

# Diagnosis for Scan-Based BIST: Reaching Deep into the Signatures \*

Ismet Bayraktaroglu  
Computer Science & Engineering Department  
University of California, San Diego  
La Jolla, CA 92093  
ibayrakt@cs.ucsd.edu

Alex Orailoglu  
Computer Science & Engineering Department  
University of California, San Diego  
La Jolla, CA 92093  
alex@cs.ucsd.edu

## Abstract

*For partitioning-based diagnosis in a scan-based BIST environment, an exact analysis scheme, capable of identifying all scan cells that receive incorrect data, is proposed. In contrast to previously suggested approaches, the scheme we propose identifies all failing scan cells with no ambiguity whatsoever. Not only do we resolve failing scan cells unambiguously, but we do so at the earliest possible instance through reexamination of already computed signatures. Intensive utilization of this highly precise diagnostic state information leads to prognostic information regarding the usefulness of running upcoming tests which in turn leads to reductions in diagnosis time in excess of 30% compared to previous approaches.*

## 1. Introduction

Increasing complexity and performance requirements of designs necessitate high cost ATEs. Yet reliance on such costly ATEs can be relaxed significantly by adopting BIST. Not only does BIST test a chip with nothing but inexpensive ATE, but furthermore does so at speed. Unfortunately, the benefits of BIST are adulterated as it introduces highly challenging problems for the diagnostic process, increasingly all the more important as the move towards deep sub-micron technologies increases failure rates. Fortunately, recent results in scan-based BIST show that it can provide diagnostic information that can pinpoint the scan cells in which faults are manifested. Yet as the subsequent process of pinpointing the exact fault location information is typically a time-consuming surface scan process, it is highly desirable to provide a refined set of scan cells for further study. Certainly, it is desirable not to miss any scan cells on which faults are manifested; yet just as important, if not even more so, is the ability to provide nothing more than the scan cells that manifest faults. It turns out that the ability to provide exact information is highly challenging. The ability to do this and do it as rapidly as possible constitutes the focus of this paper.

The fundamental challenge with BIST-based diagnosis stems from the same aspect that constitutes its strength from a test perspective. The compact signature collected at the conclusion of numerous test applications provides substantial benefits for test but creates havoc with diagnosis. As it can only vouch for the health of the complete ensemble of scan cells, the full signature provides not much of a differentiation capability between scan cells. As usual, divide-and-conquer approaches can possibly be of use since they enable taking snapshots of groups of scan cells. It is conceivable that juxtaposing the various perspectives can help refine the set of scan cells in which faults are manifested. BIST hardware does provide an additional benefit in that it can be used to provide various partitions with no necessity of introducing any specialized partitioning hardware. Such techniques have been explored previously but with no conclusive resolution as to how to pinpoint the exact set of failing scan cells.

It may be observed that aggressive pruning of the diagnostic set hinges on the application of large number of tests. While this is indeed true for current approaches, which achieve typical refinements of the set at inordinate time, we break rank with previous approaches by relying instead on insightful utilization of information regarding the *incorrectness* of a signature contained in a partition instead. We provide methods for identifying cells in which faults are manifested as soon as all ambiguity is resolvable. We compound this breakthrough by introducing methods for utilizing the current diagnostic state to not only reduce subsequent test application cost but furthermore to extract additional information from signatures already captured. Not only is the overall scheme capable of maximizing the amount of information extracted, while minimizing the number of test application steps required, but furthermore the analysis required to implement the insights brought to bear is minimal and can be comfortably folded in so as not to disturb the pipelined nature of test application.

It is evident that the result of a test application to a particular partition will be either a correctness or a failure indication. While correctness information is obviously beneficial

---

\*The work of the first author is being supported by an IBM Graduate Fellowship.

as it exonerates the participating scan cells, it has remained unclear how to utilize signature mismatch information. As previous approaches have adopted a “guilty unless proven innocent” philosophy, the failing partitions seemed not to have any new information to offer. Yet the shortcomings of this cynical perspective are also evident in that the cells truly to blame can never then be ascertained unless one assumes to have knowledge of the exact number of culprits. We provide in this work ways to utilize signature mismatch information and thus discover with certainty the culprits, with no reliance on this highly unrealistic assumption of perfect knowledge on the number of culprits.

In this work, we propose a scheme in which signature failure information in addition to correctness information is utilized in order to attain full diagnostic resolution. Processing the signature information in a pipelined manner with the diagnosis procedure helps reduce diagnosis time by providing an exact termination point. Furthermore, analysis results attained during test application help reduce the number of partitions to be examined by enabling partition skipping. Finally, extensive utilization of signature content helps further reduce diagnosis times.

