt, Germany
email: ulrich.abelein@audi.de

***Abstract*—In-field diagnosability of electronic components in larger systems such as automobiles becomes a necessity for both customers and system integrators. Traditionally, functional diagnosis is applied during integration and in workshops for in-field failures or break-downs. However, functional diagnosis does not yield sufficient coverage to allow for short repair times and fast reaction on systematic failures in the production. Structural diagnosis could yield the desired coverage, yet recent built-in architectures which could be reused in the field either do not reveal diagnostic information or necessitate dedicated test schemes.**

The paper at hand closes this gap with a new built-in test method for autonomous in-field testing and in-field diagnostic data collection. The proposed Built-In Self-Diagnosis method (BISD) is based on the standard BIST architecture and can seamlessly be integrated with recent, commercial DfT techniques. Experiments with industrial designs show that its overhead is marginal and its structural diagnostic capabilities are comparable to those of external diagnosis on high-end test equipment.

***Index Terms*—In-field diagnosis, Built-In Self-Diagnosis**

I. Introduction

As the complexity of embedded devices continues to grow, test and diagnosis at system level in the field have become more challenging. At the system level, only a limited view on the hardware structure of a chip is available and, therefore, it is often not possible to track down the exact root cause of a failure. System vendors are then forced to hand the failure information down the supplier chain until it eventually reaches the chip manufacturer. As a result, the failure diagnosis in the field requires a considerable amount of time and effort.

Built-In Self Test (BIST) has been recognized as a promising solution since it applies structural tests to isolated components and can provide structural test information to the system level. BIST can be executed in the field and allows at-speed test [1] as the test data is generated within the component. System integrators and workshops can, therefore, benefit greatly from BIST if proper diagnosis methods are provided together with BIST-equipped components.

Diagnosis techniques for random logic BIST can roughly be categorized into two groups. The first group relies on the standard BIST architecture also referred to as STUMPS architecture [2]. The diagnosis methods proposed for this architecture are based on repeated test sessions, where the bandwidth necessary for the repetitions is not suitable for in-field scenarios and the test repetitions themselves already cause overhead in terms of test time [3], [4], [5], [6], [7]. The second diagnosis technique for random logic BIST relies on a dedicated hardware architecture [8]. While only one test session is necessary for the collection of diagnostic information, the architecture has the disadvantage, that it requires a dedicated hardware structure and its corresponding synthesis and test pattern generation (ATPG) processes.

In this paper a diagnosis technique is presented, which for the first time overcomes the disadvantages of the above architectures and methods. It is based on the standard STUMPS architecture and thus can be used with the standard tool chains, ATPG and synthesis flows. Yet, it overcomes the necessity of repeated tests and extra bandwidth by adding a small response memory and slightly modifying the BIST controller. Intermediate reference signatures are stored in the response memory and are used for online test evaluation. The resulting failure information can be read out upon test completion and is input to the proposed diagnosis technique. The method is not confined to the stuck-at fault model, but is able to cover a large variety of defects. The diagnostic success of the method is even superior to that typically achieved in external testing.

The paper is structured as follows. Section II gives a short overview of the state of the art in diagnosis methods for random logic BIST and their shortcomings. Section III presents the novel diagnosis algorithm for BIST and in section IV the results obtained for a set of industrial benchmark designs are discussed. Section V provides some concluding remarks.

II. State of the Art

Traditionally, BIST approaches employ the generic architecture shown by the non-shaded blocks in figure 1. On-chip pattern generation makes use of inexpensive pseudo-random patterns to feed a number of parallel scan chains. However, most designs also require deterministic patterns in order to achieve acceptable fault coverage. To this end, an embedded pattern memory stores compressed pattern information subsequently decoded during test application. An additional controller is necessary to control the mixed pseudo-random and deterministic pattern generation. This is depicted in the shaded blocks in figure 1.

