Compact Dictionaries for Diagnosis of Unmodeled Faults in Scan-BIST

Chunsheng Liu[£*], Kumar N. Dwarakanath[‡], Krishnendu Chakrabarty[†*] and Ronald D. (Shawn) Blanton[‡]

[£]Department of Computer &
Electronic Engineering
University of Nebraska-
Lincoln, Omaha, NE 68182

[†]Department of Electrical &
Computer Engineering
Duke University,
Durham, NC 27708

[‡]Department of Electrical &
Computer Engineering
Carnegie Mellon University,
Pittsburgh, PA 15213

Abstract

We address the problem of generating compact dictionaries for the diagnosis of unmodeled faults in scan-BIST. We present dictionary organization schemes that provide two orders of magnitude reduction in dictionary size with no significant loss in resolution, and facilitate the diagnosis of unmodeled faults. Experimental results for the ISCAS-89 benchmark circuits show that various types of unmodeled faults can be efficiently located using compact dictionaries generated for single stuck-at faults.

1 Introduction

Diagnosis based on fault dictionaries alleviates the need for repeated fault simulation [5, 11]. However, as designs grow in complexity, dictionary-based diagnosis suffers from prohibitively large dictionary sizes. This problem is even more acute for scan-BIST due to the large number of pseudorandom test vectors. Several techniques have therefore been proposed for reducing dictionary size [1-4]. An important consideration in dictionary compaction is diagnostic resolution, which is determined by the number of modeled faults that correspond to the same entry in a compact fault dictionary. An effective dictionary compaction scheme should yield a small dictionary without significant loss in diagnostic resolution [2, 3, 4, 5, 6, 11].

Fault dictionaries are constructed using specific fault models, where each fault corresponds to an entry in the dictionary. However, many test responses of a faulty circuit do not match any dictionary entry, indicating the presence of an unmodeled fault that is not included in the dictionary. Since it is infeasible to incorporate all fault models in a dictionary of limited size, a practical solution must be able to indirectly identify faults that are not included in the dictionary. This is usually done using scoring algorithms [5, 7, 10, 14]. Therefore, a compact dictionary must contain useful information for the application of scoring algorithms to target unmodeled faults.

A recent dictionary-based approach attempts to make diagnosis more efficient by classifying faults based on the detection probabilities of the faults, and using a separate dictionary for each fault type [9, 15]. First, a compacted full-response dictionary $\mathcal{D}_1$ is created for a small number of vectors. This dictionary is used to diagnose faults with high detection probability. For a circuit with O outputs (including scan cells) and N vectors, a typical full-response dictionary contains an ON-bit entry for every fault. This makes a full-response dictionary prohibitively large. This is overcome by simulating an LFSR to compact each ON-bit entry in $\mathcal{D}_1$ to an S-bit signature during dictionary creation. For example, for $O = 100$, $N = 1000$, and $S = 16$, each dictionary entry is reduced from 10^5 bits to 16 bits, which represents several orders of magnitude compaction. This signature is stored in the dictionary, and it is compared with the S-bit signature generated by an on-chip S-bit LFSR during test application.

Next, intervals of test vectors as defined in Section 2 are used to generate a highly-compacted dictionary $\mathcal{D}_2$ over all BIST vectors to target the random-testable faults that have relatively low detection probability. Fault simulation is used to create this dictionary. Each bit of a dictionary entry in $\mathcal{D}_2$ denotes a pass/fail binary outcome corresponding to an interval of test vectors for a modeled fault. An interval-based dictionary is used instead of an LFSR-based dictionary here. The latter is generated using all the output values hence it is practical only for a small number of test vectors. During test application, the MISR signatures for the intervals are taken off-chip, and compared with the known fault-free interval signatures to obtain a set of interval pass/fail outcomes, which is then compared with the dictionary entries to determine the candidate faults. A third dictionary $\mathcal{D}_3$ (organized in the same manner as $\mathcal{D}_1$) is used to diagnose the remaining random-resistant faults that are targeted by "clean-up" ATPG vectors.

In order to handle unmodeled faults, a fault dictionary must contain adequate information to facilitate scoring algorithms. Scoring can be easily applied to $\mathcal{D}_2$ because this dictionary contains pass/fail status of vectors. However, for $\mathcal{D}_1$ and $\mathcal{D}_3$ the information of failing vectors and failing outputs in the responses is 'scrambled' in the LFSR, hence the LFSR signature itself cannot be used for scoring.