## 2. Previous Work

Previous research concentrates on two distinct tracks in diagnosis of scan based designs with BIST. A thread of work focuses on failing test vector identification [7, 8, 1]. An alternative approach targets the identification of scan cells that capture incorrect information during test [9, 6, 3, 2]. A limited set of efforts attacks both problems simultaneously, most notably [5].

Savir and McAnney propose a method to identify multiple errors in a test sequence through utilization of cyclic registers [7]. For high error counts, the proposed scheme results in misidentification of error-free outputs as erroneous and of erroneous outputs as error-free. Stroud and Damarla propose a technique wherein the characteristic polynomial is a factored polynomial, and show that utilization of non-primitive polynomials may reduce the aliasing probability for multiple errors [8]. Aitken and Agarwal propose a method for utilizing the quotient, instead of the signature, through a fault-free sequence generator. Even though the scheme is capable of providing improved diagnostic capability by utilizing increased information, its hardware requirements are substantially higher than regular BIST [1].

Wu and Adham propose utilization of t-error detecting codes for identification of failing scan cells [9]. Encoded signatures are attained by utilizing a programmable MISR and failing cells are determined by analyzing the resultant signatures off-line. Rajski and Tyszer propose an LFSR-based partitioning approach for detection of failing scan cells [6]. The proposed scheme utilizes a successive partitioning and elimination approach in order to narrow down

possible fault sites. Bayraktaroglu and Orailoglu examine applications of the superposition principle on diagnosis time reduction in scan-based BIST [3]. On a different note, they introduce a low-cost deterministic partitioning technique and show that such techniques surpass LFSR-based ones in that they provide diagnostic resolution in lower times [2].

Jayabrata *et al* [5] propose a pruning scheme which, while providing information about the scan cells that capture errors, significantly reduces the probability of misidentification of error-free outputs as erroneous. A partitioning scheme, in which a signature mismatch is followed by an individual examination of each cell, is utilized for failing scan cell determination. The diagnosis time for determination of failing scan cells with this approach exceeds somewhat that of the method proposed by Rajski and Tyszer [6].

## 3. Preliminaries

In a partitioning-based diagnosis scheme, scan cells are successively grouped into sets of non-overlapping partitions, each constituting a partition group. The set, comprised of the cells of the  $b^{th}$  partition of the  $c^{th}$  partition group, is denoted by  $P(c, b)$ . The fault free signature obtained in logic simulation by observing the outputs of the scan cells in the set  $P(c, b)$  is consequently denoted as the *Simulation Signature*,  $SS(c, b)$ .

A partitioning based diagnosis scheme requires a preprocessing step, during which a decision on the number of partitions per partition group is made and the fault-free signatures corresponding to each partition are calculated. The computational complexity of this preprocessing step is similar to that of BIST, as both require logic simulation of the test set and determination of the test signatures. While only one signature needs to be calculated for BIST, the calculation of the additional signatures for the diagnosis procedure does not increase the preprocessing time noticeably as signature calculation is significantly simpler than performing logic simulation on a large number of patterns. While the preprocessing step of calculating the fault free signatures for partitions is performed only once, the rest of the tasks in the diagnosis procedure have to be performed for each IC being diagnosed.

The diagnosis procedure consists of two distinct tasks: signature collection and signature analysis. A pseudo-code for both of these tasks is given in figure 1. The signature collection step of the diagnosis procedure applies the on-chip generated test repeatedly and determines the *Test Signature*,  $TS(c, b)$ , corresponding to the partitions,  $P(c, b)$ , by utilizing on-chip partitioning hardware. Generation of the test signatures requires one test application per partition; the diagnosis time is directly proportional to the number of partitions utilized during the diagnostic procedure. While signature determination requires multiple applications of the

## Signature Collection

```
for each c in Partition Groups
  PartitionSet ← PartitionsOf(c)
  for each b in PartitionSet
    Start test
    Wait until test done
    Collect signature  $TS(c, b)$  for  $P(c, b)$ 
```

## Signature Analysis

```
function UpdateLists(Group, Partition)
  if  $TS(\text{Group}, \text{Partition}) = SS(\text{Group}, \text{Partition})$ 
     $ACL \leftarrow ACL - P(\text{Group}, \text{Partition})$ 
     $SCL \leftarrow SCL \cup P(\text{Group}, \text{Partition})$ 
```

## Diagnosis Procedure

```
 $ACL \leftarrow$  All Scan Cells
 $SCL \leftarrow \emptyset$ 
for each c in Partition Groups
  PartitionSet ← PartitionsOf(c)
  for each b in PartitionSet
    UpdateLists(c, b)
```

**Figure 1. Signature collection and analysis**

## Diagnosis Procedure

```
 $ACL \leftarrow$  All Scan Cells
 $SCL \leftarrow \emptyset$ 
for each c in Partition Groups
  PartitionSet ← PartitionsOf(c)
  for each b in PartitionSet
    Start test
    UpdateLists(c, b-1)
    Wait until test done
    Collect signature  $TS(c, b)$  for  $P(c, b)$ 
```

**Figure 2. Combined diagnosis pseudo-code**

same test set, the collection of the resultant signature can be performed by a simple tester, which only has to interact with the circuit at the end of each test application to collect the resultant signatures.

