Built-in Self-Diagnosis Exploiting Strong Diagnostic Windows in Mixed-Mode Test

Cook, Alejandro; Hellebrand, Sybille; Wunderlich, Hans-Joachim

Proceedings of the 17th IEEE European Test Symposium (ETS'12) Annecy, France, 28 May-1 June 2012

doi: http://dx.doi.org/10.1109/ETS.2012.6233025

Abstract: Efficient diagnosis procedures are crucial both for volume and for in-field diagnosis. In either case the underlying test strategy should provide a high coverage of realistic fault mechanisms and support a low-cost implementation. Built-in self-diagnosis (BISD) is a promising solution, if the diagnosis procedure is fully in line with the test flow. However, most known BISD schemes require multiple test runs or modifications of the standard scan-based test infrastructure. Some recent schemes circumvent these problems, but they focus on deterministic patterns to limit the storage requirements for diagnostic data. Thus, they cannot exploit the benefits of a mixed-mode test such as high coverage of non-target faults and reduced test data storage. This paper proposes a BISD scheme using mixed-mode patterns and partitioning the test sequence into "weak" and "strong" diagnostic windows, which are treated differently during diagnosis. As the experimental results show, this improves the coverage of non-target faults and enhances the diagnostic resolution compared to state-of-the-art approaches. At the same time the overall storage overhead for input and response data is considerably reduced.

Built-in Self-Diagnosis Exploiting Strong Diagnostic Windows in Mixed-Mode Test

Alejandro Cook[1], Sybille Hellebrand[2], Hans-Joachim Wunderlich[1]
[1]Institute of Computer Architecture and Computer Engineering, University of Stuttgart, Germany
email: cook@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de
[2]Institute of Electrical Engineering and Information Technology, University of Paderborn, Germany
email: sybille.hellebrand@uni-paderborn.de

Abstract— **Efficient diagnosis procedures are crucial both for volume and for in-field diagnosis. In either case the underlying test strategy should provide a high coverage of realistic fault mechanisms and support a low-cost implementation. Built-in self-diagnosis (BISD) is a promising solution, if the diagnosis procedure is fully in line with the test flow. However, most known BISD schemes require multiple test runs or modifications of the standard scan-based test infrastructure. Some recent schemes circumvent these problems, but they focus on deterministic patterns to limit the storage requirements for diagnostic data. Thus, they cannot exploit the benefits of a mixed-mode test such as high coverage of non-target faults and reduced test data storage.**

This paper proposes a BISD scheme using mixed-mode patterns and partitioning the test sequence into "weak" and "strong" diagnostic windows, which are treated differently during diagnosis. As the experimental results show, this improves the coverage of non-target faults and enhances the diagnostic resolution compared to state-of-the-art approaches. At the same time the overall storage overhead for input and response data is considerably reduced.

Keywords: Built-in Diagnosis, Design for Diagnosis

I. INTRODUCTION

A seamless interplay between efficient test and diagnosis techniques is the key to a fast yield ramp-up of the manufacturing process. Also for in-field diagnosis a fine-grained chip-level diagnosis is of growing importance. In a recent research project funded by the German Government, for example, companies of the complete supply chain work together to improve the analytic and diagnostic capabilities of electronic control units (ECUs) in automobiles down to chip and logic level [8]. Ideally, a single test run should provide enough information to analyze failures with a high diagnostic resolution.

With respect to testing, built-in self-test (BIST) offers many advantages, such as low cost test application and reuse of the test infrastructure throughout all phases of the system life cycle. While pure pseudo-random BIST, also referred to as LBIST, may be affected by insufficient fault coverage, a fully deterministic test may require too much storage, even if the test patterns are highly compressed. Mixed-mode BIST combines the advantages of pseudo-random and deterministic test. Here, pseudo-random patterns are used to cover most of the faults, including non-target faults, and deterministic patterns are needed only for the hard to test faults. Many powerful approaches for mixed-mode BIST have been proposed, most of them reusing the pseudo-random pattern generator as part of the decompressor for encoded deterministic test data. Examples can be found in [14][15][16][20][21][22][26][29][30][33].