In this paper, we first present two partitioning schemes that include the partitioning information in $\mathcal{D}_1$ and $\mathcal{D}_3$ together with the LFSR signature such that a scoring procedure can be directly used. We then present two alternative schemes that use multiple LFSRs of smaller sizes such that the pass/fail status of the LFSR signatures can be directly used for scoring. The objective here is not to devise new scoring algorithms; rather, the focus here is on the design of compact dictionaries that facilitate scoring. Experimental results for the ISCAS-89 benchmarks and for various types of unmodeled faults show that these three dictionaries are very effective for diagnosing unmodeled faults using scoring.

[*] Supported in part by the National Science Foundation under grants CCR-9875324 and CCR-0204077.

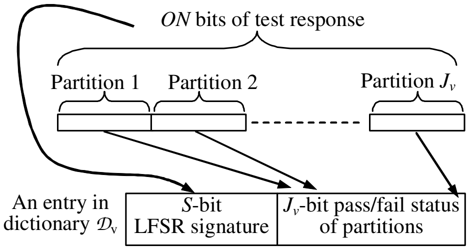
2 Diagnosis using compact dictionaries

In an interval-based pass/fail dictionary, the set of BIST patterns is divided into intervals of test vectors where each interval corresponds to a subset of consecutive vectors [12]. As described in detail in [12], two on-chip MISRs are used to collect signatures. At the end of each interval the signature containing test response of the test vectors in this interval is downloaded from the MISR and sent off-chip for failure analysis. The MISR is then reset before the start of the next interval such that a faulty interval does not affect the pass/fail status of the subsequent intervals. Two MISRs are used in an interleaved fashion such that the test application is not interrupted when one MISR stops to download its signature.

The test sequence is split into intervals of length L (L vectors per interval) without overlap and let I_i represents the i^{th} interval. Let $\mathcal{F} = \{F_1, F_2, \ldots, F_M\}$ be the set of modeled faults. These faults correspond to rows in the compact dictionary, and the intervals make up the columns. If a fault F_i is detected by a vector in interval I_j, i.e., interval I_j fails due to fault F_i, the corresponding (i, j) entry in the fault dictionary is set to 1, otherwise it is set to 0. A given set of failing and non-failing intervals corresponds to a particular bit pattern that can often be mapped to a set of candidate faults. Since the number of intervals is set to be one to two orders of magnitude smaller than the number of test vectors, the interval-based pass-fail dictionary $\mathcal{D}_2$ is significantly smaller than a maximal-resolution pass/fail dictionary. As interval length increases, the dictionary size decreases with no significant loss in resolution up to a certain threshold for stuck-at faults [9].

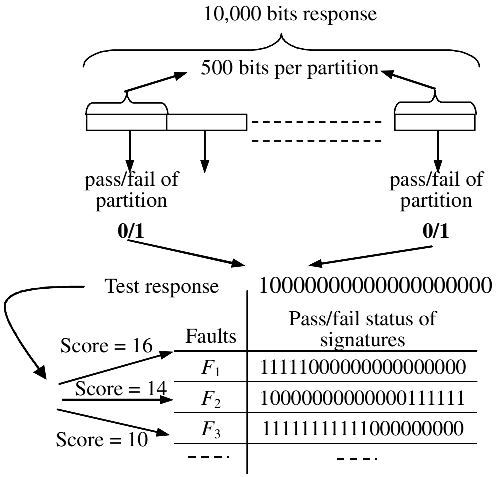
The interval-based pass-fail dictionary $\mathcal{D}_2$ does not consider the effect of the detection probability of a fault on diagnostic resolution. If an entry is made in an interval-based dictionary for an easy-to-detect fault, it is likely that all intervals will fail due to the fault, and the dictionary entry will consist of all 1s. As a result, a large number of easy-to-detect faults will map to the same dictionary entry of all 1s. Hence the interval-based dictionary is only one component of a complete diagnosis procedure. It is useful as part of an overall procedure because it is effective for the relatively hard-to-detect faults that are detected by a small number of pseudorandom vectors.

After constructing an interval-based dictionary using an appropriate interval length, usually determined by the diagnostic threshold, two smaller full-response dictionaries are created. As a pre-processing step, the set of modeled faults is partitioned into three categories: easy-to-detect, hard-to-detect, and undetectable by the pseudorandom patterns used for BIST. The interval-based dictionary is created for the hard-to-detect faults such that the resolution is not adversely affected by the easy-to-detect faults. An LFSR-based compacted full-response dictionary $\mathcal{D}_1$, as described in Section 1, is created for vectors in the first interval for the easy-to-detect faults. We have seen that the vectors in the first interval are usually adequate for these faults. Faults in the third category are not detected by the BIST vectors. Additional "clean-up" ATPG vectors are typically used for these faults, and an LFSR-based compacted dictionary $\mathcal{D}_3$, which has the same organization as $\mathcal{D}_1$, is used for this step.