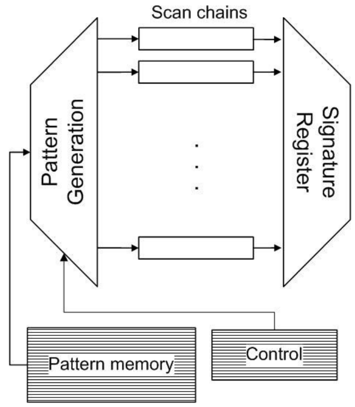

Fig. 1. BIST infrastructure for embedded diagnosis.

The resulting test response data from the complete test session is compressed into a single signature and then downloaded to an external tester where it is compared to a fault-free test signature. While this procedure is adequate to detect failures, it has been recognized that a single signature does not provide enough information to enable fault diagnosis.

Most available diagnostic solutions for BIST require several test sessions to narrow down the number of fault candidates in the diagnostic procedure. The test sessions may target specific scan elements [3], [4], [5], [6], work on different pattern sets [7], or employ different response compactors [9], [10]. Once a set of faulty signatures is identified, logic diagnosis can proceed following one of two approaches.

In indirect diagnosis, the values captured by the scan elements are computed for each pattern from the failing signatures [11], [9], [12], [13], [14], [15]. Diagnosis algorithms for combinational logic can then be used on the resulting failure information [16], [17], [18], [19].

In direct diagnosis, the fault location is identified directly from the faulty signature, without sorting out the values of each and every scan element. Such a direct approach has been proposed in [20], where the authors achieve high diagnostic resolution from the failure responses from a multiple input signature register (MISR). However, this method still requires two test sessions: one to gather the complete test response covering all patterns, and, only in the case of a faulty chip, a second test session where each test pattern response is compacted into a signature register.

More recently, a novel built-in self diagnosis (BISD) architecture was proposed in [8], requiring only a single test session and achieving high fault coverage and diagnosis resolution. The architecture, however, substantially differs from the STUMPS scheme and the high fault coverage and diagnostic resolution result from dedicated synthesis and ATPG methods. In contrast to this, the diagnosis method and according architectural modifications proposed in this paper, are built upon the standard STUMPS architecture and can be used with its well-established tool chains. Nonetheless, the advantage of a single test session and a high fault coverage and diagnostic resolution is still achieved.

Figure 2 depicts a generalization of the STUMPS architecture assumed in the approach presented below. It is based on the observations from [7], where it was shown, that diagnosis from intermediate signatures is possible. The architecture in figure 2 can collect these intermediate signatures in the field.

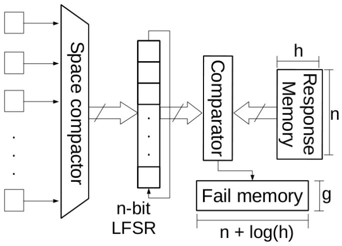

Fig. 2. BISD compactor architecture based on STUMPS.

An n-bit linear feedback shift register (LFSR) is fed by a space compactor succeeding the scan chains. A response memory is added. It contains h intermediate test responses, each of which contains n bits. After an intermediate test signature is obtained, it is compared to the expected test signature in the response memory. If the two signatures differ, the obtained signature is stored in the fail memory along with its intermediate signature index, thus resulting in a fail memory width of $n + \log h$ bits. The state of the LFSR is reset after every intermediate test response is generated. The depth of the fail memory is limited to g.

The content of the fail memory can be downloaded at system level. A diagnosis suitable for the resulting signatures is described in the following section.

III. Diagnosis on Signatures

A. The General Concept

In [7] it was already shown that if a defect behaves like a stuck-at fault, the information of the intermediate signatures from the fail memory is sufficient for diagnosis. However, in this work we assume arbitrary defects and propose a corresponding diagnosis method.

Generally, a subsequence of pattern responses from the test set is mapped into a single signature. For each subsequence, we perform a fault analysis without restricting assumptions on the faulty behavior as long as it is caused by a single defective location. In particular, non-deterministic and sequential faults are covered as well. Finally, the diagnosis results of all subsequences are evaluated to find a single culprit.