Concerning diagnosis, most of the schemes for built-in diagnosis follow a multi pass strategy. The test configuration is changed during repeated test runs to identify the failing scan cells or vectors in a first step [5][13][23][24][27][32], then known algorithms for logic diagnosis are applied to find the actual fault location [1][2][11][31]. Smart compaction schemes allow a more efficient identification of failing scan cells during embedded test, but here the output data are evaluated for each pattern or even for each scan cycle [19][25][28]. In direct diagnosis, the fault location in the logic is directly determined from a compacted test response [6][7]. Signature-based diagnosis, for example, computes a MISR signature for each pattern and performs diagnosis directly on the failing signatures [7]. However, as a reference signature is needed for each test pattern, this scheme is not suitable for an autonomous built-in self-diagnosis (BISD).

To reduce the amount of reference data to be stored, the XP-SISR scheme introduces extreme space compaction before generating the signatures [12]. In contrast to that in [9][10] a window-based diagnosis is proposed, where signatures are computed and analyzed for contiguous subsequences of several patterns. Both the XP-SISR and the window-based diagnosis scheme work with considerably reduced response data and support a fully autonomous BISD in a single test run. Nevertheless, combining these schemes with mixed-mode BIST is not straightforward, because long sequences of pseudo-random patterns would drive the storage requirements for the response data beyond acceptable limits. Therefore these schemes are implemented using rather short pseudo-random sequences only, and the pseudo-random test is treated in the same way as the deterministic test. This results in a two-fold hardware penalty. On the one hand, the test data storage for the deterministic patterns is larger than necessary, and on the other hand, the response data for the pseudo-random sequence grow extremely high even for a limited number of patterns.

A recent approach for volume diagnosis in conjunction with LBIST addresses this problem by working only on a sample of the pseudo-random patterns [3]. But failures outside the sample space must be analyzed in a second pass.

This paper presents a new approach to combine mixed-mode BIST with a fully autonomous BISD. The test is partitioned into "strong" and "weak" diagnostic windows, which are treated differently. Experimental results show that this way the overall storage requirements for deterministic test data and for response data can be minimized while keeping high fault coverage and providing a high diagnostic resolution in a single test run.

The rest of the paper is structured as follows. Section II summarizes the basic ideas of the underlying diagnostic procedure for deterministic patterns. Subsequently, the new approach for mixed-mode diagnosis is introduced in Section III. Finally, Section IV presents an experimental study validating the new approach.

II. Background

The new scheme for mixed-mode diagnosis uses the window-based diagnosis proposed in [9][10], which is fully compatible with the STUMPS architecture [4]. The test is partitioned into windows $W_1, ..., W_N$, and for each window a cumulative signature is computed and compacted as shown in Figure 1.

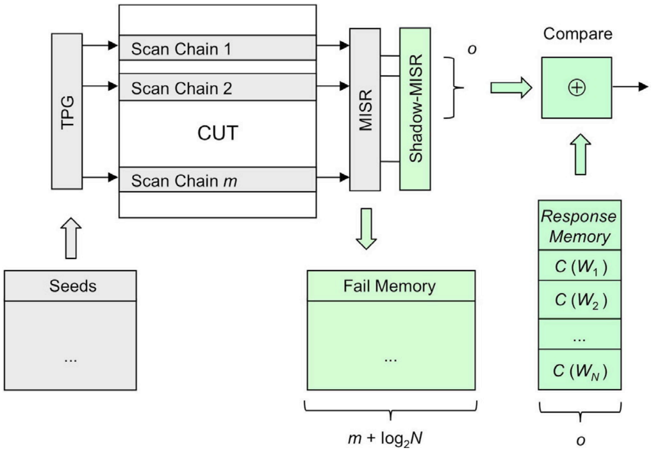

Figure 1: Window-based diagnosis with compacted signatures.

At the end of each window the obtained MISR signature is copied to a shadow-MISR. While the test continues normally with the next window, the shadow MISR runs in autonomous mode as long as the first pattern is applied. As demonstrated in [10], this way fault effects are distributed over the MISR randomly, and it is sufficient to observe only o bits of the shadow-MISR. The observed bits are compared to the respective reference data stored in the response memory. If a mismatch is detected, the complete signature is stored in the fail memory together with the index of the test window.