As no prior information is assumed about the scan cells, the signature analysis starts by placing all cells in the *Ambiguous Cell List* ( $ACL$ ). It then compares the collected signatures to the fault-free signatures and in case of a match, all cells inside the corresponding partition are removed from  $ACL$  and migrate to the set of fault-free cells, the *Sound Cell List* ( $SCL$ ). At the end of signature analysis,  $ACL$  contains all failing<sup>1</sup> scan cells and possibly includes a number of non-failing scan cells as well.

The diagnosis procedure outlined up until now represents a conceptual rendering of current efforts in partitioning based scan BIST. The diagnosis procedure relies on the fact that some partitions contain no faulty cells; if all partitions consistently embedded a faulty cell, no cell could be declared to be sound. Since usually the number of partitions

<sup>1</sup>It should be evident that scan cells themselves are not faulty but rather constitute the repository of faulty information generated in their logic cones. We use *failing scan cells* or *faulty cells* throughout the paper to denote scan cells in which faults are manifested, for reasons of simplicity and legibility.

is chosen to exceed the number of expected faulty cells [6], the existence of at least one fault free partition per partition group is guaranteed.

As signature analysis requires simple comparisons and set operations, it can be performed in parallel with signature collection. While the BIST hardware generates the signature for a partition, the tester can process the signature of the prior partition, as shown in the pseudo-code<sup>2,3</sup> of figure 2. A pipelined execution opens up new avenues for instant diagnostic state update, quite useful in the new approaches and methods that we propose in the next section.

## 4. Methodology

One of the challenging questions raised by the previously suggested diagnosis procedures is setting an appropriate termination condition for unambiguous diagnosis. Previous approaches keep sieving fault free cells whenever a partition with a correct signature is encountered, yet have no approach for ascertaining a cell to be faulty. The only way they can achieve knowledge of faulty cells is waiting until the set of ambiguous cells remaining equals the number of failing scan cells. The problem gets compounded as expecting knowledge of the number of failing scan cells prior to diagnosis is frequently unrealistic. While it is true that the test literature has relied to a large extent in the past with some success on assumptions such as a single gate level fault model, the situation here is not quite analogous since the required knowledge is actually the number of fault *manifestations* on scan cells. As various logic faults can manifest themselves in varying numbers of scan cells, schemes based on the prediction of the number of failing scan cells are frequently incapable of providing the set of faulty scan cells with certainty. In any case, it is not even obvious that a single gate level logic fault assumption holds for diagnosis purposes.

We propose instead in this paper an alternative methodology, capable of identifying faulty scan cells with no ambiguity. We achieve this by conceptually extending the sets of scan cells, from the two used previously [6] of *ambiguous*<sup>4</sup> and *non-faulty*, to three, to include the set of *certainly faulty* cells. Scan cells which are initially all presumed to be in the ambiguous set can migrate in the method we propose not only to the non-faulty set but also to the certainly faulty set. The more intricate condition that needs to be satisfied for populating the certainly faulty set is typically infrequently achieved at the early stages of the algorithm, even though its

<sup>2</sup>All pseudo-codes given in the paper may require obvious modifications to handle boundary cases; for example, the pseudo-code in figure 2 requires trivial modifications to handle the case of  $b = 0$ . We omit all such boundary case modifications to ensure legibility.

<sup>3</sup>All incremental modifications in pseudo-codes are given in bold, to help easily trace the changes.

<sup>4</sup>Members of the set of ambiguous cells are referred as “the cells who are not resolved yet” in [6].

frequency increases in latter stages. Nonetheless, increasing attention needs to be paid to reducing diagnosis time so that the speedy refinement of the latter part of the algorithm can be attained. We propose consequently a mechanism to complement the scheme so that significant time reductions in diagnosis can be attained.

This section is concerned with the description of the main technical breakthroughs suggested in this paper. We start by answering the fundamental questions of how to utilize failure information in a signature and show how focusing only on information gleaned from correct signatures, the practice until now, has led to the current predicament of being unable to identify failing scan cells unambiguously. We proceed with an examination of how diagnosis time can be reduced through advance revelations of the usefulness of the upcoming partition and extend such performance improvements by extracting even more information from the mismatching signatures so that diagnosis can be even further sped up.