In order to analyze complex defect mechanisms, we make use of the *conditional stuck-at* fault model [21]. The conditional stuck-at model represents a stuck-at fault on a single line in certain situations. That is, depending on some internal or external condition, a line may behave as a stuck-at-1 or stuck-at-0 fault for some patterns, or even as a fault-free line for other patterns that would usually excite a fault behavior. For each line *v*, we consider the conditional stuck-at-faults $cond\_0\_v$ and $cond\_1\_v$. The condition *cond* may describe arbitrary Boolean or timing properties. For instance, $(v = 1)\_0\_v$ is a permanent stuck-at-0, and $(v_{-1} = 0 \wedge v = 1)\_0\_v$ describes a slow-to-rise fault.

Diagnosis based on this fault model relies on pattern-wise information of outputs in order to reason about possible stuck-at candidates and their activation conditions. In an embedded environment, however, the circuit's responses are heavily compacted over multiple patterns and, therefore, only the aggregated effect of several patterns is available for diagnostic purposes.

To enable diagnosis on such highly compacted test responses a preprocessing step is necessary to extract the pattern-wise failure output information encoded in the signatures. The main problem at hand is then to identify, for every candidate fault from the fault model, a sequence of likely faulty and fault-free test responses, whose combined effect matches the observed faulty signature. From the resulting sequences any diagnosis routine for the conditional stuck-at model can be applied.

The following two sections describe how the most likely response sequences can be derived for every stuck-at fault with the information stored in the fail memory.

B. Generating Fault-Free Signatures for the Response Memory

We assume *m* is the maximum length of all the scan chains in the STUMPS scheme, and the results of each test pattern are compacted into a single signature in *m* clock cycles. Let *n* be less than or equal to the length of the LFSR, if the number of scan chains exceeds *n*, a space compactor may be used. Let *T* be the set of test patterns, which is partitioned into $h = \lceil \frac{|T|}{n} \rceil$ blocks B. Each block contains *n* patterns at most. The patterns of each block are compacted into a single signature.

The state transition function of an LFSR can be represented as a feedback matrix, e.g. for a type-I LFSR

$$L = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & l_1 & l_2 & \cdots & l_{n-2} & l_{n-1} \end{bmatrix}$$

where matrix elements l_i correspond to the coefficients of the LFSR generator polynomial.

The matrix $H = L^m$ describes the autonomous function of the LFSR after *m* cycles. Each block

$$B = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \tag{1}$$

provides a signature S_B. Let s_i be the signature from pattern p_i which is obtained by shifting the *m* response vectors of p_i into the LFSR starting in the all-zero state. Applying linear superposition, the final signature after applying all patterns in *B* is captured by the equation:

$$S_B = \sum_{i=1}^{n} H^{\,n-i}\, s_i \tag{2}$$

After each block *B*, the LFSR is reset and only S_B has to be stored in the response memory.

C. Analysis of Erroneous Signatures from the Fail Memory

The block *B* may contain some test patterns that activate the fault *s@0-v*, that is, a stuck-at-0 on location *v* (or *s@1-v*, a stuck-at-1 on *v*) and, according to the conditional stuck-at fault model, depending on a given condition *cond*, these patterns may also detect *cond_0_v* (or *cond_1_v*). Hence, we have to determine the signatures for the unconditional *s@0-v* (or *s@1-v*), and we have to select a pattern subsequence which fits *cond_0_v* (or *cond_1_v*).

Let *f* be such an unconditional stuck-at fault, and let s_i^f be the signature of pattern p_i in the presence of *f* if the LFSR starts in the all-zero state. The error vectors are defined as:

$$e_i^f := s_i \oplus s_i^f \tag{3}$$

We have $|e_i^f| \neq 0$ if and only if pattern p_i detects *f* in its signature.

Now assume, the real fault $\widetilde{f}$ is a conditional stuck-at fault. In this case, we have either $e_i^{\widetilde{f}} = e_i^f$, if in pattern p_i the condition is true, or we have $|e_i^{\widetilde{f}}| = 0$.

