Deterministic Partitioning Techniques for Fault Diagnosis in Scan-Based BIST

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

A deterministic partitioning technique for fault diagnosis in Scan-Based BIST is proposed. Properties of high quality partitions for improved fault diagnosis times are identified and low cost hardware implementations of high quality deterministic partitions are outlined. The superiority of the partitions generated by the proposed approach is confirmed through mathematical analysis. Theoretical analyses, worst case bounds, and experimental simulation data all confirm the superiority of the proposed deterministic approaches.

1. Introduction

Built-in self-test (BIST) is currently utilized in state-of-the-art designs both for improving test quality and for reducing test development and application cost. While BIST provides sizable benefits, a limitation in its further adoption as the main test methodology is inherent fault diagnosis challenges. While BIST reduces test application cost appreciably, fault diagnosis in a BIST environment is problematic since only limited information is available in a compact signature.

Until recently, research on diagnosis through BIST has concentrated on identifying methods for extracting information possibly embedded in the signatures. McAnney and Savir have shown that through analysis of the signature contents, it is possible to identify failing test vectors [6]. As the number of failing test vectors increases, the complexity of this scheme increases and its identification capability diminishes due to the aliasing effects of multiple failing vectors. They have further proposed a method that is capable of identifying increased number of failing test vectors through utilization of cyclic registers [8]. Stroud and Damarla have proposed a technique where the characteristic polynomial is a factored polynomial and show that utilization of non-primitive polynomials may reduce aliasing probability for multiple errors [9]. Aitken and Agarwal have proposed a method in which the quotient instead of the signature is utilized through a fault-free sequence generator [1]. Even though the scheme is capable of providing improved diagnostic capability by utilizing increased information, its hardware requirements are substantially higher than regular BIST. All schemes outlined above aim at identifying a set of failing test vectors. However, the identification capability of such schemes diminishes as the number of failing vectors increases. Yet fault effect manifestation cannot be typically limited to a few vectors. Excluding pseudo-random resistant faults, most faults are detectable through a fairly large number of vectors. Such schemes suffer consequently if applied in realistic test environments.

Recently, Rajski and Tyszer have proposed a scan partitioning based approach to the fault diagnosis problem in scan-based BIST [7]. Therein the results of the application of the same test are repeatedly partitioned, observed, and compared to the corresponding fault-free signature for each partition. A fault-free signature implies that all scan cells in the corresponding partition are fault free. Successive partitioning of the scan cells into distinct partitions helps eventually sieve all fault-free cells. By utilizing such partitioning schemes, diagnosis time can be usually reduced to less than a tenth of the time required by a scheme that would instead observe the output of each cell individually. Refinements of this partitioning technique have been proposed that utilize the superposition principle to further reduce diagnosis time [3]. Incorporation of the superposition principle into partitioning-based diagnosis approaches further reduces diagnosis time to less than half of the time attained by [7].

While partitioning-based schemes provide effective fault diagnosis in a scan-based BIST environment, generation of the partitions using pseudo-random approaches eliminates the predictability of the diagnosis results. Both the parameters of the LFSR utilized for partitioning and the location of the faulty cells affect fault diagnosis time. Furthermore, randomness of the partitions generated by the LFSR diminishes the possibility of incorporating design specific information into the diagnosis procedure.

LFSR-generated sequences have been widely researched to enable generation of test patterns that match specific test needs of designs [2, 4, 5]. The schemes proposed though require either mapping hardware to map generated patterns to desired ones [2] or intractable computation in order to determine the parameters of the LFSR [4, 5]. Since matching test needs of circuits is analogous to matching partitioning needs of specific designs or fault models, attempting to adopt prior research results into the partitioning problems posed by diagnosis does not evince a significant likelihood of yielding desirable low cost solutions.

On the other hand, deterministic partitioning of scan cells does not similarly suffer, even though the difficulty of identifying cost-effective hardware implementations with superior partitioning leading to reduced diagnostic time may be thought to doom the adoption of such methods. Nonetheless, we explore, identify, and outline in this paper the existence of deterministic partitioning schemes, amenable to low cost hardware implementations and exhibiting superior diagnostic capabilities.

Not only does the proposed deterministic approach consistently exhibit superior results, but furthermore it achieves such results with high levels of predictability. Even though diagnosis times attained by LFSRs typically cluster, nonetheless, there consistently exist LFSRs which show significant variability and consequent appreciable increase in diagnosis time. We examine in this paper the behavior of LFSRs and outline the mathematical formulation that explains the existence of such highly variable degradations. Yet even if precise conclusions were to be drawn in regards to the causes of such variable behavior, incorporating such heightened sensitivity to LFSR selection typically falls outside established industrial design flows and practices. An invaluable characteristic of the proposed deterministic partitioning approaches lies in their ability to incorporate design and fault information; such incorporation may further help reduce diagnosis time and model complex emerging fault models.

The next section of the paper reviews the fundamentals of partitioning-based diagnosis. Section 3 motivates the necessity of exploring deterministic partitioning schemes by examining the diagnostic quality of partitions. Section 4 illustrates mathematical and implementation challenges inherent in constituting a deterministic scheme and proceeds to outline both analytic and low cost hardware implementation solutions to the problem of deterministic partitioning. While section 5 provides a simulation-based comparison of LFSR-based and deterministic partitioning approaches, section 6 outlines worst case fault diagnosis analysis time for them both. Conclusions are drawn in section 7.