#### 4.1. Ascertaining A Faulty Cell

Failure of a signature  $TS(c, b)$  necessitates the existence of at least one failing scan cell in  $P(c, b)$ . The intersection of  $P(c, b)$  and  $ACL$  in this case provides a set of candidate failing cells. If this intersection produces a singleton set, the sole common cell can be unambiguously declared to be faulty. Though existence of such cases is rare at the initial stages of the diagnosis procedure, towards the final stages of the procedure, the number of cells in  $ACL$  is gradually reduced, thus increasing in turn the probability of such occurrences. The following condition can be utilized to detect failing scan cells.

$$TS(c, b) \neq SS(c, b) \text{ and } |P(c, b) \cap ACL| = 1 \quad (1)$$

A pseudo-code that utilizes information from non-matching signatures, in addition to information provided by matching signatures, is shown in figure 3. The code uses  $FCL$  to denote the *Faulty Cell List*. Not only does the code in figure 3 provide exact knowledge of failing cells, but it also does so at the earliest possible time and thus terminates instantly when all ambiguous cells have migrated to either the sound cell list or to the faulty cell list.

The speed of the process of sifting certainly faulty cells is inversely correlated to the number of ambiguous cells at hand; consequently, examination of partitions at the early stages of the diagnosis process is typically fruitless. Nonetheless, as the fruitlessness of the process is a function of the number of ambiguous cells and not of the information associated with the partition, revisiting the partition information after the ambiguous cell list is thinned out helps sift additional certainly faulty cells and speeds the diagnosis process in turn. We expand on this process in section 4.4.

```
function UpdateLists(Group, Partition)
  if TS(Group, Partition) = SS(Group, Partition)
    ACL ← ACL - P(Group, Partition)
    SCL ← SCL ∪ P(Group, Partition)
  else
    if |P(Group, Partition) ∩ ACL| = 1
      NewFaultyCellSet ← P(Group, Partition) ∩ ACL
      FCL ← FCL ∪ NewFaultyCellSet
      ACL ← ACL - NewFaultyCellSet
```

#### Diagnosis Procedure

```
ACL ← All Scan Cells
SCL ← ∅
FCL ← ∅
for each c in Partition Groups
  PartitionSet ← PartitionsOf(c)
  for each b in PartitionSet
    Start test
    UpdateLists(c, b-1)
    if ACL = ∅ terminate
    Wait until test done
    Collect signature TS(c, b) for P(c, b)
```

Figure 3. Utilizing non-matching signatures

#### 4.2. Skipping Partitions

One of the interesting things that occurs in partitioning based diagnosis is that signature match information on an upcoming partition can be preordained. The simplest case of this phenomenon occurs when all cells that participate in a partition have been already ascertained to be fault free. Obviously, such a partition is guaranteed to provide the correct signature; no test application is necessary in that case. The process of skipping such partitions preordained to provide the correct signature is denoted *Skipping Fault Free Partitions (SFFP)*. A partition,  $P(c, b)$ , can be skipped under SFFP when the following condition holds.

$$P(c, b) \subset SCL \quad (2)$$

A more intricate case of the same phenomenon occurs when we know that a partition generates an *incorrect* signature. If even a single element of the partition is known to be faulty for sure, i.e. belongs to  $FCL$ , then that partition will generate a faulty signature. The process of skipping such partitions preordained to generate an incorrect signature is denoted *Skipping Faulty Partitions (SFP)*. A partition,  $P(c, b)$ , can be skipped under SFP when the following condition holds.

$$P(c, b) \cap FCL \neq \emptyset \quad (3)$$

It is interesting to recap by looking at the partitions that do *not* get skipped. There is no way to have foreknowledge of the signature of a partition with no  $FCL$  members but at least one  $ACL$  member. Interestingly, the frequency of both of these conditions decreases in the course of the execution of the diagnostic procedure we outline, resulting in an increase in the number of skipped partitions in turn.

```

function Skippable(Group,Partition)
  if P(Group,Partition)  $\subset$  SCL return true
  if P(Group,Partition)  $\cap$  FCL  $\neq \emptyset$  return true
  return false

```

#### Diagnosis Procedure

```

ACL  $\leftarrow$  All Scan Cells
SCL  $\leftarrow \emptyset$ 
FCL  $\leftarrow \emptyset$ 
for each c in Partition Groups
  PartitionSet  $\leftarrow$  PartitionsOf(c)
  for each b in PartitionSet
    if Skippable(c,b) continue
    Start test
    UpdateLists(Last_c,Last_b)
    if ACL =  $\emptyset$  terminate
    Wait until test done
    Collect signature TS(c,b) for P(c,b)
    Last_c  $\leftarrow$  c, Last_b  $\leftarrow$  b

```

**Figure 4. Pseudo-code for partition skipping**

While both SFFP and SFP involve skipping partitions, the set of partitions skipped by the two is distinct and nonoverlapping. Therefore, their simultaneous application does not attenuate their benefits. The pseudo-code in figure 3 can be modified by adding a function to the tester functionality that checks whether a partition can be skipped before starting application of the test to that partition. The resultant pseudo-code is given in figure 4. If the partition can be skipped, the state of the partitioning hardware is modified by the tester to continue with the subsequent partition. It is possible that multiple partitions or even a whole partition group at a time can be thus skipped.

