A Partition-Based Approach for Identifying Failing Scan Cells in Scan-BIST with Applications to System-on-Chip Fault Diagnosis

Chunsheng Liu and Krishnendu Chakrabarty

Department of Electrical & Computer Engineering
Duke University, Durham, NC 27708
E-mail: {mliu, krish}@ee.duke.edu

Abstract

We present a new partition-based fault diagnosis technique for identifying failing scan cells in a scan-BIST environment. This approach relies on a two-step scan chain partitioning scheme. In the first step, an interval-based partitioning scheme is used to generate a small number of partitions, where each element of a partition consists of a set of scan cells. In the second step, additional partitions are created using an earlier-proposed random-selection partitioning method. Two-step partitioning leads to higher diagnostic resolution than a scheme that relies only on random-selection partitioning, with only a small amount of additional hardware. The proposed scheme is especially suitable for a system-on-chip (SOC) composed of multiple embedded cores, where test access is provided by means of a TestRail that is threaded through the internal scan chains of the embedded cores. We present experimental results for the six largest ISCAS-89 benchmark circuits and for two SOCs crafted from some of the ISCAS-89 circuits.

1 Introduction

As feature sizes shrink and designs become more complex, built-in self-test (BIST) and scan design are gaining acceptance as industry-wide test solutions. The combination of scan design and BIST, commonly referred to as scan-BIST, is now especially common [1]. A scan-BIST technique typically applies a large number of pseudorandom patterns to the circuit under test via internal scan chains. The responses are then captured by the scan chain and compacted to a very short signature in a multiple-input shift register (MISR). However, a drawback of this approach is that the signature in the MISR does not provide enough information for fault diagnosis, either to determine failing test vectors or to identify error-capturing (failing) scan cells.

For a scan-BIST scheme involving millions of vectors and tens of thousands of scan cells, the diagnosis and failure analysis time can be extremely high [1-4, 9]. Due to the large numbers of scan cells, it is often difficult to determine a small set of failing scan cells that can be used for failure analysis. Hence, there is a need for scan-BIST schemes that offer high diagnostic accuracy, i.e., they provide a small set of failing scan cells, without requiring excessive test application time, processing time, and additional on-chip hardware.

Early work on failing scan cell identification was based on scan chain partitioning and multiple BIST sessions. In the random-selection partitioning scheme in [5], the scan chain is partitioned into non-overlapping subchains, and each group (subchain) within the partition consists of a set of randomly-selected scan cells. Test patterns are applied to the circuit under test in multiple BIST sessions. The number of sessions is equal to the number of groups in the partition. In each session, a BIST signature is generated only for the scan cells that belong to the group within the partition corresponding to that session.

A single partition is generally insufficient for obtaining a small set of candidate failing scan cells, hence the BIST sessions must be repeated for additional partitions. These partitions are randomly generated in [5]. Subsequent efforts to increase diagnostic accuracy and reduce test application time have relied on binary search on a large number of scan cells [6], the principle of superposition [7], and deterministic partitioning [8]. However, an improvement in accuracy is often accompanied with a corresponding increase in control logic overhead and storage requirements for additional signatures [6, 8]. In addition, previous approaches have been evaluated using a small number of errors that are randomly-injected into the scan chains, and not using actual fault injection in benchmark circuits. Faults in real circuits do not produce errors in scan cells in a random fashion, and failing scan cells typically tend to be clustered.

The clustering of failing scan cells has an even greater impact on diagnosis of system-on-a-chip (SOC) designs that contain multiple embedded cores. Test access in such SOCs is often provided by means of a TestRail that is threaded through the internal scan chains of the embedded cores [10]. A spot manufacturing defect in a core-based SOC is likely to affect only a small number of cores, which implies that the failing scan cells can be expected to be clustered in the scan chains of the faulty cores. Known space diagnosis approaches that rely on random partitioning are therefore unlikely to be effective for the

* This research was supported in part by the National Science Foundation under grants CCR-9875324 and CCR-0204077.

identificaton of failing scan cells in core-based SOCs.