2. Preliminaries

While various attempts have been made to determine failing test vectors in a scan-based BIST environment, the diminution of their identification capability in the face of a large number of failing vectors dooms their applicability in current industrial designs. As the number of failing scan cells (scan cells that receive incorrect information) tends to be much smaller than the number of failing vectors, diagnosis schemes based on determination of failing scan cells have been shown to provide economical solutions for today's typical designs [7]. A partitioning-based approach for identification of failing scan cells provides low-overhead hardware support for diagnosis of industrial designs with minimal ATE intervention.

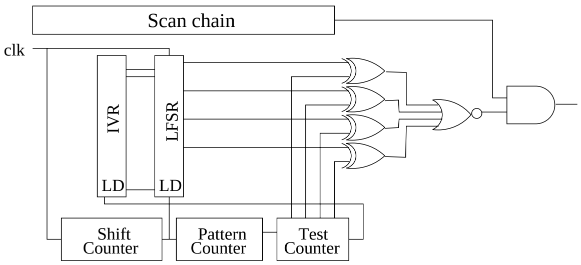


Figure 1. LFSR-based scan cell selector [7]

In a partitioning-based diagnosis scheme, scan cells are successively grouped into a set of non-overlapping partitions, each set constituting a partition group. The observation of the signatures corresponding to each partition can provide valuable information in diagnosis, since each fault-free signature indicates that all the cells in the corresponding partition are fault free. Nonetheless, a single partition group is inadequate in identifying with certainty the failing scan cells, as all cells in a partition that exhibits a failing signature constitute a candidate set of culprits. Consequently, additional applications of the same test set, yet with differing partition elements, need to be used to provide additional snapshots of failing scan cells with consequent refinement of the candidate failures. Each of these applications of the same test set, with its repartitioning of the scan cells, constitutes a partition group.

Figure 1 depicts a possible scan cell partitioning hardware implemented using an LFSR and an Initial Value Register (IVR) [7]. While the LFSR is loaded from the IVR for generation of each partition in a particular partition group, at the end of the generation of a partition group, the IVR is updated with the current state of the LFSR. The test counter value is compared to an arbitrarily selected set of r outputs of the LFSR; compaction of the output of the corresponding scan cell occurs upon a match. Since the test counter has a unique value for each partition, the partitions in each partition group are distinct. Updating the IVR with the current state of the LFSR at the end of each partition group guarantees distinctness of the partition groups as LFSR generated sequences do not repeat.

Figure 2 provides a pseudo-code of the diagnosis procedure for LFSR-generated partitions. In this pseudo-code,

```
for each partition group c
  for each partition b in c
    shift_counter = 0, LFSR = IVR
    while shift_counter != N
      if r bits of LFSR = b
        compact the current output
      shift_counter++
  IVR = LFSR
```

Figure 2. LFSR-based partitioning procedure

```
for each partition group c
  for each partition b in c
    shift_counter = 0, i = 0
    while shift_counter != N
      if P(c,b,i) = shift_counter
        compact the current output
        i++
      shift_counter++
```

Figure 3. Generic partitioning procedure

N and b correspond to the size of the scan chain and the value of the test counter, respectively. While the pseudo-code of figure 2 is specific to LFSR-based partitions, a generic pseudo-code for the diagnosis procedure scheme can be attained by modifying the cell selection logic. Such an approach is shown in figure 3, where $P(c, b, i)$ denotes the scan cell identifier of the i^{th} element of the b^{th} partition of the c^{th} partition group. Correct functionality of the diagnosis procedure based on the latter pseudo-code requires that $P(c, b, i)$ be monotonic in parameter i. While such a requirement imposes a constraint on the partitioning scheme, an implementation that is capable of handling non-monotonic $P(c, b, i)$ would require significantly higher area overheads as it would necessitate storing and subsequent reordering of the elements of the partitions.

3. Diagnostic Quality of Partitions

While partitions inside a partition group do not overlap, partitions that belong to distinct partition groups overlap in various number of scan cells depending on how the partitions are generated. The overlap among partitions effects diagnostic time as increased overlap between two partitions reduces additional information attainable in case both partitions happen to be fault-free. A partitioning scheme that is polluted by an excessive number of highly overlapping partitions unnecessarily lengthens the diagnosis procedure. In the following subsections, even though we analytically show that the average overlap among partitions is constant, variations in the number of overlapping cells effect fault diagnosis; the higher the variation in the overlap, the longer the diagnosis procedure takes.

3.1. Expected Overlap Value Analysis

The expected amount of overlap for two partitions in distinct partition groups can be analytically determined under the simplifying assumptions of random generation and equal sized partitions. Expected overlap can be obtained by summing the index-weighted variant probabilities of overlap. In the case of partitioning of N cells into P groups, this reduces to:

$$\mathcal{E}_{overlap}(N, P) = \sum_{n=0}^{N/P} n\mathcal{P}_{overlap}(n) \tag{1}$$

The probability of n overlap, $\mathcal{P}_{overlap}(n)$, can be determined by calculating the following quantities.