This can be decided by solving a set of linear equations. Let

$$d_i^f := H^{\,n-i}\, e_i^f \tag{4}$$

Now, we have to look for constants $c_1, c_2, \cdots, c_{n-1}, c_n \in \{0, 1\}$ with:

$$\begin{bmatrix} d_1^f & d_2^f & \cdots & d_{n-1}^f & d_n^f \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = S_B \oplus S_B^{\widetilde{f}} \tag{5}$$

The matrix $[d_1^f \; d_2^f \; \cdots \; d_n^f]$ can be precomputed for any unconditional stuck-at fault *f*, the correct signature S_B after pattern block *B* can be precomputed as well, and $S_B^{\widetilde{f}}$ is the observed faulty signature. Hence, equation (5) contains at least *n* equations with *n* unknowns. If equation (5) is solvable for a stuck-at fault *f*, $[c_1, c_2, \cdots, c_{n-1}, c_n]$ describes the fault conditions for the patterns $[p_1, p_2, \cdots, p_n]$, otherwise the fault location of *f* cannot be the single culprit. This approach only requires the solution of a system of linear equations after the fault simulation step usually employed for logic diagnosis.

D. Fault Diagnosis

For each pattern block B, let $\widetilde{B}$ be the set of faults that can fully explain the observed faulty behavior in $S_B^{\widetilde{f}}$, that is:

$$\widetilde{B} := \{f \,|\, \text{equation (5) is solvable for } f\}.$$

The method described in section III-C provides for each block B such a set of suspect locations (faults). The number of blocks B with $f \in \widetilde{B}$ is a measure of the fault's *evidence*. That is, the higher this number is, the more likely f is in fact the real cause of the defect behavior. With this criteria, a ranked fault list can be created for logic diagnosis as follows. Let F be the set of conditional stuck-at faults.

A mapping $evidence : F \rightarrow \mathbb{N}_0$ is defined as

$$evidence(f) := |\{B | f \in \widetilde{B}\}| \tag{6}$$

The faults f_i are ordered due to decreasing values of $evidence(f_i)$ and each fault f_i is assigned a rank, which is its position in the resulting ordered list. The fail memory contains at most g fault signatures, and there may be blocks in between providing correct signatures. If there are two faults f_1 and f_2 with $evidence(f_1) = evidence(f_2)$ we order these two faults by using the correct signatures. Let f_1' and f_2' be the unconditional counterparts of f_1 and f_2, let $\hat{B}$ be the set of blocks which provided a correct signature until the fail memory was full. We assign a fault f_1 a higher priority, if the corresponding unconditional stuck-at fault f_1' would also lead to a correct signature. In a formal way, we set

$$\begin{aligned} a_1 &:= |\{S_B^{f_1'} | S_B^{f_1'} = S_B \text{ for } B \in \hat{B}\}| && (7) \\ a_2 &:= |\{S_B^{f_2'} | S_B^{f_2'} = S_B \text{ for } B \in \hat{B}\}| && (8) \end{aligned}$$

and rank f_1 higher than f_2 if $a_1 > a_2$. The faults f_i with the highest rank are most likely culprits of the defective behavior.

To exemplify this, assume a test set was divided into four pattern blocks, B_0 provided a correct signature, and the three remaining sets $\widetilde{B}$ looked as follows:

$$\begin{aligned} \widetilde{B}_1 &= \{f_1\} && (9) \\ \widetilde{B}_2 &= \{f_2, f_1, f_3\} && (10) \\ \widetilde{B}_3 &= \{f_2\}. && (11) \end{aligned}$$

Hence, $evidence(f_1) = 2$, $evidence(f_2) = 2$ and $evidence(f_3) = 1$. If the unconditional f_2' led to a correct signature in B_0, but f_1' did not, we would rank f_2 before f_1.

E. Example

Figure 3 shows a piece of circuitry whose response is compacted by a four-bit LFSR with generator polynomial $X^4 + X^3 + 1$.

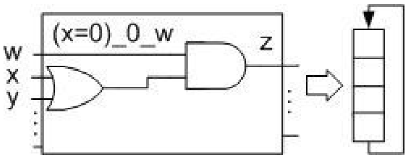

Fig. 3. Example circuit

Let us assume there is a wired-and fault between the lines w and x. This failure behavior can be modeled as the conditional stuck-at fault $\widetilde{f} = (x = 0)\_0\_w$. Let us further assume a signature block size of four patterns and that the observed response from the fail memory equals $S_B^{\widetilde{f}} = [0\ 1\ 0\ 0]^t$.