We present a new scan-BIST approach for determining failing scan cells for fault diagnosis. The proposed approach relies on a two-step scan chain partitioning scheme. In the first step, an interval-based partitioning scheme is used to generate a small number of partitions, where each group within a partition (referred to as an interval) consists of a set of scan cells that are consecutively ordered in the scan chain. In the second step, additional partitions are created using the random-selection partitioning method. Two-step partitioning leads to higher diagnostic resolution than a scheme that relies only on random-selection partitioning, with only a small amount of additional hardware. Moreover, this approach is especially suitable for core-based SOCs since it targets clusters of failing scan cells. We improve the hardware architecture in [5] by implementing this two-step method with only two additional registers.

The organization of this paper is as follows. In Section 2, we describe the two-step partitioning scheme and present the corresponding scan-BIST architecture. In Section 3, we present the motivation for the two-step partitioning scheme and experimental results for the ISCAS-89 benchmark circuits with a single scan chain each. In Section 4, we extend the method to multiple scan chains as well as to failing scan cell identification in core-based SOCs. Finally, in Section 5, we present experimental results for two SOCs crafted from the ISCAS-89 circuits, considering the scenarios of a single scan chain and multiple scan chains.

2 Scan Chain Partitioning

In this section, we first review random-selection partitioning and present the scan-BIST architecture that is necessary for its implementation. We then describe the proposed interval-based partitioning scheme and describe the scan-BIST architecture needed to implement the proposed two-step scheme.

2.1 Random-Selection Partitioning

The identification of failing scan cells using random-selection partitioning proceeds as follows. First, the scan cells under diagnosis are randomly selected and placed in a number of non-overlapping groups [5, 8]. This corresponds to the first partition. Multiple BIST sessions are then used for identifying failing scan cells, where each session corresponds to a group of scan cells within the partition. In the i^{th} BIST session, the MISR generates a signature only for the test responses captured by the scan cells in the i^{th} group of the partition. (The test responses for the other scan cells are masked.) In order to improve the diagnostic accuracy, a different partition is generated next, and the BIST sessions are repeated for this partition. This process is continued until a pre-defined number of partitions have been used, and signatures generated for the groups of scan cells within these partitions. Since there is overlap between the groups of scan cells in different partitions, the principle of inclusion and exclusion can be used to prune non-failing scan cells. Additional pruning techniques can further refine the set of failing candidate scan cells [7].

Figure 1 illustrates a scan cell selection hardware architecture implemented using an LFSR and Initial Value Register (IVR) [5] (The shaded blocks are used for two-step partitioning.) At the beginning of test application for each group within a partition, the LFSR is loaded from the IVR. The output of any r stages of the LFSR can be regarded as a r-bit binary label associated with a specific scan cell. Clearly, if the number of groups in a partition is b, the length of Test Counter 1 is $r = \lceil log_2 b \rceil$. On every shift, the label is compared with the current group number, i.e., the content of Test Counter 1. If a match occurs, the corresponding scan cell content is allowed to enter the compactor for signature analysis, otherwise it is masked. For the next group in the partition, i.e., test session, the LFSR is re-loaded with the content of the IVR and a new group within the partition is set to selectively mask scan cells. Meanwhile, the Test Counter 1 is incremented such that a different set of scan cells is compared for this group. At the end of each partition, the IVR is updated with the current value of the LFSR to create a different partition.

In the above random-selection scheme, scan cells in a group within a partition are selected randomly, hence the elements of a group are likely to be scattered over the length of the scan chain. An alternative is to place in the same group a set of scan cells that are consecutively ordered in the scan chain, i.e., each group within a partition is an *interval* of scan cells and the partition consists of non-overlapping intervals. One possible interval-based approach is to make all groups within a partition include the same number of scan cells, except for the boundary cases of the first or the last groups. However, this deterministic partitioning with fixed interval length requires expensive control logic in the selection hardware [5].

2.2 Two-Step Partitioning

We next propose a scan cell selection scheme that utilizes both random-selection and interval-based partitioning. The interval length, i.e., the number of scan cells in a group within a partition, is randomly generated for interval-based partitioning, and this leads to a simpler hardware implementation.