The diagnostic analysis of the fail memory is based on the conditional stuck-at fault model [17]. For each line v, the conditional stuck-at faults *cond*_0_v and *cond*_1_v are considered, where the condition *cond* describes arbitrary Boolean or timing properties. If *cond* is met, the fault is active and the line is forced to either 0 or 1. For instance, $(v_{-1}=0 \wedge v=1)$_0_v describes a slow-to-rise fault. To deal with complex defect mechanisms a relationship between observed fault effects and activating patterns can be established in *cond*. For a pattern sequence $P = (p_1, ..., p_n)$ and a circuit line v, the conditional stuck-at fault $(p_i \mid P)$_0_v sets v to '0' as long as p_i is active in the pattern sequence P. In the same way, the conditional stuck-at fault $(p_i \mid P)$_1_v sets v to '1' when the pattern p_i occurs in P.

As an entry in the fail memory corresponds to an observed signature $S_{obs}(W)$ for a window $W = (p_1, \dots, p_n)$, the conditional stuck-at faults $(p_1 \mid W)$_0_v, …, $(p_n \mid W)$_0_v and $(p_1 \mid W)$_1_v, …, $(p_n \mid W)$_1_v are analyzed for each circuit line v. The impact of the individual faults on the reference signature can be pre-computed and stored, such that the final diagnosis step consists in finding all combinations of faults which can explain the difference $S_{obs}(W) \oplus S_{ref}(W)$ to the full reference signature. This corresponds to solving a system of linear equations, and if a solution is found for a given line v, this line is identified as a candidate fault location. The number of windows, in which line v is a candidate fault location, is used as a measure of evidence. The higher the evidence score is, the more likely a fault at line v is in fact the real cause of the failure, and is consequently ranked higher in the candidate list. If two faults explain the same number of windows, then the fault sensitized less often is considered the more likely candidate.

To have a unique solution for the mentioned systems of equations, the number of variables should be less than or equal to the number of equations, which implies that the number of patterns in a window should be less than or equal to the number of bits in the MISR signature. Therefore this approach is suitable for deterministic test or mixed-mode test with short pseudo-random sequences. Applying it directly to long pseudo-random sequences would result in an extremely high volume of response data.

III. New Approach

The benefit of pseudo-random patterns in mixed-mode BIST is two-fold. On the one hand, they reduce the storage requirements for deterministic test data, and on the other hand, the potential coverage of non-target faults improves the test quality. To retain these advantages, the diagnosis approach presented in this section addresses mixed-mode patterns, which contain long pseudo-random sequences. Clearly, the key challenge is keeping the size of the response memory feasible without dropping important information on non-target faults in the pseudo-random sequence. The main idea to achieve this goal is to identify regions of the pseudo-random test that provide essential diagnostic information. As a heuristic measure the detection profile for a given set of target faults F is used. "Strong" windows are determined, such that each target fault is covered by at least one "strong" window. The algorithm consists of a test generation, a partitioning, and the actual diagnosis step.

A. Test Generation

To generate a mixed-mode test with r pseudo-random patterns, first r patterns produced by the pseudo-random pattern generator of the target BIST architecture are fault simulated against a set of target faults F. The generated pseudo-random test is denoted by $T_{rand}(r)$, and the subset of detected faults by $F_{rand}(r)$. Then ATPG patterns are determined for the remaining hard to detect faults $F_{hard}(r)$. The obtained deterministic test set is denoted by $T_{det}(r)$.

B. Partitioning

The basic procedure to partition the test into "strong" and "weak" diagnostic windows works as follows.

(a) In a first step the "essential" random patterns are extracted from $T_{rand}(r)$.

Definition: A subset $E_{rand}(r)$ in $T_{rand}(r)$ is called essential, if the patterns in $E_{rand}(r)$ detect all faults in $F_{rand}(r)$ and $E_{rand}(r)$ is a set of minimum cardinality with this property.

As computing $E_{rand}(r)$ corresponds to solving a complex covering problem, the set of essential random patterns is approximated during fault simulation, using only those patterns of $T_{hard}(r)$ which detect a new fault.