- The number of possible overlaps, $\binom{N/P}{n}$
- The number of possible non-overlaps, $\binom{N-N/P}{N-n}$
- The number of possible partitions, $\binom{N}{N/P}$

Multiplication of the first two quantities provides the number of partitions that overlap in n elements. Dividing this number by the total number of partitions results in an expression for the probability of n overlap. Substituting the resultant probability, $\mathcal{P}_{overlap}$, in equation 1 results in:

$$\mathcal{E}_{overlap}(N, P) = \frac{\sum_{n=0}^{N/P} n\binom{\frac{N}{P}}{n}\binom{N-\frac{N}{P}}{\frac{N}{P}-n}}{\binom{N}{\frac{N}{P}}} \tag{2}$$

The combinatorial formulation in equation 2 can be resolved by utilizing the following identity.[1]

$$\sum_{k=0}^{m} k\binom{p}{k}\binom{q}{m-k} = p\binom{p+q-1}{m-1} \tag{3}$$

Substituting $\frac{N}{P}$, n, $\frac{N}{P}$, and $N - \frac{N}{P}$ for m, k, p, and q, respectively, yields

$$\mathcal{E}_{overlap}(N, P) = \frac{N}{P^2} \tag{4}$$

The identical result is obtained even if nonrandom partitions are utilized as long as partitions in a particular group are assumed to cover all scan cells and be nonoverlapping. This can be easily shown by observing that the expected size of a partition is N/P. The overlap of a partition to the union of all the sets in a distinct partition group is the partition size, since the partition group has to cover all N scan elements. The expected overlap to one partition in the partition group consequently is the expected partition size times the probability of selecting that partition, i.e. $(N/P)(1/P) = N/P^2$.

Since the expected overlap value is identical for any partition group generation scheme and as a matter of fact exactly equal to that of randomly generated partitions, the

[1] A proof of the identity in equation 3 is given in the appendix.

LFSR width	Chain Length	Partitions	Expected Overlap	RMS deviation
14	1024	16	4	2.72
14	1024	8	16	6.15
15	1024	8	16	5.25
14	2048	8	32	11.90

Table 1. RMS deviation of partition overlaps

LFSR width	Chain Length	Partitions	Error Count	Correlation
14	1024	16	15	0.6986
14	1024	8	7	0.4804
15	1024	8	7	0.4648
14	2048	8	7	0.5785

Table 2. Correlation between RMS deviation and expected fault diagnosis time

cause of possibly differing partition quality needs to be searched within higher order measures, such as overlap deviations. Overlap deviations can be converted into a single quality measure by calculating the Root Mean Square (RMS) of the deviations. While deterministic partitions can be made to exhibit zero RMS as long as the partition overlap equals uniformly the expected overlap value, RMS for LFSR-generated partitions can be shown to be non-zero and varying. We calculate the RMS value for the partitionings that are generated by all primitive polynomials of degree 14 and 15 (756 and 1800 primitive polynomials, respectively). Table 1 provides parameters of the LFSRs utilized for partitioning, the expected overlap, and the RMS deviation in overlap values.

The table indicates that the overlap among partitions varies from the expected value significantly with RMS deviation ranging between 33% to 68% of the expected overlap. The RMS deviation numbers reported in table 1 constitute the average value of the deviations for all primitive polynomials of the corresponding degree. While the RMS values indicate significant overlap deviations, it is unclear up-front whether such deviations translate into diagnostic time degradations. We perform correlation analysis in the next subsection in order to validate the diagnostic quality of the RMS measure.

3.2. Correlation Analysis

For each simulation parameter in table 1, the diagnosis procedure is simulated in order to find the expected fault diagnosis time for each primitive polynomial. Furthermore, the correlation between the RMS overlap deviation and the expected fault diagnosis time is calculated. The results are reported in table 2.

Correlation results reported in table 2 exclude the primitive polynomials that result in extremely high diagnosis times; including such polynomials would have only increased correlation levels. Example cases of anomalous diagnosis times and their associated analyses are provided in section 6. The results indicate that there indeed exists a significant, positive correlation between the expected fault diagnosis time and the RMS partition quality measure proposed. As reduction of overlap between partitions evidently results in reduced diagnosis time and as overlap variation can be minimized through deterministic approaches only, we examine in the next section the generation of deterministic partitions that achieve an RMS value of zero and thus exhibit superior diagnostic power over LFSR-based schemes.

0	**1**	**2**	**3**	**4**	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	6	12	18	24	5	11	17	23	**4**	10	16	22	**3**	9	15	21	**2**	8	14	20	**1**	7	13	19
0	11	22	8	19	5	16	**2**	13	24	10	21	7	18	**4**	15	**1**	12	23	9	20	6	17	**3**	14
0	16	7	23	14	5	21	12	**3**	19	10	**1**	17	8	24	15	6	22	13	**4**	20	11	**2**	18	9
0	21	17	13	9	5	**1**	22	18	14	10	6	**2**	23	19	15	11	7	**3**	24	20	16	12	8	**4**

Table 3. Deterministic partitioning of 25 cells

4. Construction of Deterministic Partitions

A partitioning with zero RMS overlap deviation has to have uniformly minimal overlap with all partitions, imposing a strict constraint on the structure of the partitions. In the case of number of partitions equaling partition size, a partition has to have a single overlap to *each* partition inside *every other* partition group. This constraint limits sharply the number of attainable partition groups, and makes their identification and construction nontrivial.

A deterministic partitioning that satisfies the aforementioned constraints for a scan chain size of 25 is provided in table 3. The scan chain is partitioned into 5 partition groups; each row in the table corresponds to a partition group with 5 partitions. It can easily be verified that the number of overlapping cells for the partitions in table 3 is consistently equal to 1 for partitions in distinct partition groups and to 0 for partitions in the same group. The minimal overlap property can be traced at least for a single partition by observing in bold all the elements of the first partition in the first partition group in this table.