The selection hardware is sketched in Figure 1. The logic includes two additional registers, Shift Counter 2 and Test Counter 2, represented by shaded blocks. The interval-based partitioning proceeds as follows. Initially, the LFSR is loaded with a pre-computed seed in the IVR. The seed is associated with a number of bits from the LFSR and these bits can be viewed as a random number representing the length of the current interval. The seed is selected to ensure that a pre-defined number of groups generated by these bits can cover the entire scan chain. Usually there exist a number of such seeds for a given

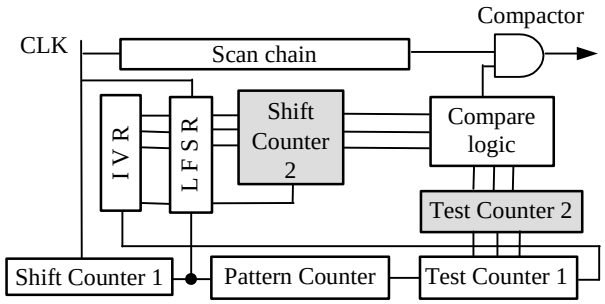
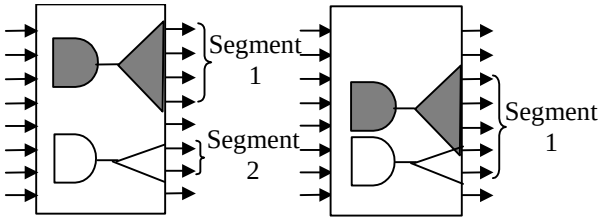


Figure 1. Scan cell selection hardware (additional logic for two-step partitioning is shown shaded).

circuit. In the beginning of each group, Shift Counter 2 is loaded with the current interval length from the LFSR. Simultaneously, Test Counter 1 is incremented and its content is transferred to Test Counter 2. At the end of each interval, Test Counter 2 is decremented. When Test Counter 2 reaches 0, the compare logic outputs '1'. Therefore, the contents of the scan cells are selected and shifted into the compactor; this continues until Shift Counter 2 also reaches 0. The procedure continues for all the patterns in the test and the signature collected in the compactor is the response for the current interval in the partition. This process then iterates until all the intervals in the partition are handled. To use more than one interval-based partition, the LVR is initialized with a different seed and the above procedure is repeated.

We next present an example to illustrate interval-based partitioning. Suppose we have a scan chain with 16 scan cells, and the first partition includes four groups of scan cells. We also assume that the three selected bits of the LFSR, which is seeded by the IVR, generate four pseudorandom patterns 101, 110, 011, and 010. These correspond to intervals of length 5, 6, 3, and 2, respectively. The content of Test Counter 1 and Test Counter 2 are initially both 0. Therefore, the compare logic enables the AND gate to transfer the content of the scan cells to the compactor. Since Shift Counter 2 is next set to 5 and decremented thereafter, the scan cells 1 to 5 are scanned out to the compactor. After all the BIST patterns are applied, Test Counter 1 is incremented, hence Test Counter 2 is now initialized with the value 1. The LFSR is reloaded with the same seed from the IVR and once again Shift Counter 2 is initialized with the value 5. When Shift Counter 2 is decremented to 0 by the shift clock, a carry signal drives the LFSR to shift once and the next random output '6' is loaded into Shift Counter 2. Meanwhile, the content of Test Counter 2 is decremented to 0. As a result, the compare logic outputs a '1' and the scan cells from positions 6 to 11 are transfer their contents to the into compactor. The remaining two intervals are handled in a similar fashion.

After test application using a small number of interval-based partitions, the random-selection scheme is used to create the remaining partitions. This can be done by simply disabling Shift Counter 2 and Test Counter 2 or bypassing them. An important advantages of this scan-BIST architecture is that it can readily incorporate various



(a) Non-overlapping cones (b) Overlapping cones

Figure 2. Fault and associated fault cones.

other partitioning schemes without hardware modifications. Moreover, the entire diagnosis process can be carried out without interruptions or manual intervention. This is in contrast to the adaptive scheme in [6], where test application must be frequently interrupted to execute a binary search procedure.