### 4.3. Signature Content

Utilization of signature match information alone, with no content, is sufficient to identify failing scan cells unambiguously. Once a failing scan cell has been unambiguously identified, utilization of signature content enables determination of its signature. The trivial equation 4

$$P(c, b) = P(c, b) - (P(c, b) \cap SCL) \cup (P(c, b) \cap SCL) \quad (4)$$

leads to the following result due to the linearity of a signature compactor.

$$TS(c, b) = TS_{P(c,b)-(P(c,b) \cap SCL)} \oplus TS_{P(c,b) \cap SCL} \quad (5)$$

Since the set of cells  $P(c, b) \cap SCL$  is evidently fault free:

$$TS_{P(c,b) \cap SCL} = SS_{P(c,b) \cap SCL} \quad (6)$$

$$TS_{P(c,b)-(P(c,b) \cap SCL)} = TS(c, b) \oplus SS_{P(c,b) \cap SCL} \quad (7)$$

Equation 7 can be utilized in conjunction with the condition in equation 1 to determine the signature of a failing scan cell. Once the signature of a failing cell has been determined, it can be put to use to help reduce diagnosis time further. Otherwise, a partition that contains a single faulty cell

but multiple ambiguous cells eliminates no cell from  $ACL$ . If removal of the faulty cells from the partition through superposition results in a fault free signature, the remaining cells, thus proven to be fault free, can migrate from  $ACL$  to  $SCL$ . In an analogous fashion, a faulty signature may be utilized to detect a faulty cell, if the reduced partition contains only one ambiguous cell. Again, diagnosis time can be further reduced by utilizing the content of the signature itself. We denote these types of optimizations as *Faulty Signature Utilization (FSU)*.

It is interesting to observe that both the FSU and the SFFP augmentations can be incorporated into the algorithm with no conflict; after all, FSU is invoked only in the case of a faulty signature while SFFP is invoked only in the case of a correct signature. Yet in the case of SFP, its simultaneous invocation with FSU needs to be more carefully orchestrated as they both apply exactly under the same condition of a signature mismatch. We examine this issue empirically in section 5 and provide consequent recommendations based on experimental data.

The pseudo-code that has been proposed to utilize signature match information only can be augmented to determine the signature of the faulty cells. The signatures attained from test application and corresponding partitions are updated by utilizing the faulty signature information. If the signature of a faulty cell  $n$  is determined to be  $TS_n$ , a partition containing the cell  $n$  and its signature can be updated by:

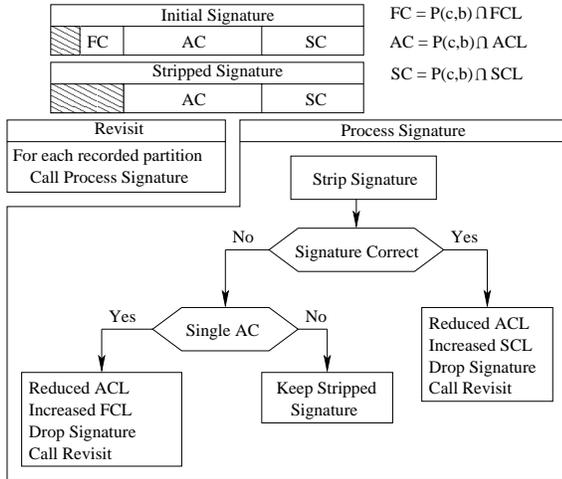
$$TS(c, b) = TS(c, b) \oplus TS_n, \text{ if } n \in P(c, b) \quad (8)$$

$$P(c, b) = P(c, b) - n, \text{ if } n \in P(c, b) \quad (9)$$

### 4.4. Revisiting Partitions

While the methods outlined up until now constitute a powerful ensemble, one of the aspects that they overlook is that the information to be extracted from a signature is a function of the current knowledge regarding cell status. It follows from this observation that it may pay to reexamine signatures that have already been examined; increasing knowledge about cells may thus help generate new knowledge at no cost from previously examined signatures.

The operations that augment our state of knowledge regarding scan cell status are the migration of a single cell from the *Ambiguous Cell List* to the *Faulty Cell List*, as shown in section 4.1, or the migration of a number of scan cells from the *Ambiguous Cell List* to the *Sound Cell List*, as already studied in previous work in this area [6, 3, 2]. The approach outlined in section 4.3 can increase the efficiency of both methods by peeling off faulty cell information from the signature. Both of these operations, whether they introduce a single faulty cell or possibly multiple healthy cells, merit revisits of the previous partitions and typically pay off richly in providing new information regarding cell status in



**Figure 5. Revisiting previous partitions**

turn. In any case, as the hard work of applying the test and capturing of new signatures has already been accomplished, revisiting the signatures is a time-effective process that by no means can impact temporally the smooth execution of the test application pipeline.