The patterns in a given pattern block are:

$$B = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix} \right\}$$

For the sake of simplicity, let us consider only the candidate fault $f = s@0\_w$. The signatures s_i and s_i^f are found by simulating the fault-free and faulty circuit response.

$$s_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} s_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} s_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} s_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$s_0^f = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} s_1^f = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} s_2^f = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} s_3^f = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

By applying equations (3) and (2) we find the error signatures and fault-free block signature, respectively. For simplification, we assume $m = 1$ and, therefore, $L = H$

$$e_0^f = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} e_1^f = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} e_2^f = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} e_3^f = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} ; S_B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Now, equation (4) gives:

$$d_0^f = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} d_1^f = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} d_2^f = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} d_3^f = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

From equation (5), a system of linear equations is derived:

$$\begin{aligned} c_2 &= 1 \\ c_1 + c_3 + c_4 &= 1 \\ c_1 + c_3 &= 0 \\ c_1 + c_2 &= 1 \end{aligned}$$

The equation system has the solution: $c_1 = 0$, $c_2 = 1$, $c_3 = 0$, $c_4 = 1$. This means that f can exactly match the observed signature of $\widetilde{f}$ if and only if f is activated only in patterns 2 and 4. A similar procedure can be applied to every fault in the circuit.

IV. Experimental results

A. Experimental setup

In order to analyze the efficiency of the proposed BISD scheme, several experiments with industrial circuits have been conducted. These circuits were kindly provided by NXP. The results in this section quantify the performance of the BISD technique in terms of hardware overhead, fault coverage and diagnosis resolution. Table I shows the characteristics of the circuits used in the experiments. The first column shows the circuit name, the second and third columns show the number of gates and the number of pseudo-primary outputs, respectively. The fourth and fifth columns show k, the scan chain count, and m, the length of the longest scan chain. The last column shows the number of collapsed stuck-at faults in the circuit.

Table II shows the size of the test pattern set for each target circuit along with its corresponding fault coverage.

Circuit	#gates	#PPO	k	m	stuck-at faults
p100k	84356	5829	270	53	162129
p141k	152808	10502	264	45	283548
p239k	224597	18495	260	61	455992
p259k	298796	18495	360	61	607536
p267k	239687	16621	260	62	366871
p269k	239771	16621	360	62	371209
p279k	257736	17835	385	59	493844
p286k	332726	17835	385	60	648044
p295k	249747	18521	330	62	472124
p330k	312666	17468	320	64	540758
p378k	341315	17420	325	64	816534

TABLE I. Circuit characteristics

design	# p	fc
p100k	5397	99.56%
p141k	5642	98.86%
p239k	4778	98.84%
p259k	4919	99.10%
p267k	5191	99.60%
p269k	5164	99.60%
p279k	5360	97.89%
p286k	6224	98.34%
p295k	7916	99.15%
p330k	9165	98.95%
p378k	664	100.00%

TABLE II. Pattern set and fault coverage

B. BISD hardware overhead

In order to quantify the additional memory required for the implementation of the proposed BISD scheme, the size of the response and fail memory is compared to the memory requirement for input pattern storage. The method presented in [22] is chosen for this comparison, as it is one of the most efficient pattern encoding schemes available to date. The first column in Table III shows the circuit name, while the second column shows the required storage for test pattern generation taken from [22]. Columns three and four show the relative storage overhead when 4 and 8 patterns are compacted into a signature.

Design	Input ([22])[kB]	4 patterns Inc. (%)	8 patterns Inc. (%)
p100k	7.25	78.32	41.1
p141k	36.18	16.16	8.35
p239k	17.97	27.7	14.41
p259k	23.54	21.75	11.3
p267k	47.95	11.24	5.83
p269k	47.44	11.31	5.86
p279k	48.37	11.49	5.95
p286k	63.69	10.09	5.2
p295k	–	–	–
p330k	76.56	10.60	5.43
p378k	–	–	–

TABLE III. Additional requirements for fail and response memory.