3 Motivation for Two-Step Partitioning

Prior work on the identification of failing scan cells is based on random-selection partitioning, which is especially effective if the failing scan cells are uniformly distributed across a scan chain. However, in practice, faults in a circuit are likely to lead to clustered failing scan cells, thereby invalidating the uniform distribution assumption implicit in random-selection partitioning. An error due to a fault can only be captured by a set of scan cells that lie within the fault cone, i.e., the scan cells that can be reached by a sensitized path from the fault site. The locations of these error-capturing scan cells in the scan chain depend on the scan chain ordering, but there is nevertheless a clear dependence between the circuit structure and the distribution of failing scan cells. Therefore, errors caused by a fault are restricted to a small segment of the scan chain. If multiple faults exist in the circuit, the fault cones may either be non-overlapping, leading to non-overlapped segments of failing scan cells, as in Figure 2(a), or the fault cones may lead to overlapped segments of failing scan cells, which can be viewed as one expanded segment; see Figure 2(b).

An obvious drawback of random-selection is that since scan cells in any group of a partition are selected randomly, a segment associated with a fault cone is often fragmented, and error-capturing scan cells in a segment are placed in different groups of a partition. Since a single failing scan cell in a group renders all the scan cells in that group to be candidate fails, random-selection decreases the likelihood that a group of non-failing scan cells will be pruned from the candidate set. Note however that this problem is severe only for the first few partitions. Once the candidate set is reduced through pruning, random selection is extremely effective in reducing the candidate set further.

In contrast to random selection partitioning, interval-based partitioning leads to significant pruning for the first few partitions. Since scan cells are grouped into non-overlapping intervals, a segment associated with a fault cone can be covered by a small number of consecutive intervals. This can be done by carefully selecting the

True failing scan cells: 2
00001100000000000000000000000000000000000000000000000000

Interval-based partitioning:
Group 1: 1-4
Group 2: 5-12
Group 3: 13-28
Group 4: 29-52
No. of suspect failing scan cells: 8
00001111111100000000000000000000000000000000000000000000

Random-selection partitioning:
Group 1: 14,15,18,26,35,36,37,47,52
Group 2: 4,7,9,12,16,19,24,27,31,38,42,48,50
Group 3: 3,6,8,11,13,17,23,25,30,34,41,46,49,51
Group 4: 1,2,5,10,20,21,22,28,29,32,33,39,40,43,44,45
No. of suspect failing scan cells: 30
11101101011010001001111010011101110000111011111001010

Figure 3. Candidate failing scan cells determined using a single partition based on interval-based and random selection, respectively.

number of groups within a partition and the seed used to generate interval length. As a result, candidate failing scan cells are restricted to these intervals and this can lead to a significantly smaller set of failing candidates. We have seen in our experiments that one or two interval-based partitions are usually adequate to reduce the candidate size considerably, following which random selection can be employed for further pruning.

We next highlight the effectiveness of interval-based partitioning through the example in Figure 3. For the sake of simplicity, we use one of the smaller ISCAS-89 benchmark circuits, namely s953. Suppose it is implemented as a full-scan circuit with a single scan chain. We inject a single stuck-at fault into this circuit and observe the number of failing scan cells for a randomly-chosen input pattern that detects this fault. For this example, there are two failing scan cells, denoted by '1' in the figure. (The non-failing scan cells are denoted by '0'.)

We then generate one partition each using the interval-based method and the random-selection method, respectively. The partition in each case consists of 4 groups. The figure shows the corresponding partition contents and the candidate failing scan cells determined from pass/fail of each group within the partition. Interval-based partitioning yields much better resolution than random-selection partitioning. This is because the interval-based method places the two failing scan cells in the same group, while the random-selection method disperses them to two different groups.

Despite the obvious advantage of interval-based partitioning highlighted by Figure 3, random selection is superior in one important aspect. By providing greater randomness, it outperforms interval-based partitioning when a large number of partitions are employed. For example, if the first and the last scan cell in the (ordered) scan chain are failing, then they always belong to two different intervals, irrespective of the number of partitions generated. However, in a random-selection partition, they may be included in the same group. Therefore, we combine the benefits offered by the two approaches by adopting a two-step procedure. We first create a small number of partitions using interval-based method. This allows us to achieve rapid corse-grained resolution. Next, the random-selection method is used to generate other partitions, which allows us to achieve fine-grain resolution. For the sake of simplicity, we use only one interval-based partition for the two-step method in our simulations, even though we have observed that in some cases, the use of more interval-based partitions leads to higher diagnostic resolution.