ON bits of test response
Partition 1
Partition 2
Partition J_v
An entry in dictionary $\mathcal{D}_v$
S-bit LFSR signature
J_v-bit pass/fail status of partitions

Figure 1. Composite signature for scoring in an enhanced LFSR-based dictionary using vector-based-partitioning.



Faults	Pass/fail status of signatures
F_1	11111000000000000000
F_2	10000000000000111111
F_3	11111111111000000000
----	----

Figure 2. An example of a scoring algorithm based on the pass/fail status of partitions using vector-based-partitioning.

3 Diagnosis of unmodeled faults

The three compact dictionaries include only modeled faults (single stuck-at faults in [9]). As mentioned in Section 1, the single stuck-at model has often been shown to be inadequate in practice [7]. In order to handle unmodeled faults, scoring algorithms have been proposed in the literature [5, 7, 14] and a number of architectures have been developed to handle unmodeled faults.

Compared to existing dictionary schemes for unmodeled faults, LFSR-based compact dictionaries offer a number of advantages. First, they are usually three orders of magnitude smaller than the uncompacted dictionaries, with almost no impact on diagnostic resolution [9]. Second, since they are directly created from full-response dictionaries, the number of faults in an equivalence class is much smaller than in a pass/fail dictionary, thus provides a better resolution. (A set of faults having the same dictionary entry constitute an equivalence class.) Third, the dictionary entry has the same format as the signature obtained from an on-chip LFSR, hence a direct comparison can be made without any further additional processing. In case of a fault-free circuit, this can speed up the comparison process.

An LFSR-based dictionary (either $\mathcal{D}_1$ or $\mathcal{D}_3$) can be made useful for scoring through two partitioning schemes as described below.

3.1 Vector-based partitioning

In *vector-based-partitioning*, we perform a partition on the test response information over all the outputs. For each *ON*-bit entry in a full response dictionary, where *O* is the number of outputs (including scan cells) and *N* is the number of test vectors, we divide the entry into J_v partitions of equal length. For each partition, we compare its content to the fault-free value and record only the pass/fail status. We also record the original *S*-bit LFSR value. The resulting entry in such an LFSR-based dictionary is therefore a composite signature, which contains J_v bits of pass/fail status of partitions and an *S*-bit LFSR value. As a result, we need to not only obtain the *S*-bit signature in the output MISR at the end of the BIST operation, but also transfer the response of each partition from the chip after the partition is applied. This can be implemented by adding another shift register at the output of the MISR as proposed in [8]. The pass/fail status of each partition can then be computed by signature analysis without resetting the MISR after each partition [8]. The resulting size of a dictionary entry is therefore increased to $S + J_v$ bits, as shown in Figure 1. However, this new dictionary contains pass/fail status information for each partition of ON/J_v bits, which makes it suitable for scoring algorithms.

We note that the addition of the partitioning information in dictionary entry does not reduce the importance of the *S*-bit LFSR signature. The latter can be used to diagnose modeled faults as in the original LFSR-based dictionary without partitioning. Without the *S*-bit signature, the J_v-bit partitioning information yields low diagnostic resolution for easy-to-detect modeled faults. We refer to such an enhanced LFSR-based dictionary as $\mathcal{D}_v$.

We illustrate the above procedure in Figure 2. We also show an example of scoring based on the notion of Hamming distance for the 20 status bits between dictionary entries and the test response. The scores shown in Figure 2 are obtained by subtracting the Hamming distance from the total number of pass/fail status bits (J_v = 20) in the test response. It can be seen that there exists a trade-off between the effectiveness of compaction and the accuracy of scoring. A larger value of J_v, i.e. partitioning a dictionary entry into more sections, leads to higher accuracy for scoring, but the compaction ratio is reduced proportionately.