(b) Each essential pattern in $E_{rand}(r)$ defines a strong diagnostic window of length $n = 2^k$. To simplify control, the strong windows are positioned around the essential patterns, such that the starting index of a window is a multiple of 2^k (see Figure 2). I.e. if t_j is an essential pattern, then the window around t_j starts with pattern t_i, where $i = 2^k \cdot (j \text{ div } 2^k)$, and t_j has position $j \bmod 2^k$ in the window.

(c) To keep as much diagnostic information as possible on non-target faults, all random patterns between two strong diagnostic windows are treated as one "weak" diagnostic window. Thus, the size of a weak window can be a multiple of 2^k, and a signature is generated only at the end of a weak diagnostic window.

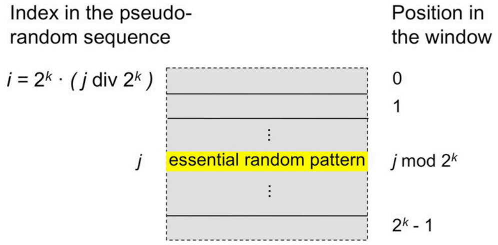

Figure 2: Strong diagnostic window.

(d) Finally, the deterministic patterns $T_{det}(r)$ are divided into diagnostic windows of size 2^k, which are also referred to as strong windows. Overall, the test is then structured as shown in Figure 3.

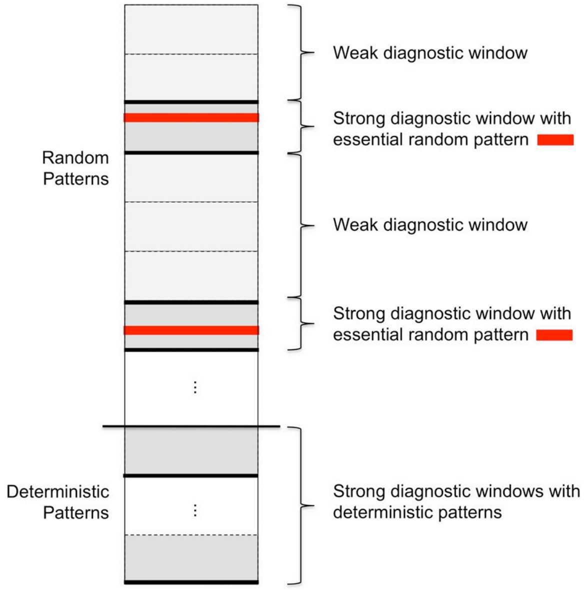
Weak diagnostic window
Strong diagnostic window with essential random pattern
Random Patterns
Weak diagnostic window
Strong diagnostic window with essential random pattern
Deterministic Patterns
Strong diagnostic windows with deterministic patterns

Figure 3: Aligned diagnostic windows.

To minimize the number of strong windows, the number of essential random and deterministic patterns can be further reduced by reverse order fault simulation of the patterns in $E_{rand}(r) \cup T_{det}(r)$ against F. Deleting unnecessary patterns provides reduced sets $E^*_{rand}(r)$ and $T^*_{\det}(r)$ as the basis for the strong windows. The overall partitioning flow then proceeds as shown in Figure 4.

The size of the response memory is determined not only by the number of strong windows, but also by their distribution in the pseudo-random sequence. As the weak windows can have varying size, additional control information is necessary to identify the beginning of weak and strong windows. If the number of strong diagnostic windows is large, this information is provided by tagging all patterns in the pseudo-random sequence with index 0 mod 2^k appropriately with one extra bit. If only few strong diagnostic windows are needed, alternatively the index of the starting pattern (div 2^k) can be stored with the compacted signature in the response memory.

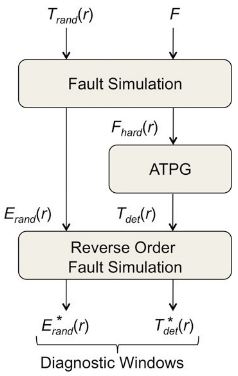

Figure 4. Partitioning the test into diagnostic windows.