4 Experimental Results

In this section, we present simulation results for the full-scan versions of the ISCAS-89 benchmark circuits. We assume that each of these circuits contains a single scan chain. For each circuit, we inject 500 single stuck-at faults. In contrast to earlier work based on random error injection in scan cells, we inject faults into the circuits and examine their effect on the errors captured in scan cells. We calculate the diagnostic accurancy in terms of *diagnostic resolution DR*, which is defined as follows:

$$DR = \frac{\sum_{\forall \text{faults } f} \text{Number of candidate failing scan cells} - \sum_{\forall \text{faults } f} \text{Number of actual failing scan cells}}{\sum_{\forall \text{faults } f} \text{Number of actual failing scan cells}}$$

A low value of *DR* (*DR* = 0 in the ideal case) denotes high diagnostic accuracy.

In Table 1, we first present results for a small circuit, s953, when the number of partitions is varied from 1 to 8. We consider a BIST test session consisting of 200 pseudorandom patterns. It can be seen that when the number of partitions is small, the interval-based method provides better resolution than random selection. However, for a larger number of partitions, the random selection method yields higher resolution. As discussed in Section 3, this result is expected since random selection provides fine-grained resolution for a larger number of partitions. In all the cases, the two-step method shows the best resolution, and its *DR* value is almost 50% smaller than that for random-selection.

Next in Table 2, we present simulation results for the six largest ISCAS-89 benchmarks. Since the simulation time is very high, we use only 128 pseudorandom patterns for each BIST session. We use a primitive-polynomial LFSR of degree 16 to create partitions. In order to make an objective comparison, we use the same number of partitions for both methods. In Columns 6 and 7, we list the diagnostic resolution obtained for the two methods with the post-processing pruning technique proposed in [7].

Two-step partitioning provides greater diagnostic accuracy than random selection for every circuit. For the

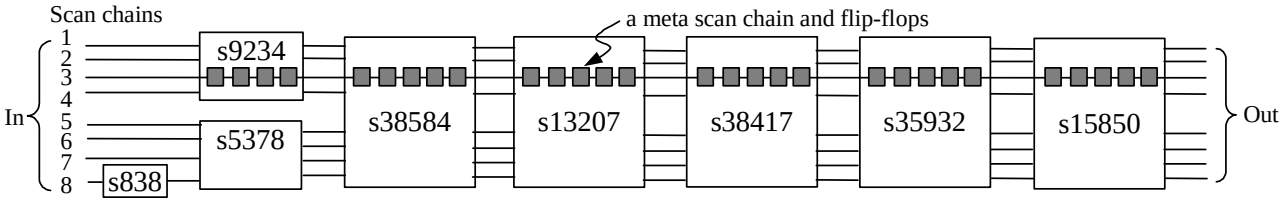


Figure 4. Scan chain configuration for ISCAS-89 modules in d695.

Number of partitions	*DR* (Interval-based)	*DR* (Random-selection)	*DR* (Two-step)
1	3.06	5.79	—
2	3.28	3.48	1.87
3	2.23	2.74	1.52
4	2.13	2.16	1.18
6	1.82	1.46	0.76
8	1.82	1.16	0.54

Table 1. Diagnostic resolution obtained for s953 with varying number of partitions and different partitioning schemes.

Circuit	No. of partitions	No. of groups in one partition	*DR* without pruning		*DR* with pruning	
			Random-selection	Two-step	Random-selection	Two-step
s9234	48	6	2.84	2.28	1.76	1.44
s13207	128	8	3.3	3.2	0.46	0.42
s15850	128	8	7.25	6.43	4.33	3.96
s35932	128	16	3.67	1.25	1.77	0.87
s38417	128	16	5.93	1.23	2.36	0.55
s38584	128	16	3.11	2.85	1.61	1.56

Table 2. Diagnostic resolution of the six largest ISCAS-89 benchmarks with the two partitioning schemes.

larger circuits, the *DR* values are reduced by as much as 80%. The pruning technique of [7] can be combined with two-step partitioning to obtain even higher diagnostic accuracy. We also note that the *DR* values here are larger than those obtained by random error injection using a small number of errors. This is because in a real circuit, some faults may cause a large number of failing scan cells that make partitioning and pruning less effective.