3.2 Output-based partitioning

An alternative *output-based-partitioning* can be performed on the circuit outputs over all the test vectors. In this method, we split each *O*-bit output responses into J_o sub-partitions. The output responses from the i^{th} sub-partition of all the test vectors form the i^{th} partition. We compare the content of each partition to the fault-free value and record only the pass/fail status. Therefore, the resulting entry in such an LFSR-based dictionary is still a composite signature containing J_o bits of pass/fail status of partitions and an *S*-bit LFSR value. During the BIST procedure, a MISR is still used to collect all the responses to form the *S*-bit LFSR signature, but we need a separate LFSR for collecting response from each partition. This scheme is especially suitable for the widely-used STUMPS architecture, where response from each scan chain forms a partition and can be associated with one LFSR signature. For the above example, if *O* = 1000, *N* = 10, J_o = 20 and *S* = 16, each dictionary entry is still reduced from 10^4 bits to 36 bits, which still represents two orders of magnitude compaction, see Figure 3. The pass/fail status of the 20 partitions can be viewed as a 20-bit entry that can be used by a scoring algorithm. We refer to such an LFSR-based dictionary as $\mathcal{D}_o$.

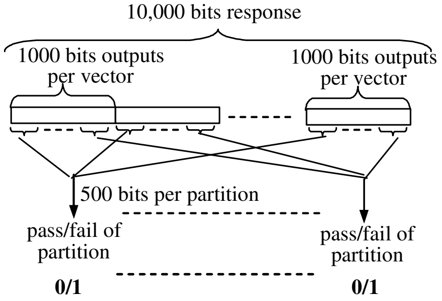


Figure 3. An illustration of output-based-partitioning.

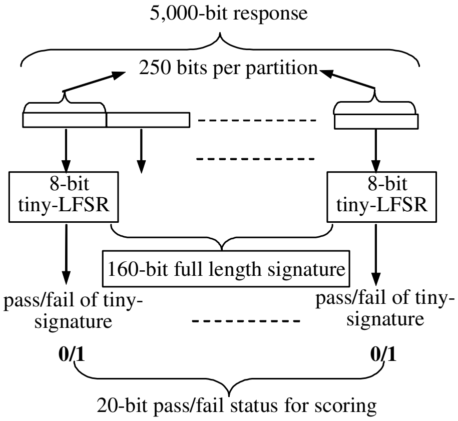

5,000-bit response
250 bits per partition
8-bit tiny-LFSR
8-bit tiny-LFSR
160-bit full length signature
pass/fail of tiny-signature
pass/fail of tiny-signature
0/1
0/1
20-bit pass/fail status for scoring

Figure 4. Alternative partitioning scheme for vector-based-partitioning using multiple tiny-LFSRs.

3.3 Enhanced schemes using tiny-LFSRs

While the above two schemes facilitate the diagnosis of unmodeled faults using LFSR-based dictionaries, they suffer from implementation problems. In order to apply the test application in one session, the vector-based-partitioning requires sophisticated on-tester control logic [8]. In addition, when a faulty outcome is observed, the process needs to be rolled back and a complex signature calculation is required to determine pass/fail of each partition. If multiple test sessions are allowed, where the signature of each partition can be downloaded after a test session, additional time for downloading is inevitable. Similar difficulties exist for the output-based-partitioning scheme.

We next propose two enhancements to the above partitioning schemes. These alternatives rely on the same partitioning strategy as the above two schemes, hence they produce the same scoring outcome. However, instead of generating all partition signatures using a single *S*-bit LFSR, they use a set of smaller LFSRs, or *tiny-LFSR*s, to generate a partition signature, refered to as a *tiny-signature* for each tiny-LFSR. Therefore, each tiny-LFSR is associated fixed partition. These tiny-signatures are precomputed using simulation, and then compared off-chip with on-chip generated tiny-signatures during test application. All the tiny signatures can be concatenated to form a full-length signature for diagnosis, while the pass/fail status of each tiny signature can be used for scoring. Since the length of test response compacted in a tiny-LFSR is much less than that of the entire test response, the length of tiny-LFSR can be shorter than the long LFSR used above. Moreover, because each tiny-LFSR is connected to only one partition, the signatures can be examined at the end of the entire test session without the need for intermediate downloads and complex control. This can potentially lead to less test application time.

The first enhancement is similar to the vector-based-partitioning scheme in that we perform a partition on the test response information over all the outputs. For each *ON*-bit entry in a full response dictionary, we divide the entry into J_v partitions of equal length. We then apply LFSR compaction using a Q_v-bit tiny-LFSR, where $Q_v << S$, to each ON/J_v-bit partition. Therefore, the resulting entry in such an LFSR-based dictionary contains J_v tiny signatures concatenated together, and the length of each dictionary entry is now J_vQ_v /S times as large as the original LFSR-based dictionary. Since each signature is derived for only ON/J_v bits of test response data instead of *ON*, the reduction in LFSR size does not significantly increase the probability of aliasing. This new dictionary contains pass/fail status for each partition of ON/J_v bits, hence it is now suitable for scoring algorithms. An example is illustrated in Figure 4.