Even though such partitions have the minimum possible overlap, their hardware generation is highly challenging. Examination of the relationships across groups yields the recurrence relation in equation 5, while equation 6 can be used to denote a simple definition of the initial partition group. In these and the following equations, c, b, i, and S (equal to 5 in the example of table 3) correspond to the partition group number, the partition number, the location inside a partition, and partition size, respectively. A numbering convention wherein c, b, and i are assumed to vary between 0 and $S - 1$ inclusive is utilized throughout. $P(c, b, i)$ indicates the numeric identifier of a scan cell in location i of partition b in partition group c, as noted in section 2.

$$\begin{aligned} P(c+1,b,i) &= P(c,(b+i)\,mod\,S,i) && (5)\\ P(0,b,i) &= bS+i && (6) \end{aligned}$$

While these equations are straightforward, generation of the partitions directly through these equations requires high area overhead due to recursion; a direct implementation of a recurrence relation requires storage of the previous result. In the following subsections, alternative recurrence relations amenable to nonrecursive solutions are derived for these partitions. Various hardware implementations for such formulations are provided as well.

4.1. Implementation Issues

A simple hardware implementation for the partition groups in table 3 requires a solution of the recurrence relations in equations 5 and 6. Examination of the stride characteristics within partitions provides equation 7, while examination of the stride across partitions provides equation 8.

$$\begin{aligned} P(c,b,i+1) &= (P(c,b,i)+cS+1)\,mod\,S^2 && (7)\\ P(c,b+1,i) &= (P(c,b,i)+S)\,mod\,S^2 && (8) \end{aligned}$$

Now that inductive relationships are established for all three arguments of P, a nonrecursive solution can be established by embedding the induction effects, resulting in equation 9.

$$P(c,b,i) = ((ci+b)S+i)\,mod\,S^2 \qquad (9)$$

The existence of partitions with minimal overlap, as shown in table 3, is not coincidental. The following theorem shows that partition groups generated by equation 9 always exhibit the minimal overlap of 1 for any prime number S.

Theorem: Partitions generated by equation 9 have an overlap of 1 whenever they are in distinct partition groups and have no overlap whenever they are in the same partition group, for S prime and c, b, and $i < S$.

Proof: In order for the elements of two partitions, $P(c_1,b_1,i_1)$ and $P(c_2,b_2,i_2)$, to overlap, the following equality has to be satisfied:

$$P(c_1,b_1,i_1) = P(c_2,b_2,i_2) \qquad (10)$$

Since $S > |i_1 - i_2|$, equation 10 holds iff $i_1 = i_2 = i$. Therefore, the overlap condition reduces to:

$$(c_1 i + b_1)S \overset{mod\ S^2}{=} (c_2 i + b_2)S \qquad (11)$$

which can be simplified by dividing by S to:

$$(c_1 - c_2)i \overset{mod\ S}{=} b_2 - b_1 \qquad (12)$$

For partitions in different partition groups, i.e. $c_1 \neq c_2$, only two cases need to be analyzed:

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19
4	20	21	22	23	24

Table 4. Remainder and quotient groups

- $b_1 = b_2$: has a unique solution, $i = 0$.
- $b_1 \neq b_2$: a unique solution exists for S prime.

For partitions inside the same partition group, i.e. $c_1 = c_2$, overlap necessitates equality of b_1 and b_2; a partition can only overlap with itself inside a partition group.

We have shown that partitions within distinct partition groups have an overlap of 1 and those within the same partition group do not overlap. While partitions generated by equation 9 fulfill the minimal overlap requirement and have an RMS overlap deviation of *zero*, they do not fulfill the monotonicity requirement discussed in section 2.

Since there exists no restriction on the order of the numeric identifiers inside a partition, they can easily be reordered to satisfy the monotonicity requirement. However, reordering these identifiers inside partitions destroys the regular recurrence relations. In order to generate partitions with low cost hardware implementation, we investigate the structure of the construction of the partitions that are generated by equation 9 and utilizing the structure we provide an alternative partitioning for which the increasing order property holds. Determination of the structure of partitions generated by equation 9 requires grouping the scan cell identifiers in terms of their remainder and quotient to S. Table 4 provides such a grouping of S^2 identifiers when S equals 5. In this table, columns and rows correspond to the remainder and to the quotient groups, respectively.

Notice that all the partitions in table 3, except the ones in the first row, include 1 element from each of the remainder and quotient group. The elements in each partition are always composed of the elements of the remainder groups $0, 1, 2, \cdots, S-1$. The elements of the first partition of each partition group can be generated by visiting the quotient groups in increments of $0, 1, 2, \cdots, S-1$. Finding the quotient groups for higher numbered partitions is trivial as the recurrence relation in equation 8 implies that the quotient groups of the elements in partition $b+1$ exceed by one the elements in partition b. Not only the partition in table 3, but also all such partitions that are generated by equation 9, for S prime, exhibit the same regular structure. We denote the class of partitions that are generated by equation 9 as *Remainder Uniform Partitions*.

The observations in the previous paragraph enable the construction of partitions with the monotonically increasing

0 5 10 15 20	1 6 11 16 21	2 7 12 17 22	3 8 13 18 23	4 9 14 19 24
0 6 12 18 24	1 7 13 19 20	2 8 14 15 21	3 9 10 16 22	4 5 11 17 23
0 7 14 16 23	1 8 10 17 24	2 9 11 18 20	3 5 12 19 21	4 6 13 15 22
0 8 11 19 22	1 9 12 15 23	2 5 13 16 24	3 6 14 17 20	4 7 10 18 21
0 9 13 17 21	1 5 14 18 22	2 6 10 19 23	3 7 11 15 24	4 8 12 16 20