5 Application of Two-Step Partitioning to Core-Based SOCs

Fault diagnosis in SOCs has received relatively little attention in the literature, one reason being the lack of availability of internal structural information about intellectual property cores. Here we consider the problem of identifying the error-capturing scan cells in the embedded cores when a test access mechanism (TAM) such as a TestRail [10] is used for test application.

One possible diagnosis approach is to design a test bus as a TAM and then use a scan cell selection mechanism to identify failing scan cells for one core at a time, as proposed in Sections 2 and 3. However, it suffers from the drawback that since each core may have its own unique scan chain architecture, a single scan-BIST configuration cannot accommodate all the cores in the SOC. In addition, this method may require frequent reloading of test patterns to the ATE for the different cores.

An alternative is to use a TestRail and a daisy-chain architecture [10], where "meta" scan chains on the SOC are threaded through the internal scan chains of the embedded cores. Test patterns are transported to the cores and the test responses are transported from the cores using the meta scan chains in a single test session. Test application continues until a core runs out of test patterns. This core is then by-passed and the process repeats for other cores until all the cores run out of test patterns.

Two-step partitioning is especially suitable for fault diagnosis in SOCs with a daisy-chain architecture. Spot defects on an integrated circuit usually affect a small area of the die, implying that only a small number of cores are likely to be faulty during a test session. If we assume that only one core in the SOC contains failing scan cells, then only a small segment of the meta scan chain contains failing scan cells. As result, the interval-based partitioning approach is especially attractive because the segments containing failing scan cells are clustered within a few small groups when the meta scan chains are partitioned.

In order to highlight the advantages of the proposed diagnosis approach, we present results for two SOCs. The first SOC is crafted by stitching together the six largest ISCAS-89 benchmarks, and assuming that a single meta scan chain is threaded through the internal scan chains of the cores. The second SOC is a variant of d695 from the ITC'02 SOC Test benchmarks [11]. We only consider the full-scan ISCAS-89 modules in d695. We use a daisy-chain architecture with an 8-bits-wide TAM. The scan chains in the cores are reorganized to construct 8 balanced meta scan chains on the SOC, and the cores are daisychained as shown in Figure 4. For the sake of illustration, we show some scan cells in one meta scan chain using shaded squares. We are unable to use the other ITC'02 benchmarks because they do not contain sufficient information for carrying out fault simulation.

For each set of experiments, we assume that only one core contains failing scan cells. For each core that is assumed to be faulty, we inject 500 single stuck-at faults and calculate the corresponding *DR* value. We note that the effect of multiple faults can be viewed similarly with that of single fault, as discussed in Section 3.

A total of 8 partitions are used for both SOCs. To determine the number of groups in a partition, our strategy is to use more groups on the longer meta scan chains. The first SOC contains a rather long meta scan chain so we use

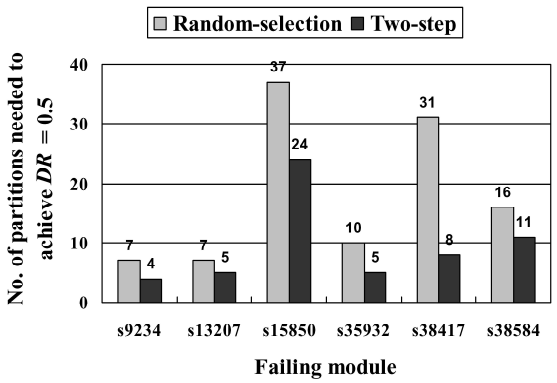


Figure 5. No. of partitions required to attain a *DR* value of 0.5 for different failing modules in the SOC with a single scan chain.

32 groups for each partition in this case. For the second SOC, the scan chains are relatively shorter, hence and the number of groups for each partition is set to 8.

We compare random-selection and two-step partitioning for the two SOCs, and present simulation results in Table 3 and 4. For the two-step method, only the first partition is interval-based. It can be seen that in all cases, the two step method outperforms the random-selection method significantly. In some cases, a 10X improvement is achieved. Even with pruning, interval-based partitioning performs significantly better.