C. Diagnosis

If faulty signatures are observed at the end of both weak and strong diagnostic windows, then the diagnosis procedure of Section II is applied to the strong windows only. Fault simulation of the weak windows is used to validate the results. If several candidate fault locations can explain the faulty behavior with the same probability, then the reference signatures of the weak windows are analyzed in more detail to improve the diagnostic resolution. For this purpose the conditional stuck-at faults $(p_1 \mid W)$_0_v, …, $(p_n \mid W)$_0_v and $(p_1 \mid W)$_1_v, …, $(p_n \mid W)$_1_v are simulated for all affected weak windows W and all candidate fault locations v identified by the analysis of the strong windows.

If a faulty signature appears only at the end of a weak window, then the direct diagnosis procedure described in [18] is applied (or any other direct diagnosis procedure that works on larger windows).

IV. Experimental Results

Several experiments with industrial circuits have been performed to validate the mixed-mode diagnosis approach proposed in Section III. The relevant characteristics of the circuits, kindly provided by NXP, are listed in Table I.

TABLE I. CIRCUIT CHARACTERISTICS

Circuit	*#Gates*	*#PPO*	*Scan Chains*	*Max. Length*	*# 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

The first column shows the circuit name, and columns two to five indicate the number of gates, the number of pseudo-primary outputs, the number of scan chains, and the length of the longest scan chain. The last column shows the number of collapsed stuck-at faults.

In all experiments reported below, the set of target faults F comprises all stuck-at faults. Furthermore, 32-bit MISRs are used, and the test is partitioned into blocks of n = 32 patterns. Consequently, strong diagnostic windows always contain 32 patterns, and weak diagnostic windows contain a multiple of 32 patterns.

To analyze the impact of the pseudo-random patterns on the test data storage, a mixed-mode BIST with 4096 pseudo-random patterns as used in [10] is compared to one with 100000 pseudo-random patterns. Table II shows the achieved fault coverage after applying the complete test and the characteristics of the deterministic test set $T^*_{\text{det}}(r)$ for r = 4096 and r = 100000 pseudo-random patterns.

TABLE II. COMPARISON OF INPUT DATA

Circuit	*Fault Coverage*	$T^*_{\text{det}}(4096)$		$T^*_{\text{det}}(100000)$	
		Patterns	*Spec. Bits*	*Patterns*	*Spec. Bits*
p100k	99.56%	414	60539	118	18843
p141k	98.85%	704	375597	494	254410
p239k	98.84%	618	162221	431	102537
p259k	99.10%	830	221598	523	121104
p267k	99.60%	678	393979	578	325800
p269k	99.58%	693	396025	533	322150
p279k	97.89%	917	397743	668	279820
p286k	98.33%	1511	568528	935	365475
p295k	99.15%	2553	579146	748	362839
p330k	98.95%	5587	986122	5191	866846

The fault coverage is identical in both cases, as the same commercial ATPG tool was used to detect as many faults in $F_{hard}(r)$ as possible. For the respective deterministic test sets both the number of patterns and the overall number of specified bits are provided. As the table shows, for all circuits, the use of a larger number of inexpensive pseudo-random patterns decreases the overall number of specified bits and thus the expected size of the seed memory.

Concerning diagnosis, the test with the short pseudo-random sequence is treated in the same way as in [10], i.e. the complete test is partitioned into 32-pattern windows. For the test with the long pseudo-random sequence, the strong and weak diagnostic windows are identified according to the method presented in Section III. Table III shows the results when 8 bits of the MISR are observed (o = 8) as recommended in [10].

For each window, 8 bits of the reference signature have to be stored in the response memory. So the number of windows corresponds to the reference data in bytes. Additionally, for the proposed approach each 32-pattern block must be tagged with one extra bit to identify the starting positions of strong windows. Therefore, the amount of control data in bytes is given by the overall number of patterns divided by $32 \cdot 8$.

TABLE III. COMPARISON OF DIAGNOSTIC DATA ($o = 8$)

Circuit	*Response Memory for r = 4096 [Byte]*	*Response Memory using Weak and Strong Windows for r = 100000*			
		# Strong Windows	# Weak Windows	Control Data [Byte]	Overall Cost [Byte]
p100k	141	1279	470	391	2140
p141k	150	1260	526	391	2177
p239k	148	1405	524	391	2320
p259k	154	1694	541	391	2626
p267k	150	1325	496	391	2212
p269k	150	1350	477	391	2218
p279k	157	1776	593	391	2760
p286k	176	2182	517	391	3090
p295k	208	2575	415	391	3381
p330k	303	1215	508	391	2114

The response memory for the mixed mode BIST grows with the length of the pseudo-random sequence. However, this effect is overcompensated by the savings for the seed memory. As test pattern compression is out of the scope of this paper, the size of the seed memory is estimated by the overall number of specified bits. The results are summarized in Table IV.