While any type of knowledge merits a reexamination of the set of previously obtained signatures, not every signature is worth reexamining. A simple way to verify this observation is to ponder the fate of a signature which matches the simulation signature immediately upon being obtained; all its ambiguous cells are declared sound and the signature has no further information to provide at that point. The same approach and analysis though can be extended to faulty signatures also! Once the effect of the scan cells known to be faulty has been stripped from the signature, the correctness of the underlying signature is an indication of the remaining information content in the signature; a signature is worth revisiting only if it continues to be incorrect!

It is important therefore to think of the signatures in their stripped versions, with the effect of the cells known to be faulty peeled, in order to explore the revisit of the previously obtained signatures. Such stripped signatures can benefit both from new faulty cell information just as well as from information regarding new healthy cells. As a matter of fact, the impact of new faulty cell information is more far reaching. In that case, the stripped signature is peeled once more to uncover the effects of the newly ascertained faulty cell; if the peeled signature turns out to be correct, all remaining ambiguous cells are declared healthy in turn, starting another recursive invocation of the revisit procedure! If the final peeled signature continues to be incorrect, it is evident that the signature still embeds information to be extracted and the signature is returned for a later revisit.

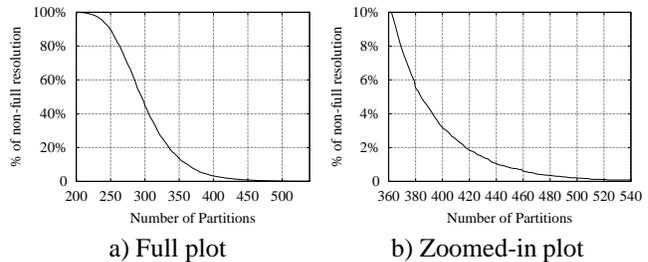
In the case of new information regarding a set of healthy cells, additional information can be extracted only if the re-

maining ambiguous cells in the partition are reduced to one. In that case, that one cell is the only possible cause for the failing signature; it can consequently be declared faulty, in the process extinguishing the usefulness of the information resident in the partition. Of course, the consequent knowledge in this case of a new faulty cell constitutes cause for a recursive invocation of the revisit procedure yet once more. A graphic representation of this recursive partition revisit procedure is given in figure 5.