Table 5. Quotient Uniform Deterministic partitioning of 25-cell scan cells

0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15
0 4 8 12	1 5 9 13	2 6 10 14	3 7 11 15
0 5 10 15	1 4 11 14	2 7 8 13	3 6 9 12
0 6 11 13	1 7 10 12	2 4 9 15	3 5 8 14
0 7 9 14	1 6 8 15	2 5 11 12	3 4 10 13

Table 6. Deterministic partitioning of 16 cells

order property by interchanging the quotient for the remainder in the previous paragraph. The new class of partitions are denoted analogously as *Quotient Uniform Partitions*. A consequent partitioning, in the case of $S = 5$, is provided in table 5. A comparison of tables 3 and 5 reveals that except for the first group, both partitionings are identical, yet utilize different orderings of partition groups and partitions. Consequently, the total number of partition groups in both tables that satisfy the minimal overlap property is 6. Indeed, exhaustive computer simulations for S up to 7 indicate that for a scan size of S^2 there exists a maximum of $S + 1$ partition groups satisfying the minimal overlap criterion. The same number of minimally overlapping partition groups also exists for non-prime partition sizes. However, no simple recurrence relation that yields low cost hardware implementations can be identified for non-prime partitions. Table 6 provides 5 minimally overlapping partition groups for $S = 4$.

The solution of the recurrence relation for Quotient Uniform Partitions provides the following result:

$$P(c, b, i) = (ci + b) \; mod \; S + iS \qquad (13)$$

Equation 13 can be utilized in the pseudo-code in figure 3 for hardware implementation. The result of such an implementation is shown in figure 4. However, such an implementation still requires significant hardware overhead, especially for the implementation of $P(c, b, i)$; two multipliers, two adders and a modulo operator are required for the implementation of $P(c, b, i)$. An alternative, improved implementation is shown in figure 5, which is based on the pseudo-code of figure 6, utilizing the difference of $D = P(c, b, i + 1) - P(c, b, i)$, where $P(c, b, i)$ is defined by equation 13. The difference, D, reduces to $c + S$ unless $(c(i + 1) + b) \; mod \; S < (ci + b) \; mod \; S$, in which case it reduces to c. This implementation requires significantly fewer hardware components compared to the implementation in figure 4.

In the modified implementation, both the register *Add* and the counter *ShiftIncrement* are set to the first element of the current partition, which is always equal to the partition number. Register *Add* continuously holds the current remainder group of the current element in the partition subsequently; the current remainder group is updated whenever that element is reached in the scan chain. The *Shift-Increment* counter is reloaded with the difference between $P(c, b, i + 1)$ and $P(c, b, i)$ upon reaching zero. After the whole scan chain is shifted, the process restarts from the beginning. At the end of the test application session, the signature is shifted out, the partition count is incremented, and the process repeated.

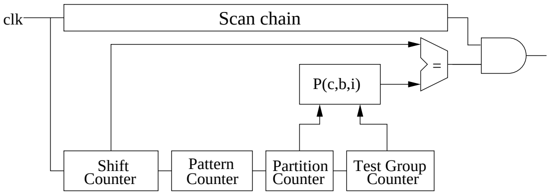


Figure 4. Absolute value based partitioning hardware

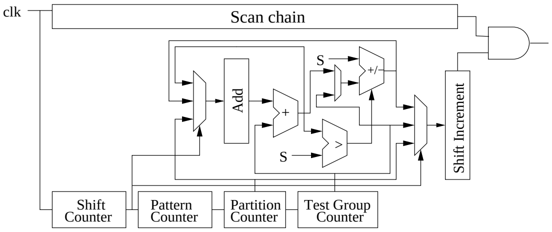


Figure 5. Successor difference based partitioning hardware

The deterministic partitioning hardware depicted in figure 5 requires one $log(S)$ bit register, two $log(S) + 1$ bit adders and comparators, one $log(S) + 1$ bit counter, and 3 multiplexers in addition to the hardware requirements of the BIST hardware. The hardware requirement for determinis-

```
for each partition group c
  for each partition b in c
    ShiftCounter = 0
    Add = b, ShiftIncrement = b
    while ShiftCounter != N
      Until ShiftIncrement = 0
        ShiftIncrement--, ShiftCounter--
      compact the output of the current cell
      Add = Add + c
      If Add >= S
        Add = Add -S
        ShiftIncrement = c
      Else
        ShiftIncrement = c + S
```

Figure 6. Successor difference based partitioning procedure

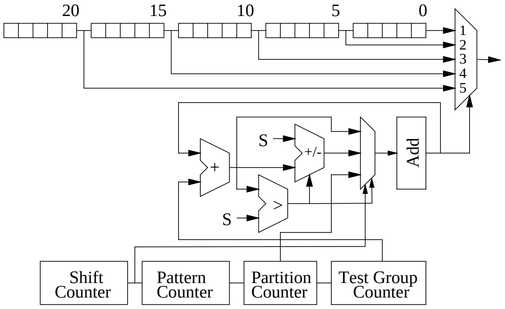


Figure 7. Tapped Partitioning

tic partitioning is slightly higher than that of LFSR-based partitioning, which requires one LFSR, one register and one $log(S)$ bit comparator. Even though the size of the LFSR and of the register are typically user defined, LFSRs of size 16 provide satisfactory results [7].