TABLE IV. COMPARISON OF OVERALL COST [BYTE]

Circuit	*Mixed-Mode BIST with r = 4096*			*Mixed-Mode BIST with r = 100000*		
	SM	RM	Σ	SM	RM	Σ
p100k	7568	141	7709	2356	2140	4496
p141k	46950	150	47100	31814	2177	33991
p239k	20278	148	20426	12818	2320	15138
p259k	27700	154	27854	15138	2626	17764
p267k	49248	150	49398	40725	2212	42937
p269k	49504	150	49654	40269	2218	42487
p279k	49718	157	49875	34978	2760	37738
p286k	71066	176	71242	45685	3090	48775
p295k	72394	208	72602	45355	3381	48736
p330k	123266	303	123569	108356	2114	110470

Here SM denotes the estimated size of the seed memory, RM the size of the response memory, and Σ the overall size of seed and response memory. All entries are given in bytes. Comparing the overall memory requirements, it can be observed, that the mixed-mode BIST with 100000 patterns reduces the overall cost for all cases.

In order to analyze the diagnostic accuracy, a total of 400 faults have been randomly and uniformly injected into each circuit. The fault set consists of 100 stuck-at faults, 100 crosstalk faults, 100 delay, and 100 wired-AND faults. The depth of the fail memory has been set to 100. That is, a maximum of 100 faulty responses, either from strong or weak windows can be stored in the fail memory and subsequently used for logic diagnosis. A fault is considered as correctly diagnosed, if it is one of the top 5 fault candidates in the ranked list after the responses in the fail memory have been analyzed. The diagnostic resolution is then computed as the percentage of faults that have been correctly diagnosed. Table V summarizes the results and compares them to the results achieved by the method proposed in [10]. Here 4096 pseudo-random patterns are used and the complete test is partitioned into windows of 32 patterns.

Table V shows that the proposed method slightly reduces the diagnostic for stuck-at faults in five cases (p100k, p269k, p295k, p286k, and p330k), which can be explained by the deletion of some of the stuck-at patterns. However, the diagnostic resolution for non-target faults is considerably improved, except in one case (delay faults in p286k).

Finally Table VI evaluates the impact of the weak windows. Column two lists the number of additionally detected non-target faults, while column 3 provides the number of faults which cannot be diagnosed correctly by analyzing the strong windows only. To achieve this improvement, the information obtained from the weak windows has been applied to the top 50 candidates identified by analyzing the strong windows only. Overall, the results show that the weak windows can both improve the defect coverage and diagnostic resolution.

TABLE V. DIAGNOSTIC RESOLUTION

Circuit	*Stuck-At*		*Crosstalk*		*Delay*		*Wired AND*	
	[10]	r = 100000	[10]	r = 100000	[10]	r = 100000	[10]	r = 100000
p100k	99 %	98 %	86 %	93 %	90 %	92 %	96 %	96 %
p141k	98 %	98 %	81 %	88 %	91 %	93 %	94 %	95 %
p239k	98 %	98 %	87 %	91 %	92 %	97 %	97 %	99 %
p259k	99 %	100 %	83 %	90 %	93 %	97 %	93 %	97 %
p267k	99 %	99 %	74 %	82 %	88 %	95 %	92 %	93 %
p269k	99 %	98 %	80 %	93 %	89 %	94 %	93 %	96 %
p279k	95 %	95 %	78 %	89 %	87 %	90 %	91 %	95 %
p286k	96 %	94 %	79 %	90 %	89 %	86 %	93 %	97 %
p295k	95 %	94 %	70 %	78 %	69 %	74 %	88 %	88 %
p330k	98 %	96 %	85 %	85 %	86 %	90 %	91 %	91 %

TABLE VI. IMPACT OF WEAK WINDOWS FOR R = 100000

Circuit	***Additionally Detected Non-Target Faults***	***# Faults with Improved Diagnosis***
p100k	3	1
p141k	3	4
p239k	2	4
p259k	-	-
p267k	3	6
p269k	-	4
p279k	-	3
p286k	-	-
p295k	-	4
p330k	3	1

V. CONCLUSIONS

Mixed-mode BIST provides an efficient solution for both manufacturing and in-field testing. Partitioning the pseudo-random part of the test into strong and weak windows allows the seamless integration of test and diagnosis into a fully autonomous BISD scheme, which improves the coverage of non-target faults and provides a high diagnostic resolution at reduced hardware cost.