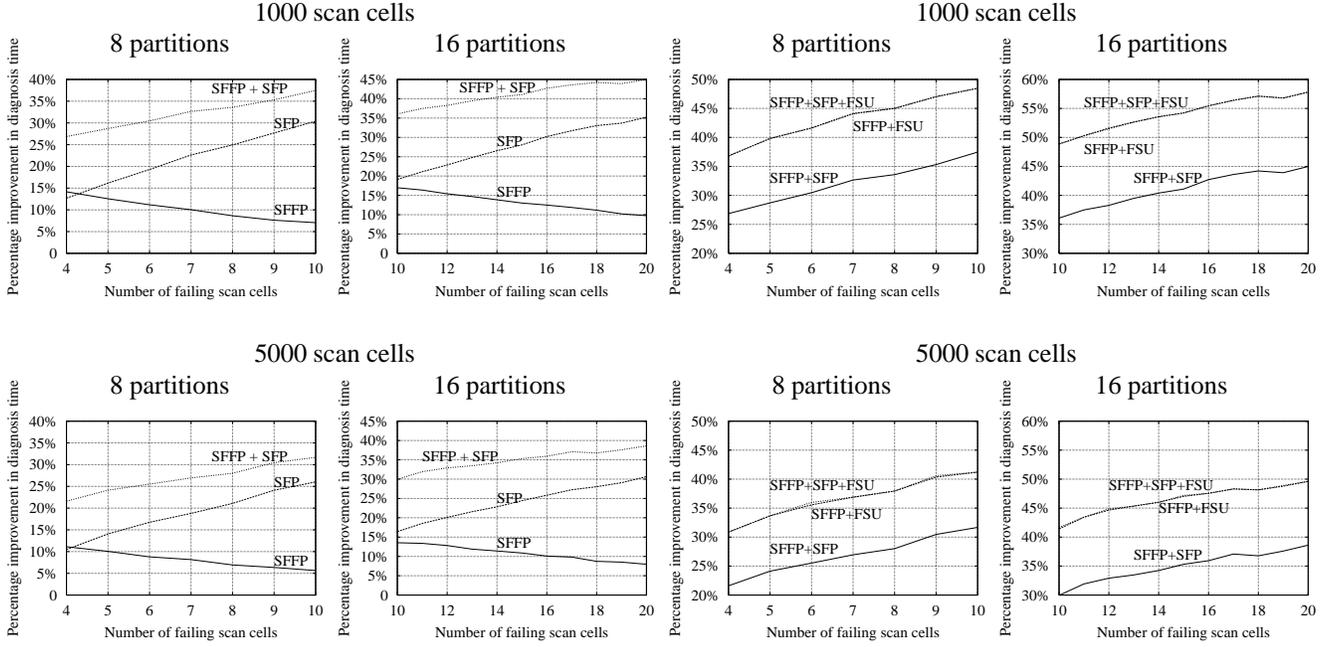
## 5. Simulation Results

In this section, we initially report on experiments that provide a quantitative assessment of the improvements attained by utilizing signature mismatch information, which results in full diagnostic resolution. We follow this up with further experiments to assess the level of diagnostic time reduction attained by skipping useless partitions and by making use of signature content. The appropriate recursive revisits are effected in both cases.

It is challenging to provide a quantitative measure of the improvements attained by incorporating failure information as the proposed scheme provides accurate diagnostic information, as opposed to the ambiguous information achieved by preceding schemes. We compare the proposed method to previous methods by focusing on the termination condition, essential for any partitioning based diagnosis scheme. Previously, a preset number of partitions had been utilized as a termination condition. The scheme that we propose utilizes instead an automatically adjustable condition that ensures termination immediately upon resolving all ambiguous cells. We identify a quantitative termination condition for the preceding schemes by simulating the diagnosis procedure on randomly inserted failures under the assumption of a certain number of failing scan cells. During simulations, utilizing knowledge of the exact number of failures inserted, the required diagnosis time for full resolution can be determined for each random failure insertion experiment. Random fault insertion simulations are repeated 10,000 times for a scan chain of length 5,000 and with 16 partitions per partition group. Each time 15 failures are randomly inserted into the scan chain and the number of partitions required for full diagnosis resolution is recorded.



**Figure 6. Diagnosis time distribution**



**Figure 7. Diagnosis time reduction through Partition Skipping**

**Figure 8. Diagnosis time reduction through Exploitation of Signature Content**

Figure 6a shows the percentage of the trials for which the specified number of partitions proved insufficient for full diagnostic resolution. Figure 6b shows a zoomed-in version of the lower right portion of the plot of Figure 6a. The zoomed-in version clearly indicates that 10% of the diagnosis tries required more than 360 partitions for full resolution. While the average number of partitions required is 301, setting the number of partitions to 301 implies missing full resolution approximately 42% at the time. Guaranteeing full resolution for 99.9% of the fault insertion experiments would necessitate utilization of 540 partitions, i.e. an almost 80% increase over average diagnosis time.

Applying the proposed scheme on the previous experimental setup shows that full diagnostic resolution is attained through 303 partitions on the average. On the other hand, it is easy to observe in figure 6a that terminating the preceding approaches upon reaching 303 partitions would again leave approximately 42% of the trials with incomplete diagnostic resolution. This experimental data makes it evident that one aspect of the approaches we propose in this paper, namely utilization of signature *mismatch* information buys us either a 40% reduction in diagnosis times for comparable (full) diagnosis or alternatively provides us much higher diagnostic resolution at comparable time points.

As the diagnosis procedure proposed for full diagnostic resolution is thus proven to be time-effective, individual methods proposed in this work are compared relative to each other and to a baseline method that utilizes both signature match and mismatch information, but no signature

content exploitation nor any partition skipping.

In order to determine the expected diagnosis time, random fault insertion is performed on scan chains of length 1,000 and 5,000. Through simulation, diagnosis time is determined for a set of 500 fault insertions and the resultant diagnosis times are averaged to produce the expected diagnosis time. For both scan chain sizes, the number of partitions is chosen to be 8 and 16; the number of failing scan cells ranges from 4 to 10 and from 10 to 20, respectively.

The simulation results reported in figure 7 for the case when signature match information is utilized indicate that the combined application of both partition skipping methods reduces fault diagnosis time by 20% to 45% depending on scan size and the number of faulty partitions. The performance of SFFP dwindles as the number of faults increases; the increased number of faults reduces the probability of encountering a fault free partition. In the case of SFP, on the other hand, the probability of encountering faulty partitions increases as the number of failing scan cells increases, increasing the effectiveness of SFP. The results in figure 7 confirm the fact that total improvement is equal to the sum of improvements of SFFP and SFP individually, as suggested in section 4.