As Table III shows, the BIST memory increases on average by 22.1% and 11.5% for 4-pattern and 8-pattern signatures, respectively. The figures for the circuits p295k and p378k cannot be estimated as their required BIST memory is not available in [22].

The memory requirements for circuit p100k are significant. This, however, results from the well compressible input patterns and not from an insufficient response compaction. For all other circuits, the overhead is negligible compared to that of the seed memory on the input side. For 4-pattern compaction the memory sizes are already in the order of magnitude of that in [8]. For eight pattern compaction the memory increase drops even below 10% for the larger circuits.

C. Diagnostic resolution

In order to evaluate the achievable diagnostic resolution of the proposed compaction method, a total of 400 faults: 100 stuck-at faults, 100 crosstalk faults, 100 delay and 100 wired-and faults were randomly and uniformly injected into each circuit. In these experiments a fault is said to be correctly diagnosed if it is the single most likely fault candidate at the top of the ranking list after the responses in the fail memory have been analyzed.

A 32-bit LFSR was chosen for the BISD architecture with pattern blocks of 4 and 8 patterns. The depth of the fail memory was set to 50.

Table IV shows the detailed results of the diagnosis experiments. Columns two to eight show the diagnostic resolution for diagnosis with multiple-pattern signatures. The last four columns show the diagnostic resolution for the circuit in bypass mode (i.e. original circuit in external test). For some fault models (wired-and and stuck-at) the diagnostic resolution even increases for multiple-pattern compaction. This behavior results from the diagnostic quality of the test set and the fixed depth of the fail memory: when more patterns are compacted into a signature a larger portion of the test set can be analyzed before the fail memory becomes full.

Table V shows the diagnostic resolution for each circuit averaged over all injected faults. Columns two to five show the diagnostic average and its difference from the diagnostic resolution in bypass mode, which is shown in column six. The bypass mode corresponds to the original circuits for which the diagnostic resolution appears to be low (on average about 70%). There are three reasons for this: first, the test sets generated by commercial tools are not diagnostic test sets and are not optimized for a high resolution. Second, the test sets only target stuck-at faults and do not activate non-target faults very often. Finally, the metric used here for resolution is conservative as only the placement at the top of a ranked fault list is considered as a success.

For four pattern compaction the BISD scheme has on average no influence on the diagnostic resolution. For 8-pattern compaction, which yields a significant decrease in data volume, the loss in resolution is negligible.

In order to assess the computational effort required for the proposed diagnosis method, Table VI shows the execution times for the largest of the targeted circuits. As the table shows, the diagnosis of a given fault takes, on average, little under 1 minute when 4 patterns comprise the final signature, and under 2 and a half minutes when 8 patterns form one signature.

V. Conclusion

A Built-In Self-Diagnosis scheme for random logic circuits based on the standard STUMPS architecture has been proposed. Several test patterns are compacted into intermediate re-

	4 patterns				8 patterns				bypass			
Circuit	Stuck	Cross	Delay	Wired-And	Stuck	Cross	Delay	Wired-And	Stuck	Cross	Delay	Wired-And
p100k	73%	68%	74%	75%	70%	67%	72%	76%	70%	68%	76%	75%
p141k	85%	62%	79%	71%	81%	62%	78%	71%	83%	61%	79%	67%
p239k	82%	76%	85%	82%	84%	74%	82%	82%	80%	76%	85%	82%
p259k	78%	68%	82%	76%	78%	68%	79%	75%	78%	70%	82%	77%
p267k	79%	61%	75%	74%	77%	61%	71%	71%	79%	63%	70%	69%
p269k	75%	65%	72%	76%	72%	62%	71%	74%	72%	66%	74%	75%
p279k	70%	60%	70%	70%	68%	60%	68%	71%	67%	60%	73%	67%
p286k	75%	56%	67%	68%	76%	57%	64%	68%	76%	56%	67%	68%
p295k	59%	42%	46%	51%	67%	41%	44%	52%	66%	45%	47%	54%
p330k	71%	67%	70%	71%	71%	65%	67%	71%	71%	65%	72%	71%
p378k	87%	91%	95%	93%	87%	93%	95%	93%	87%	91%	95%	93%