The second alternative is similar to the output-based-partitioning that can be performed on the output responses over all the test vectors. We split the *O*-bit outputs into J_o partitions, each associated with a Q_o-bit tiny-LFSR, where $Q_v << S$. For each test vector, the output response from the i^{th} partition is scanned into the i^{th} tiny-LFSR, and a signature is produced over all the test vectors. The J_o tiny signatures can then be concatenated to form a dictionary entry, As a result, the length of each dictionary entry is now J_oQ_o /S times as large as the original LFSR-based dictionary. Note that this scheme is also suitable for implementation in a STUMPS architecture, where each scan chain can be associated with one tiny-LFSR.

With the inclusion of these new architectures, the LFSR-based dictionaries can now be used to target unmodeled faults, while their inherent advantages on size, resolution and easiness to use are not compromised.

4 Experimental results

In this section, we present simulation results for the full-scan ISCAS-89 benchmark circuits. We first examine the resolution of an interval-based dictionary $\mathcal{D}_2$ for *all* single stuck-at faults using an average measure *DE*, the *diagnostic expectation*, which is defined as the average size of an equivalence class. It can be calculated as follows:

$$DE = \frac{\sum_{\forall \text{faults } f} \text{Size of } f\text{'s equivalence class}}{\text{number of faults}}$$

A smaller value of *DE* indicates higher diagnostic resolution.

For each benchmark circuit, we considered a total of 10,000 pseudorandom vectors to construct a pass/fail dictionary for all the "hard-to-detect" faults that are not detected by the first interval. We limit ourselves to 10,000 vectors because we also generate comparative data for full-response dictionaries. Even with only 10,000 patterns, the full-response dictionaries are gigantic, and it takes excessive CPU time. Moreover, we have observed in experiments that as we increase the number of vectors to 100,000 and increase the interval length from 100 to 1,000 simultaneously, i.e., keep dictionary size the same, there is no loss of diagnostic resolution. Therefore, $\mathcal{D}_2$ can be efficiently generated for a much larger number of patterns. The generation of $\mathcal{D}_1$ and $\mathcal{D}_3$ takes relatively less time since they are generated using only a small number of patterns.

We examine the effectiveness of using a simulated LFSR and the partitioning schemes to compact the first-step full-response dictionary $\mathcal{D}_1$. In Table 1, we list the dictionary sizes and the *DE* values before and after compaction based on a 16-bit LFSR and partitioning schemes. For the sake of simplicity, we only list the results using vector-based partitioning with the partition length from 50 to 200; the results using output-based partitioning are similar. It can be seen that by using compact dictionaries, we obtain two orders of magnitude reduction in size, yet there is almost no loss in diagnostic resolution. Compared to [9], where no partitioning information is included, the dictionaries are larger here but the diagnostic resolution is improved. The enhanced partitioning schemes using tiny-LFSR leads to much larger dictionaries, hence they are more suitable for circuits that have relatively short test responses.

The full-response dictionaries in this paper have been asumed to include both faulty and fault-free outputs. However, a more efficient full response dictionary can be based only on the failing outputs [3]. Since the LFSR-based

Circuits	Size before comp-action (Mbits)	*DE* before comp-action	Original scheme without tiny-LFSR		Enhanced scheme using tiny-LFSR	
			Size after comp-action (Mbits)	*DE* after comp-action	Size after comp-action (Mbits)	*DE* after comp-action
s9234	6.7	3.02	0.18	3.02	0.8	3.02
s13207	43.1	2.37	0.52	2.37	1.3	2.79
s15850	43.8	2.74	0.55	2.74	3.55	2.74
s38417	332	2.22	1.99	2.22	13.1	2.23
s38584	386	1.98	2.32	1.98	15.5	1.98
s35932	396	7.1	2.29	7.1	15	7.2

Table 1. Dictionary size and diagnostic expectation of $\mathcal{D}_1$ for larger ISCAS-89 circuits.