Diagnosis results for the case when the signature content is utilized are provided in figure 8. As can be seen in the figures, skipping partitions by itself accounts for more than half of the improvements. As is evident from the fact that the lines marked “SFFP + FSU” and “SFFP + SFP + FSU” can hardly be distinguished, utilization of FSU virtually

Method	Number of Partitions	% Reduction
Original	40.7	NA
+ SFFP	34.2	16.0
+ SFP	29.6	27.2
+ FSU	24.9	38.8

**Table 1. Diagnosis results for s5378**

eliminates the additional benefits that can be expected from SFP; apparently, once information can be extracted from faulty signatures through FSU, skipping them becomes of dubious value. It is interesting to note furthermore that the simultaneous application of all the methods that we propose denoted by the top line in all the graphs of figure 8 never fails to reduce diagnosis time by at least 30%. As a matter of fact, as the number of failing scan cells increases, the delivered performance improvements increase to reach an average of approximately 50% as can be seen at the right hand side of these graphs.

While in general the number of partitions chosen needs to exceed the number of failing cells, selection of the number of partitions cannot be accomplished by considering the faults that have the largest number of manifestations on scan cells, as doing so will increase average diagnosis time appreciably. Therefore, during the diagnosis of some ICs, the number of failing cells may exceed the number of partitions. The proposed scheme, however, keeps the diagnosis time within a reasonable range even for such cases, as the improvement of the scheme proposed increases with the number of failing cells.

In order to validate the proposed scheme, the diagnosis procedure is applied to a benchmark circuit, s5378 [4]. Fault simulations are performed to determine fault free and faulty signatures for each scan cell, which is consequently utilized in order to generate signatures for LFSR-generated partitions. While fault simulations for determining the signature for each fault are computationally complex, such simulations are utilized in this work for verification purposes only. In an actual diagnosis environment, fault free signatures will be determined through logic simulation and incorrect signatures generated by the circuit under test. The circuit s5378 has 35 primary inputs and 189 flip-flops. The diagnosis procedure is performed with 8 partitions and for faults that drive at most 9 scan cells. The average diagnosis time for all the faults is reported in table 1. In this table, the second row provides diagnosis times attained without application of any diagnosis time reduction methods proposed in this work. Subsequent rows provide diagnosis times and percentage reduction in diagnosis times over the predecessor row for application of each additional method proposed in this work. All failing scan cells are determined exactly for all the faults. While diagnosis time is reduced by 27.2% if no signature content is utilized, its utilization boosts the reduction in diagnosis time to 38.8%.

## 6. Conclusion

As ever shorter time-to-market considerations increasingly get challenged by the lengthy process of diagnosis, providing cost-effective diagnosis methodologies that can pinpoint failure causes becomes imperative. BIST-based test methodologies, increasingly commonplace in today's designs, raise interesting challenges in utilizing information laden with ambiguity. We outline in this paper methodologies capable of pinpointing *all* failing scan cells *unambiguously*.

The methods we propose for resolving all such ambiguity surprisingly help keep diagnosis time below the schemes previously proposed. Furthermore, application of partition skipping and extensive utilization of signature content during the diagnosis procedure even further reduces diagnosis times. As such techniques that rely on information extraction and partition pruning are to be effected offline, no computational hardware cost is incurred. The results outlined show that not only is full resolution attained, thus reducing design debugging time, but furthermore BIST invocations are sharply reduced. The twin benefits promise to expand the applicability of BIST-based techniques to diagnosis and thus match and complement the strides and penetration achieved by them in the test arena.

## References

- [1] R. C. Aitken and V. K. Agarwal. A diagnosis method using pseudo-random vectors without intermediate signatures. In *International Conference on Computer-Aided Design*, pages 574–580, 1989.
- [2] I. Bayraktaroglu and A. Orailoglu. Deterministic partitioning techniques for fault diagnosis in scan-based BIST. In *International Test Conference*, pages 273–282, October 2000.
- [3] I. Bayraktaroglu and A. Orailoglu. Improved fault diagnosis in scan-based BIST via superposition. In *Design Automation Conference*, pages 55–58, June 2000.
- [4] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *Proceedings of the International Symposium on Circuits and Systems*, pages 1929–1934, 1989.
- [5] J. Ghosh-Dastidar, D. Das, and N. A. Toubia. Fault diagnosis in scan-based BIST using both time and space information. In *International Test Conference*, pages 95–102, 1999.
- [6] J. Rajski and J. Tyszer. Diagnosis of scan cells in BIST environment. *IEEE Transactions on Computers*, 48(7):724–731, July 1999.
- [7] J. Savir and W. H. McAnney. Identification of failing tests with cycling registers. In *International Test Conference*, pages 322–328, 1988.
- [8] C. E. Stroud and T. R. Damarla. Improving the efficiency of error identification via signature analysis. In *VLSI Test Symposium*, pages 244–249, May 1995.
- [9] Y. Wu and S. M. I. Adham. Scan-based BIST fault diagnosis. *IEEE Transactions on Computer-Aided Design*, 18(2):203–211, February 1999.