TABLE IV. Diagnostic resolution

	4 patterns		8 patterns		bypass
Circuit	Avg	*delta*	Avg	*delta*	
p100k	72.5%	+0.2	71.2%	–1.0	72.2%
p141k	74.2%	+1.8	73.0%	+0.5	72.5%
p239k	81.2%	+0.5	80.5%	–0.2	80.8%
p259k	76.0%	–0.8	75.0%	–1.8	76.8%
p267k	72.2%	+2.0	70.0%	–0.2	70.2%
p269k	72.0%	+0.2	69.8%	–2.0	71.8%
p279k	67.5%	+0.8	66.8%	0.0	66.8%
p286k	66.5%	–0.2	66.2%	–0.5	66.8%
p295k	49.5%	–3.5	51.0%	–2.0	53.0%
p330k	69.8%	0.0	68.5%	–1.2	69.8%
p378k	91.5%	0.0	92.0%	+0.5	91.5%

TABLE V. Average diagnostic resolution

Circuit	4 patterns [s]	8 patterns [s]
p295k	44.7	140.0
p330k	53.1	108.0
p378k	53.7	76.6

TABLE VI. Execution times

sponse signatures and are compared to their corresponding reference signatures stored on-chip. The scheme is accompanied by a dedicated diagnosis routine, which takes multiple-pattern compaction into account. The scheme can be implemented with only minimal modifications to the available design-for-test infrastructure and with insignificant storage overhead.

Acknowledgment

This work has been supported by the BMBF within the project DIANA in collaboration with Audi AG and Infineon Technologies AG.

References

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

[2] P. Bardell and W. McAnney, "Self-testing of multichip logic modules." in *Proceedings of the IEEE International Test Conference*, 1982, pp. 200–204.

[3] I. Bayraktaroglu and A. Orailoglu, "Gate level fault diagnosis in scan-based BIST." in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2002, pp. 376–381.

[4] Y. Nakamura, T. Clouqueur, K. 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 Systems*, vol. 15, no. 7, pp. 790–800, 2007.

[5] J. Ghosh-Dastidar and N. A. Touba, "A rapid and scalable diagnosis scheme for BIST environments with a large number of scan chains," in *Proceedings of the 18th IEEE VLSI Test Symposium*, 2000, pp. 79–85.

[6] J. Rajski and J. Tyszer, "Diagnosis of scan cells in BIST environment," *IEEE Transactions on Computers*, vol. 48, no. 7, pp. 724–731, 1999.

[7] P. Wohl, J. A. Waicukauski, S. Patel, and G. Maston, "Effective diagnostics through interval unloads in a BIST environment," in *Proceedings of the 39th Conference on Design Automation*, 2002, pp. 249–254.

[8] M. Elm and H.-J. Wunderlich, "BISD: Scan-based built-in self-diagnosis," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2010, pp. 1243 –1248.

[9] 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 and Exhibition*, 2004, pp. 1302–1307.

[10] 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*, 2002, pp. 382–386.

[11] 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. 62–634, 2005.

[12] N. Touba, "X-canceling MISR: An X-tolerant methodology for compacting output responses with unknowns using a MISR," in *Proceedings of the IEEE International Test Conference*, 2007, pp. 1 –10.

[13] 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, no. 3, pp. 421 – 432, 2004.

[14] 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, pp. 83–89.

[15] 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*, 2003, pp. 107–112.

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

[17] J. Waicukauski and E. Lindbloom, "Failure diagnosis of structured VLSI," *IEEE Design & Test of Computers*, vol. 6, no. 4, pp. 49–60, 1989.

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

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

[20] 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*, 2006, pp. 1 –9.

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

[22] A.-W. Hakmi, S. Holst, H.-J. Wunderlich, J. Schloffel, F. Hapke, and A. Glowatz, "Restrict encoding for mixed-mode BIST," in *Proceedings of the 27th IEEE VLSI Test Symposium*, 2009, pp. 179–184.