Circuits	Pass/fail dictionary size before compaction (Mbits)	Overall size after compaction (Mbits)	*DE* after compac-tion	Size of diction-ary in [5] (Mbits)	*DE* for diction-ary in [5]
s9234	58.1	0.54	2.86	2.30	1.87
s13207	89.2	0.94	3.62	9.84	1.64
s15850	106.8	1.05	2.65	8.83	1.7
s38417	274.3	3.21	2.24	55.9	1.5
s38584	340.0	3.7	2.04	63.5	1.22
s35932	351.1	4.58	6.3	60.9	3.9

Table 2. Overall dictionary size and *DE* values for the larger ISCAS-89 circuits and the corresponding results obtained in [5].

compaction is independent of the underlying dictionary organization scheme, similar compaction can also be carried out for full-response dictionaries based on the information about failing outputs.

In Table 2, we present results on the total dictionary size and the *DE* values when we take into account the dictionaries for all the three steps. We also compare our results to a recent compact dictionary approach based on output compaction [5]. We note here that our method is based primarily on BIST vectors, while the work in [5] was aimed at deterministic ATPG vectors where the number of test vectors are significantly smaller than the BIST case. Precisely, dictionaries in both of these two methods should be viewed as low resolution dictionaries. Column 2 in Table 2 shows the dictionary size of a pass/fail dictionary based on 10,000 BIST vectors. Clearly, for large circuits, even a pass/fail dictionary is impractical. For the interval-based dictionary used in the second step, we choose the interval length *L* to be 100. We add the dictionary sizes for the three steps $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and list them in Column 3. In Column 4, we list the *DE* values of the proposed method. It is calculated over all the three dictionaries and all the detectable faults. In Columns 5 and 6, we list dictionary sizes and *DE* values obtained from the compaction method described in [5] using ATPG vectors. It can be seen that even though our method uses a much larger amount of BIST vectors than the ATPG vectors in [5], the overall dictionary size is significantly smaller, and diagnostic resolution is in most cases comparable to that in [5].

We next present simulation results for the diagnosis of unmodeled faults using scoring on the LFSR-based dictionary $\mathcal{D}_1$, which includes the partitioning information described in Section 3. We use the partitioning schemes proposed in Section 3.1 and 3.2. We do not present results using enhanced schemes based on tiny-LFSRs since they lead to the same scoring outcome.

We choose six types of unmodeled faults from [13] (without feedback): AND-bridging faults, OR-bridging faults, Dominant-A bridging faults, Dominant-B bridging faults, Stuck-open faults and Transition faults, and present the corresponding results in Tables 3. For each fault type, we randomly select 40 unmodeled faults one at a time, and use the scoring procedure presented in Section 3 to determine a set of candidate fault locations.

We first verify that the faulty response in each case does not match any of the entries in the dictionary based on single stuck-at faults, implying the presence of an unmodeled fault. We then use the pass/fail status of partitions to compute the score for each single stuck-at fault in the dictionary, as shown in Figure 3, and the faults that have the highest score are included in the suspect fault candidate set. The node associated with each fault candidate is deemed as one possible endpoint of the fault. For example, in the case of an bridging fault, if one or more endpoints of the bridge are included in the candidate set, we conclude that the bridging fault is correctly identified.

Note that smilar localization of fault sites can be done using commercial fault simulation and test generation tools such as TetraMAX. However, they rely on uncompacted test responses; here we are using only BIST signatures for the diagnosis.

Circuits	Output-based partitioning		Vector-based partitioning		Full-response dictionary	
	Average size of candidate set *S*	Percent-age of faults correctly identified	Average size of candidate set *S*	Percent-age of faults correctly identified	Average size of candidate set *S*	Percent-age of faults correctly identified
(a) AND-bridging faults						
s13207	89.6/9.6	71.0	19.6/4.0	35.9	7.5/1.2	56.4
s38417	131/42.8	50.0	42.2/37.4	24.3	20.4/1.7	56.4
s38584	77.3/11.8	36.4	77.0/27.7	64.7	10.8/5.4	86.8
(b) OR-bridging faults						
s13207	80.1/38.9	74.4	13.7/3.7	33.3	7.6/2.5	56.4
s38417	124/45.6	71.4	44.9/38.9	29.7	20.1/5.3	66.7
s38584	40.5/5.6	35.7	53.0/12.2	47.1	17.6/6.6	73.7
(c) Dominant-A bridging faults						
s13207	68.2/32.8	66.7	13.6/3.1	23.1	8.3/3.0	61.5
s38417	174/59.4	68.8	40.5/38.9	18.4	21.9/2.8	64.1
s38584	31.3/9.7	70.0	58.9/14.8	48.6	18.2/7.4	79.5
(d) Dominant-B bridging faults						
s13207	62.9/31.3	71.8	20.5/5.7	33.3	6.1/2.3	66.7
s38417	132/39.8	75.0	42.0/39.3	20.5	14.3/3.2	56.4
s38584	67.7/23.4	71.4	53.8/10.7	29.4	13.5/6.8	84.2
(e) stuck-open faults						
s13207	71.8/24.1	69.2	4.9/1.2	53.8	5.1/1.2	53.8
s38417	236/94.6	89.7	21.2/12.4	48.7	5.8/1.4	53.8
s38584	68.1/6.8	66.7	70.4/11.8	68.8	37.7/4.7	77.1
s35932	106/22.2	100	4.1/1.2	100	3.7/1.0	100
(f) transition faults						
s13207	70.1/25.1	64.1	5.1/1.1	51.3	5.0/1.2	53.8
s38417	236/94.5	79.5	20.8/12.3	46.2	5.4/1.4	51.3
s38584	68.4/7.1	64.7	66.0/12.1	68.8	35.6/4.5	77.1
s35932	106/22.4	100	4.2/1.2	100	3.8/1.0	100