ACKNOWLEDGMENT

This work has been funded by the DFG under contracts HE 1686/3-2 and WU 245/5-2.

REFERENCES

[1] M. Abramovici and M. A. Breuer, “Fault diagnosis based on effect cause analysis: An introduction,” Proc. 17th Design Automation Conf. (DAC’80), Minneapolis, MN, USA, 1980, pp. 69-76.

[2] M. E. Amyeen, D. Nayak, and S. Venkataraman, “Improving precision using mixed-level fault diagnosis,” Proc. IEEE Int. Test Conf. (ITC’06), Santa Clara, CA, USA, 2006, pp. 1-10.

[3] M. E. Amyeen et al., “Logic BIST Silicon Debug and Volume Diagnosis Methodology,” Proc. IEEE Int. Test Conf. (ITC’11), Anaheim, CA, USA, Sept. 2011, pp. 1-10

[4] P. H. Bardell and W. H. McAnney, “Self-Testing of Multichip Logic Modules,” Proc. IEEE Int. Test Conf. (ITC’82), Philadelphia, PA, USA, Nov. 1982, pp. 200-204.

[5] I. Bayraktaroglu and A. Orailoglu, “Gate Level Fault Diagnosis in Scan-Based BIST,” Proc. Design, Automation and Test in Europe (DATE’02), Paris, France, 2002, pp. 376-381.

[6] W.-T. Cheng et al., “Compactor Independent Direct Diagnosis,” Proc. Asian Test Symp. (ATS’04), Kenting, Taiwan, Nov. 2004, pp. 204-209

[7] W.-T. Cheng et al., “Signature based diagnosis for logic BIST,” Proc. IEEE Int. Test Conf. (ITC’06), Santa Clara, CA, USA, 2006, pp. 1-9

[8] P. Clarke, “Germany backs Infineon ECU research with $6.2 million,” EE Times, July 19, 2010, http://www.eetimes.com/electronics-news/4204682/Germany-backs-Infineon-ECU-research

[9] A. Cook, M. Elm, H.-J. Wunderlich, U. Abelein, “Structural In-Field Diagnosis for Random Logic Circuits,” Proc. Eur. Test Symp. (ETS’11), Trondheim, May 2011, pp. 111-116

[10] A. Cook, S. Hellebrand, T. Indlekofer, H.-J. Wunderlich, “Diagnostic Test of Robust Circuits,” Proc. Asian Test Symp. (ATS’11), New Delhi, India, November 2011.

[11] R. Desineni, O. Poku, and R. D. S. Blanton, “A logic diagnosis methodology for improved localization and extraction of accurate defect behavior,” Proc. IEEE Int. Test Conf. (ITC’06), Santa Clara, CA, USA, 2006, pp. 1-10.

[12] M. Elm and H.-J. Wunderlich, “BISD: Scan-Based Built-In Self-Diagnosis,” Proc. Design Automation and Test in Europe (DATE’10), Dresden, Germany, March 8-12, 2010, pp. 1243-1248.

[13] J. Ghosh-Dastidar and N. A. Touba, “A rapid and scalable diagnosis scheme for BIST environments with a large number of scan chains,” Proc. 18th IEEE VLSI Test Symposium (VTS’00), Montreal, Canada, 2000, pp. 79-85.

[14] A.-W. Hakmi et al., “Restrict encoding for mixed-mode BIST,” Proc. 27th IEEE VLSI Test Symposium (VTS’09), Santa Cruz, CA, USA, 2009, pp. 179-184.

[15] S. Hellebrand et al., “Built-in test for circuits with scan based on reseeding of multiple polynomial linear feedback shift registers,” IEEE Trans. Computers, Vol. 44, No. 2, February 1995, pp. 223-233.