An alternative, tapped implementation strategy for partitioning is shown in figure 7. This implementation requires access to internal points of the scan chains and therefore introduces routing overhead. This implementation directly mimics the structure of the partitions. Access to the internal points of the scan chain provides direct access to the quotient groups and the shifting process provides access to the remainder groups. Since partitions have a single element from each remainder group, the output multiplexer can select the correct quotient group whenever necessary. For example, for the second partition of the third partition group, the multiplexer receives the select signals 3,1,4,2,5. At the first cycle, scan cell 10 is selected. After the shift operation, the content of scan cell 1 is shifted into scan cell 0 and is consequently selected for compaction by the multiplexer. Another shift operation shifts the content of scan cell 17 into scan cell 15 which is consequently selected by the multiplexer. At the end of 5 cycles, all necessary scan cell outputs for the partition are compacted at the output.

If extra signature compactors are available, signatures for all partitions in a test group can be produced in parallel by connecting the output of the last scan cell to the input of the first scan cell. Diagnosis time can be sharply increased thereupon, albeit at the expense of some hardware overhead. While it would seem at first sight that diagnosis time is reduced by a further factor of S, the actual reduction is only a factor of $S/2$, as the connection from the last scan cell to the first scan cell during diagnosis eliminates the capability of shifting in new patterns while shifting out results.

4.2. Arbitrary Partitioning of Scan Cells

For the partitions generated so far, the number of elements in the partitions is assumed to equal the number of partitions. Yet it is known [7] that the optimal partitioning requires the number of partitions to be one more than the number of error presumed and thus optimality necessitates the ability to handle arbitrary partitionings of the scan cells. In this subsection, we show that the properties developed in the previous subsection can be utilized to generate arbitrary sized partitions as long as the size of the partitions is chosen to be prime during partition generation.

	0	1	2	3
0	0	5	10	15
1	1	6	11	16
2	2	7	12	17
3	3	8	13	18
4	4	9	14	19

Table 7. Remainder and quotient groups for $S = 5$ and $P = 4$

0 6 12 18 4	5 11 17 3 9	10 16 2 8 14	15 1 7 13 19
0 7 14 16 3	5 12 19 1 8	10 17 4 6 13	15 2 4 11 18
0 8 11 19 2	5 13 16 4 7	10 18 1 9 12	15 3 6 14 17
0 9 13 17 1	5 14 18 2 6	10 19 3 7 11	15 4 8 12 16

Table 8. Deterministic partitioning of 20 cells

In order to generate P partitions of size S, we generate the quotient and remainder group with respect to S for all $P \times S$ elements in the scan chains. Table 7 provides such a partitioning for $P = 4$ and $S = 5$. In this table, rows and columns correspond to the remainder and quotient groups, respectively.

The partitions are generated by choosing one element from each row and column. The columns are traversed in increments of 1 and the rows are traversed in increments of $1, 2, 3$, and 4, generating 4 distinct partition groups. The first partition of the first partition group is constructed as $0, 6, 12, 18$, and 4. The second partition of the first group is generated as $5, 11, 17, 3, 9$. The first partition of the second group is generated as $0, 7, 14, 16, 3$. Notice that for all of the partitions in table 8 the columns are traversed in increments of 1, and the rows are traversed in increments of $1, 2, \cdots, S-1$.

In order to be able to partition N elements into P partitions, the partition size S needs to be the minimum prime

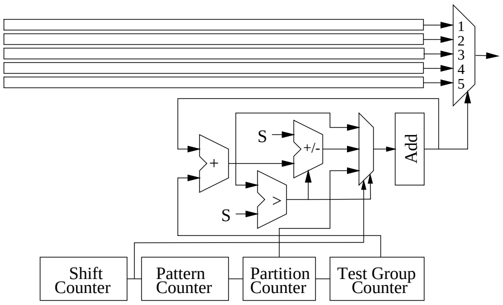
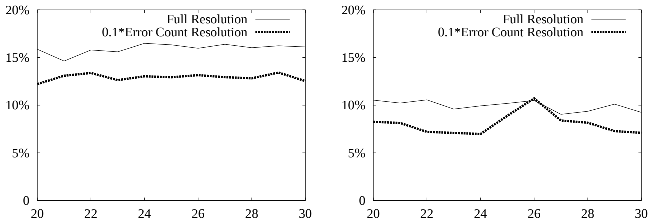
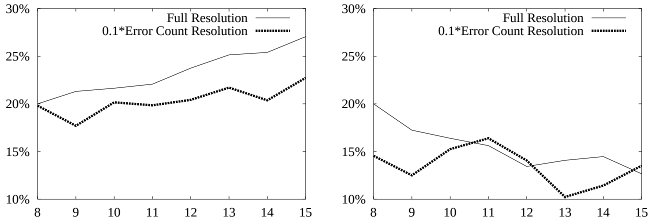


Figure 8. Multiple scan chain partitioning



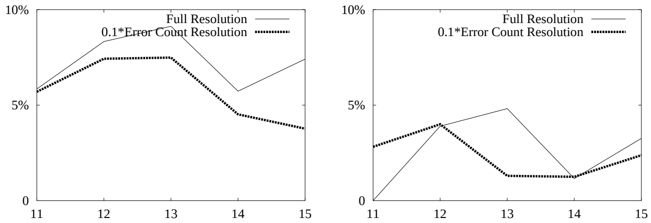
Without superposition With superposition

Figure 9. Improvements for $S = P = 31$



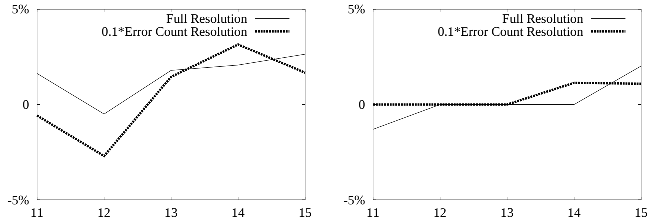
Without superposition With superposition

Figure 10. Improvements for $S = P = 17$



Without superposition With superposition

Figure 11. Improvements for $S = 67$**,** $P = 16$



Without superposition With superposition

Figure 12. Improvements for $S = 127$**,** $P = 16$

number satisfying the condition $P \times S \geq N$. While a simple recurrence relation can be found for the partitions in table 8, its implementation is not straightforward since the scan cell identifiers inside a partition do not consistently increase. Fortunately, the tapped implementation shown in figure 7 can be used in this case as well. The tapped implementation is especially useful in the case of multiple scan chains. Instead of using internal taps, multiple scan chains can be considered as a single chain for partitioning purposes, and each output can be utilized as an internal tap point to the scan chain. Figure 8 depicts such an implementation.