Table 3. Scoring results using LFSR-compacted dictionaries for unmodeled faults.

Types of faults	Interval-based pass/fail dictionary $\mathcal{D}_2$		Maximum-resolution pass/fail dictionary	
	Average size of candidate set *S*	Percentage of faults correctly identified	Average size of candidate set *S*	Percentage of faults correctly identified
AND-bridging	84.5/10.8	32.0	65.4/11.1	38.2
OR-bridging	52.8/16.0	60.0	28.8/9.9	64.5
Dominant-A	47.0/16.8	58.3	57.4/11.1	50.0
Dominant-B	60.0/2.3	56.5	43.6/2.4	53.6
Stuck-open	58.8/5.6	36.4	54.0/4.7	30.4
Transition	41.1/11.5	59.1	43.3/8.4	33.3

Table 4. Scoring results using interval-based dictionaries $\mathcal{D}_2$ for unmodeled faults for s38584.

We present the average size of the candidate set *S* over all the faults and the percentage of fault locations that are correctly identified. For each experiment, the sizes of candidate sets are presented using two measures in the form of A/B. The first represents the average size of the equivalent class, assuming this is a low resolution dictionary and additional fine-grained methods are to be applied on top of these results to finally locate the true failure. The second assumes that after each detection, the candidate fault is to be examined to determine if it is indeed the failure, until the first true failure is affirmed. A smaller value indicates that the fault can be identified after a smaller number of attemps, hence located faster.

We did not consider any layout information in our random selection of bridging faults; we expect the results in Table 3 to be better if a more sophisticated scoring algorithm is employed, and layout information is used to extract realistic bridging faults.

The results are presented for both the output-based and the vector-based partitioning schemes, which have the same dictionary size. It can be seen from the results that a majority of the unmodeled faults can be identified, i.e., the corresponding locations are included in the candidate set, by the output-based scheme. However, this method also yields large candidate sets by the first measure (equivalent class). In contrast, the vector-based scheme leads to much smaller candidate sets, but it is less effective in identifying unmodeled faults. The choice of an appropriate scheme can therefore be made based on diagnosis requirement. The candidate sets are quite large in some cases because here the nodes associated with an unmodeled fault are selected randomly over the entire circuit without considering the layout, which often makes the faults hard to identify. We expect much smaller candidate sets and higher diagnosability percentage if layout information is available and improved scoring algorithms are used. We also note that as we increase the amount of compaction, i.e., by using larger partitions, both the size of candidate set and the diagnosability percentage increase.

For the sake of comparison, we also include the average size of the candidate sets and the percentage of diagnosed unmodeled faults when the same scoring procedure is applied to a full-response dictionary without compaction. It can be seen that the average size of S by the first measure (equivalent class) is smaller for the full-response dictionary. However, in many cases the average size of S by the second measure (first detect) are comparable with the full-response dictionaries for this simple scoring algorithm and compact dictionaries, even though they are two orders of magnitude smaller. This clearly indicates that the faults can be located fast using the LFSR-based compaction dictionaries. It also justifies the advantages of using compact dictionaries. Note that in many cases the percentage of successful diagnosis given by the output-based partitioning is even better than the full-response dictionary. Intuitively this is due to the larger candidate set of the compact dictionaries, which should be mainly attributed to the simple scoring algorithm. Also under such a scheme, a fault receiving the highest score in a full-response dictionary may not have the top ranking in a compact dictionary, and a better candidate could be chosen.