The diagnosis time is determined to a large extent by the number of partitions required to achieve a desired *DR* value. Viewed from another perspective, the increase in diagnostic accuracy implies shorter diagnosis time and a smaller number of partitions required to obtain a desired resolution. In Figure 5, we show the number of partitions needed to obtain a *DR* value of 0.5 (without pruning) with the two partitioning methods for the first SOC with a single scan chain. We observe that the two-step approach requires a smaller number of partitions than the random selection method with a corresponding reduction in diagnosis time.

Failing module (core)	*DR* without pruning		*DR* with Pruning	
	Random-selection	Two-step	Random-selection	Two-step
s9234	0.227	0.01	0	0
s13207	0.166	0.047	0	0
s15850	4.24	0.4	1.259	0.117
s35932	1.02	0.096	0	0
s38417	3.881	0.556	1.148	0.228
s38584	2.805	0.977	0.253	0.094

Table 3. SOC diagnostic resolution: single scan chain.

Failing module (core)	*DR* without pruning		*DR* with pruning	
	Random-selection	Two-step	Random-selection	Two-step
s9234	0.538	0.08	0	0
s13207	1.224	0.80	0.043	0.043
s15850	1.396	0.677	0.224	0.119
s35932	5.712	2.227	3.932	1.524
s38417	10.291	2.777	3.667	1.000
s38584	0.805	0.283	0.029	0.013

Table 4. SOC diagnostic resolution: multiple scan chains.

6 Conclusions

We have presented a new two-step partition-based method for determining failing scan cells in scan-BIST. First, an interval-based partitioning scheme is used to generate a small number of partitions. The remaining partitions are then created using traditional random-selection partitioning. We have shown that two-step partitioning can be implemented with a small amount of hardware. Experimental results for the ISCAS-89 benchmarks show that two-step partitioning offers higher diagnostic resolution than random-selection partitioning. We have shown that this approach is especially suitable for identifying failing scan cells in SOCs with a daisy-chain test access architecture. Experimental results on diagnostic resolution for two SOCs crafted from the ISCAS-89 benchmarks show that interval-based two-step partitioning significantly outperforms random-selection partitioning.

References

[1] P. Song, F. Motika, D. R. Knebel, R. F. Rizzolo and M. P. Kusko, "S/390 G5 CMOS Microprocessor Diagnostics", *Proc. Int. Test Conf.*, pp. 1073-1082, 1999.

[2] J. Ghosh-Dastidar, D. Das and N. A. Touba, "Fault Diagnosis in Scan-based BIST using Both Time and Space Information", *Proc. Int. Test Conf.*, pp. 95-102, 1999.

[3] T. R. Damarla, C. E. Stroud and A. Sathaye. "Multiple Error Detection and Identification via Signature Analysis", *JETTA*, vol. 7, pp. 193-207, 1995.

[4] C. Liu, K. Chakrabarty and M. Goessel, "An Interval-Based Diagnosis Scheme for Identifying Failing Vectors in a Scan-BIST Environment", *Proc. DATE Conf.*, pp. 382-386, 2002.

[5] J. Rajski and J. Tyszer, "Diagnosis of Scan Cells in BIST Environment", *IEEE Trans. Comp.*, vol. 48, pp. 724-731, July, 1999.

[6] J. Ghosh-Dastidar and N. A. Touba, "A Rapid and Scalable Diagnosis Scheme for BIST Environment with a Large Number of Scan Chains", *Proc. VTS*, pp. 79-85, 2000.

[7] I. Bayraktaroglu and A. Orailoglu, "Improved Fault Diagnosis in Scan-based BIST via Superposition", *Proc. DAC*, pp. 55-58, June 2000.

[8] I. Bayraktaroglu and A. Orailoglu, "Deterministic Partitioning Techniques for Fault Diagnosis in Scan-Based BIST", *Proc. Int. Test Conf.*, pp. 273-282, Oct 2000.

[9] Y. Wu and S. Adham, "BIST Fault Diagnosis in Scan-Based VLSI Environment", *Proc. Int. Test Conf.*, pp. 48-57, 1996.

[10] E. J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg, and C. Wouters, "A Structured And Scalable Mechanism for Test Access to Embedded Reusable Cores", *Proc. Int.Test Conf.*, pp. 284-293, 1998.

[11] E. J. Marinissen, V. Iyengar and K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs", *Proc. Int. Test Conf.*, 2002, to appear.