5. Simulation Results

In order to determine the effect of the deterministic partitioning schemes on the expected fault diagnosis time, we perform a set of simulation experiments. We compare our results to ones obtained by the LFSR-based fault diagnosis procedure suggested in [7]. We also compare our results to the ones obtained by utilization of the superposition principle as suggested in [3].

Simulations are performed on two prime number of partitions, 17 and 31, due to their proximity to 16 and 32; LFSR-based partitioning forces the number of partitions to be powers of two. For a fair comparison the number of partitions needs to be identical, yet the two schemes display conflicting requirements in this matter. The primes 17 and 31 are selected not only because they are adjacent to the corresponding powers of 2, but also because they end up bestowing a slight advantage on alternating schemes.

Figure 9 shows the improvement in diagnosis time for a deterministic partitioning of 31 partitions compared to a pseudo-random partitioning of 32 partitions. Similarly, figure 10 shows the improvement in diagnosis time for a deterministic partitioning of 17 partitions compared to a pseudo-random partitioning of 16 partitions. The figures show improvement for both full diagnostic resolution in which all failing cells are exactly identified and diagnostic resolution of $0.1 \times Error\ Count$[2]. The results of [7] have been utilized whenever available, as in the case of some of the diagnosis results for the resolution metric of $0.1 \times Error\ Count$; results have been generated otherwise by implementing the procedure outlined.

The results indicate that deterministic partitioning improves diagnosis time consistently. A few parameters effect the level of improvement: the improvement is high for full diagnostic resolution, without superposition, and for smaller partition sizes. While pseudo-random partitioning exhibits good diagnostic resolution up to a point easily, full resolution benefits significantly from deterministic techniques. Utilization of the superposition principle effectively removes the overlap between the partitions that are superposed and therefore reduces the improvement due to the minimal overlap property of deterministic partitions. However, the superposition principle is only useful for faults that do not create sequential behavior [3].

Additional simulation experiments were performed on non-prime number of partitions with results shown in figures 11 and 12. In this case, the number of partitions can be

[2] The diagnosis resolution metric of $0.1 \times Error\ Count$ originally suggested in [7] is defined as the time at which that many fault-free scan cells remain unsieved and still reside in the candidate failing scan cell set.

arbitrarily chosen, affording us the ability to undertake a set of comparisons using identical parameters.

While the simulation results indicate that in general deterministic partitions are better than the pseudo-random counterparts, the improvement in the case of non-prime partitions is attenuated. Results of the non-prime partitions, as was earlier observed in the case of prime partitions, indicate that as partition size increases, the effect of the varying overlap of LFSR-based partitions is reduced.

6. Worst Case Behavior Comparison

In this section, we perform a worst case diagnosis time analysis for both deterministic and LFSR-based partitioning, for the case of partition sizes equaling the number of partitions. In this case, as shown in section 3, there exists only one overlap between any two partitions in distinct partition groups, and no overlap within the partitions in the same partition group. Therefore, any pair of elements can cohabit in a partition only once.

Assume that there are n faults. Each fault-free cell has to reside in a partition that excludes all n faulty cells at least once in order to be able to achieve full diagnostic resolution. The fault free cell and a particular faulty cell can reside in the same partition only once; otherwise, the single overlap property between partitions would be violated. Since there are n faults in the system, a fault-free cell can reside within the same partition of a faulty cell at most n times. Therefore, at the worst case, achieving full diagnosis of n faults necessitates utilization of $n + 1$ partition groups.

For example, consider the partitioning in table 5. If the faulty cells are 5, 6, 7, and 8, detecting that cell 0 is fault-free will require five partition groups since cell 0 resides within the same partition as faulty cell 5, 6, 7, and 8 in the first partitions of partition groups 0, 1, 2, and 3, respectively. In the first partition of partition group 4, no faulty cell exists and therefore cell 0 can be then pronounced to be fault free.

The worst case analysis would become considerably more complicated if we allow the partition size to exceed the number of partitions. In this case, the number of overlapping elements between partitions can be greater than one and a faulty and fault-free cell can reside within the same partition multiple times.

In the case of LFSR-based partitioning, even if the number of partitions is equal to the size of the partitions, there can still be more than one overlap among the partitions. If two cells, one fault-free and one faulty, continuously reside in the same partitions, the fault-free cell cannot be pronounced fault-free, and full resolution can take indefinitely.