Finally in Table 4, we present the scoring results for unmodeled faults using the interval-based pass/fail dictionary $\mathcal{D}_2$ based on 10,000 random vectors and interval length $L = 100$. Experiments are carried out for s38584 and six types of unmodeled faults. The faults selected are relatively hard to detect and 40 faults of each type are considered. As in Table 5, we present the average size of the candidate set S using two measures over all the faults, and the percentage of fault locations that are correctly identified. We also present results for the maximum-resolution dictionary, i.e., interval length $L = 1$. It can be seen that the compact interval-based dictionaries are nearly as effective for diagnosing unmodeled faults as the maximum-resolution dictionaries, even though they are two orders of magnitude smaller. In almost all the cases the average size of S by the second measure (first detect) and the percentage of fault sites included in S are comparable with those of the maximum-resolution dictionaries for this simple scoring algorithm and the compact dictionaries. Once again, in some cases, the compact dictionaries can achieve smaller candidate size and higher detection rate, which can also be attributed to the simple scoring scheme as discussed above, a better candidate could be chosen in a compact dictionary.

5 Conclusions

We have shown how compact dictionaries can be designed for the diagnosis of unmodeled faults in scan-BIST. A combination of three compact dictionaries can be used to obtain two orders of magnitude reduction in dictionary size without significantly compromising diagnostic resolution for single stuck-at faults. We have presented LFSR-based dictionary organization schemes that facilitate the use of scoring algorithms for the diagnosis of unmodeled faults. Experimental results for the ISCAS-89 benchmark circuits show that various types of unmodeled faults can be efficiently located using scoring algorithms and compact dictionaries generated for single stuck-at faults.

References

[1] R. C. Aitken and V. K. Agarwal. “A Diagnosis Method Using Pseudo-random Vectors without Intermediate Signatures”, *Proc. ICCAD*, pp. 574-580, 1989.

[2] V. Boppana, I. Hartanto and W. K. Fuchs, “Full Fault Dictionary Storage Based on Labeled Tree Encoding”, *Proc. VTS*, pp. 174-179, 1996.

[3] S. Chakravarty and V. Gopal, “Techniques to Encode and Compress Fault Dictionaries”, *Proc.VTS*, pp. 195-200, 1999.

[4] P. G. Ryan and W. K. Fuchs, “Dynamic Fault Dictionaries and Two-Stage Fault Isolation*”, IEEE Trans. VLSI*, vol. 6, pp. 176-180, Mar 1998.

[5] D. Lavo and T. Larrabee, “Making Cause-Effect Effective: Low-Resolution Fault Dictionaries”, *Proc. ITC*, pp. 278-286, 2001.

[6] B. Arslan and A. Orailoglu, “Fault dictionary size reduction through test response superposition”, *Proc.ICCD*, pp. 480-485, 2002.

[7] P. C. Maxwell et al., “IDDQ and AC Scan: The War Against Unmodeled Defects”, *Proc. ITC*, pp. 250-258, 1996.

[8] J. Savir, “Salvaging Test Windows in BIST Diagnostics”, *Proc. VLSI Test Symposium*, pp. 416-425, 1997.

[9] C. Liu and K. Chakrabarty, “Compact Dictionaries for Fault Diagnosis in BIST”, *Proc. ISQED*, pp. 105-110, 2003.

[10] S. D. Millman, E. J. McCluskey and J. M. Acken, “Diagnosing CMOS Bridging Faults with Stuck-At Fault Dictionaries”, *Proc. ITC*, pp. 963-968, Oct 1991.

[11] I. Pomeranz, “On Pass/Fail Dictionaries for Scan Circuits”, *Proc. ATS*, pp 51-56, 2001.

[12] C. Liu and K. Chakrabarty, “Failing Vector Identification Based on Overlapping Intervals of Test Vectors in a Scan-BIST Environment”, *IEEE Trans. CAD*, vol. 22, pp. 593-604, 2003.

[13] K. N. Dwarakanath and R. D. Blanton, “Universal Fault Simulation Using Fault Tuples”, *Proc. DAC*, pp. 786-789, 2000.

[14] D. B. Lavo, I. Hartanto and T. Larrabee, “Multiplets, models, and the search for meaning: improving per-test fault diagnosis”, *Proc. ITC*, pp. 250-259, 2002.

[15] C. Liu and K. Chakrabarty, "Compact Dictionaries for Fault Diagnosis in Scan-BIST", accepted for publication in *IEEE Trans. Comp.*, 2003.