[16] S. Hellebrand, H.-J. Wunderlich, and A. Hertwig, “Mixed-mode BIST using embedded processors,” Proc. IEEE Int. Test Conf. (ITC’96), Washington, DC, USA, Oct. 1996, pp. 195-204.

[17] S. Holst and H.-J. Wunderlich, “Adaptive debug and diagnosis without fault dictionaries,” Proc. 12th Eur. Test Symp. (ETS’07), Freiburg, Germany, 2007, pp. 7-12.

[18] S. Holst and H.-J. Wunderlich, “A Diagnosis Algorithm for Extreme Space Compaction,” Proc. Design, Automation and Test in Europe (DATE'09), Nice, France, April 20-24, 2009, pp. 1355-1360

[19] B. Keller and T. Bartenstein, “Use of MISRs for Compression and Diagnostics,” Proc. IEEE Int. Test Conf. (ITC’05), Austin, TX, USA, Nov. 2005, pp. 1-9

[20] B. Koenemann, “LFSR-coded test patterns for scan designs,” Proc. Eur. Test Conf. (ETC’91), Munich, Germany, 1991, pp. 237-242.

[21] B. Koenemann et al., “A SmartBIST variant with guaranteed encoding,” Proc. Asian Test Symp. (ATS’01), 2001, pp. 325-330.

[22] C. V. Krishna and N. A. Touba, “3-stage variable length continuous-flow scan vector decompression scheme,” Proc. IEEE VLSI Test Symp. (VTS’04), April 2004, Napa Valley, CA, USA, pp. 79-86.

[23] A. Leininger, M. Goessel, and P. Muhmenthaler, “Diagnosis of scan chains by use of a configurable signature register and error-correcting codes,” Proc. of the Design, Automation and Test in Europe (DATE’04), Paris, France, 2004, pp. 1302-1307.

[24] C. Liu, K. Chakrabarty, and M. Goessel, “An interval-based diagnosis scheme for identifying failing vectors in a scan-BIST environment,” Proc. Design, Automation and Test in Europe (DATE’02), Paris, France, 2002, pp. 382-386.

[25] S. Mitra and K. S. Kim, “X-Compact: An Efficient Response Compaction Technique,” IEEE Trans. on Computer-Aided Design of Circuits and Systems, Vol. 23, No. 3, March 2004, pp. 421-432.

[26] J. Rajski, J. Tyszer, and N. Zacharia, “Test data decompression for multiple scan designs with boundary scan,” IEEE Trans. Computers, Vol. 47, No. 11, Nov. 1998, pp. 1188-1200.

[27] J. Rajski and J. Tyszer, “Diagnosis of scan cells in BIST environment,” IEEE Trans. on Computers, Vol. 48, No. 7, July 1999, pp. 724-731.

[28] J. Rajski, J. Tyszer, C. Wang, and S. Reddy, “Finite memory test response compactors for embedded test applications,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 4, April 2005, pp. 62-634.

[29] N. A. Touba and E. J. McCluskey, “Altering a pseudo-random bit-sequence for scan-based BIST,” Proc. IEEE Int. Test Conf. (ITC’96), Washington, DC, USA, October 20-25, 1996, pp. 167-175.

[30] E. H. Volkerink and S. Mitra, “Efficient seed utilization for reseeding based compression,” Proc. IEEE VLSI Test Symp. (VTS’03), April 27 – May 1, 2003, Napa Valley, CA, USA, pp. 232-240.

[31] J. Waicukauski and E. Lindbloom, “Failure diagnosis of structured VLSI,” IEEE Design & Test of Computers, Vol. 6, No. 4, 1989, pp. 49-60.

[32] P. Wohl, J. A. Waicukauski, S. Patel, and G. Maston, “Effective diagnostics through interval unloads in a BIST environment,” Proc. 39th Design Automation Conf. (DAC’02), New Orleans, LA, USA, 2002, pp. 249-254.

[33] H.-J. Wunderlich and G. Kiefer, “Bit-flipping BIST,” Proc. Int. Conf. on Computer-Aided Design (ICCAD’96), San Jose, CA, USA, Nov. 10-14, 1996, pp. 337-343.