In LFSR-based partitioning, the least significant $\log b$ bits of the LFSR are compared to the current partition number, and upon equality, the current scan cell output is compacted. Assuming that the initial value of the LFSR is I, and the state transition matrix for the LFSR is A, the condition for two scan cells i and j to be on the same partition can be written as:

$$(A^i - A^j)I = \begin{bmatrix} \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} \log b \\ x \\ \vdots \\ x \end{bmatrix} \tag{14}$$

At the end of the partition group, the initial value of the LFSR is updated to $A^N S$ where N is the length of the scan chain. Therefore, two scan cells reside in the same partition for M test partition groups, if the following condition holds.

$$A^{kN}(A^i - A^j)I = \begin{bmatrix} \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} \log b \\ x \\ \vdots \\ x \end{bmatrix} \quad 0 \le k < M \tag{15}$$

While the condition in equation 15 identifies a worst case scenario, simulation over all primitive polynomials indicates that this condition effects a limited number of scan cell pairs and does not significantly contribute to the expected diagnosis time. As simulation results discussed in section 2 indicate that certain polynomials produce appreciably longer diagnosis times, the cause of the anomalous behavior needs to be sought elsewhere. Examination of the distribution of the overlap amount between the partitions indicates that the anomalous behavior is due to the lack of overlap among certain partitions. Lack of overlap among certain partitions implies a larger overlap among other partitions. A cell resides in the i^{th} partition if the value in the LFSR's least significant $log(b)$ bits at time T equals i. The partition that the same cell resides in the next partition group is determined by the state of the LFSR at time $T+N$. The problem arises whenever no cells of a partition can reside in a partition of a subsequent partition group. The states of an LFSR at time T and $T + N$ are correlated and the relation between them depends on the value of the N^{th} power of the state transition matrix as shown in equation 16. The anomalous behavior can be shown to be a function of the structure of A^N, consequently.

$$S(T + N) = A^N S(T) \tag{16}$$

The matrix A^N shown in figure 13 is obtained for a scan chain length of 128 and the primitive polynomial $7EAB_{hex}$ of degree 14. Assuming that the number of partitions is 4, i.e. $\log(b) = 2$, an analysis of the matrix A^N indicates that

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}$$

Figure 13. A state transition matrix raised to the scan chain length

0 and 2 at time T cannot be followed by 2 and 3 at time $T + 128$. Similarly, 1 and 3 at time T cannot be followed by 0 and 1 at time $T + 128$. Therefore, partitions 0 and 2 in a partition group do not overlap with partitions 2 and 3 in the subsequent partition group. A similar argument for partitions 1 and 3 also applies.

7. Conclusion

A deterministic partitioning technique for diagnosis of designs with scan-based BIST is proposed in this work. The deterministic technique is compared to that of LFSR-based techniques and the superiority of the deterministic technique is proven through both analysis of the partition overlaps and fault diagnosis simulations.

Though hardware generation of deterministic partitioning is highly challenging especially within low area overhead, the regular partition structures identified in this work enable such low cost hardware implementations. Further, the superior properties of the deterministic partition structures outlined in this work are proven analytically. The deterministic nature of the partitions also enables a straightforward worst case analysis, thus obtaining a strict upper bound on the diagnostic time. A corresponding analysis for LFSR-based partitions indicates that the worst case diagnosis time can be significantly longer, though such cases rarely occur, and that certain LFSRs exhibit appreciable diagnostic time degradation. The implementation regularity, the associated reduction in hardware overhead, the average diagnostic time superiority, and the imperviousness to diagnostic time deviations, all shown in this work, introduce deterministic partitioning as a powerful new BIST-based diagnosis tool, one whose importance is only bound to increase as its potential to incorporate design and fault effect information gets subsequently to be exploited.

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] S. Akers and W. Jansz. Test set embedding in a built-in self-test environment. In *International Test Conference*, pages 257–263, 1989.

[3] I. Bayraktaroglu and A. Orailoglu. Improved fault diagnosis in scan-based BIST via superposition. In *Design Automation Conference*, pages 55–58, 2000.

[4] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois. Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers. *IEEE Transactions on Computers*, 44:223–233, February 1995.

[5] M. Lempel, S. Gupta, and M. Breuer. Test embedding with discrete logarithms. *IEEE Transactions on CAD of Integrated Circuits and Systems*, 14:554–566, May 1995.

[6] W. H. McAnney and J. Savir. There is information in faulty signatures. In *International Test Conference*, pages 630–636, 1987.

[7] J. Rajski and J. Tyszer. Diagnosis of scan cells in BIST environment. *IEEE Transactions on Computers*, 48(7):724–731, July 1999.

[8] J. Savir and W. H. McAnney. Identification of failing tests with cycling registers. In *International Test Conference*, pages 322–328, 1988.

[9] C. E. Stroud and T. R. Damarla. Improving the efficiency of error identification via signature analysis. In *VLSI Test Symposium*, pages 244–249, 1995.

Appendix

In section 3, the following identity was utilized in order to derive the expected overlap among the partitions.

$$\sum_{k=0}^{m} k\binom{p}{k}\binom{q}{m-k} \quad = \quad p\binom{p+q-1}{m-1} \qquad (3)$$

A proof for this can be given as follows.

$$\begin{aligned}
(1+x)^p &= \sum_{k=0}^{p}\binom{p}{k}x^k \\
x\frac{d}{dx}(1+x)^p &= \sum_{k=0}^{p}k\binom{p}{k}x^k \\
x\frac{d}{dx}(1+x)^p(1+x)^q &= px(1+x)^{p+q-1} \\
\sum_{k=0}^{p}k\binom{p}{k}x^k\sum_{l=0}^{q}\binom{q}{l}x^l &= \sum_{n=0}^{p+q-1}p\binom{p+q-1}{n}x^{n+1}
\end{aligned}$$

Matching the coefficients of the x^m terms in the final equation by utilizing the binomial expansion yields the desired identity.