An Efficient Method for Improving the Quality of Per-Test Fault Diagnosis

Chunsheng Liu
Department of Computer & Electronic Engineering
University of Nebraska-Lincoln, Omaha, NE 68182

Abstract
Per-test fault diagnosis methodology has been shown to be an effective one for the identification of complex defects. In this paper, we improve a recent per-test technique by applying additional diagnosis on the outputs of the circuit. The new method brings in more evidence to support the true failures, hence improves the diagnostic quality. We show that this method can very well address several problems in previous work.

1. Introduction
As feature sizes shrink and design complexity grows rapidly, fault diagnosis has become extremely difficult. A major concern in failure analysis is that the suspected candidates given by a diagnosis algorithm can often be incomplete and even misleading [2-4].

A recent methodology is based on the concept of “one test at a time”, or per-test diagnosis [5-8]. In these approaches, test patterns are viewed as independent and diagnosis is carried out on each test pattern and produces a candidate fault set for each of them. Recently, per-test diagnosis has been substantially improved by introducing the notion of “multiplet”. A multiplet is a set of single candidate faults, all the single faults in a multiplet can be used together to completely explain the faulty responses produced by all the failing test patterns [7, 8].

Per-test diagnosis using multiplet is best presented in [7] and [8]. In [7], a Single Location At-a-Time (SLAT) approach is presented by assuming that there exist some test patterns that can cause some single site faults to produce observable failing responses. Each of these test patterns (SLAT patterns) is then associated with a number of such single site faults (usually single stuck-at faults), and each fault can be used to explain the failing responses of that test pattern. A set of single faults, which can together explain the faulty responses of all the SLAT patterns, are found and this set is deemed as a multiplet. Later in [8], an improved Single Test At-a-Time (iSTAT) approach is introduced. iSTAT still produces the multiplets based on the SLAT strategy. But it applies a scoring algorithm to rank the multiplets and only the ones with highest score are selected. It is shown that scoring can significantly reduce the number of candidate multiplets, hence improve the diagnostic resolution.

In this paper, we propose a new approach that can further improve the diagnostic quality of SLAT and iSTAT. It uses the same strategy as iSTAT to produce scored multiplets. However, in addition to using the response of each failing test pattern, it also exploits the information associated with each failing output pin and generates a new list of scored multiplets. This can be referred to as Single Output At-a-Time (SOAT). The multiplets from iSTAT and SOAT are then combined and a scored fault list is produced through a new scoring algorithm. This approach can achieve a diagnostic quality superior to both SLAT and iSTAT on diagnostic accuracy and failure coverage. It also helps to reduce the occurrence of aliasing. For the sake of simplicity, it will be referred to as STSOAT (Single Test and Single Output At-a-Time) in the subsequent discussion.

The paper is organized as follows. In Section 2, we review the approaches of SLAT and iSTAT, and point out some weakness that can be improved. In Section 3, we introduce the proposed STSOAT approach and the corresponding scoring strategy. Finally in Section 4, we present experimental results.

<table>
<thead>
<tr>
<th>Test number</th>
<th>Matching Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>A, B</td>
</tr>
<tr>
<td>3</td>
<td>C, D</td>
</tr>
</tbody>
</table>

Table 1. A simple example of exact matching tests.

2. Per-Test Diagnosis Algorithms
The concept of per-test fault diagnosis, where the test patterns (or tests hereinafter) are analyzed one at a time (or STAT), has been adopted in several previous work, such as PIOROT [6], SLAT [7] and iSTAT [8]. These algorithms can be best illustrated using the example shown in Table 1. In this table, single stuck-at fault A is an exact match for test 1, which means the predicted failing outputs with A injected in the circuit match exactly the observed failing outputs of test 1. Similarly, both faults A and B match test 2 and both faults C and D match test 3. The algorithm in [5] will simply report a set of faults (A, B, C, D), any of which can cause a test to fail. The PIOROT algorithm [6] will produce the same list, but every fault will have a score representing the number of tests that are explained by the fault. For instance, A will receive a higher score than others since it can explain two tests.

The SLAT algorithm [7], however, will create several multiplets as follows: (A, C), (A, D), (A, B, C) and (A, B, D). Note that each multiplet can explain the outputs of all the three matching tests. Redundancies are removed such...
that only the minimum-sized multiplets are reported. For example, (A, B, C, D) will be removed since it is sufficient for only one of C and D to explain test 3.

The iSTAT algorithm [8] produces similar multiplets as SLAT, but it yields a more accurate result by applying a scoring algorithm. The algorithm is based on the well-known Dempster-Shafer method with a belief function [1]. The procedure can be illustrated using the example in Table 1. For test 1, the belief assigned to A is \( p_1(A) = 0.99 \) and the reserved probability for aliasing is \( p_1(\Phi) = 0.01 \). For test 2, however, a belief of 0.495 is assigned to both A and B since they appear to be equally evident. Faults C and D are weighted similarly. The scores for multiplets are calculated progressively by an orthogonal combination of two belief functions. The combined probabilities of multiplets \( (A) \) and \( (A,B) \) are:

\[
p(A) = p_2(A) p_1(A) + p_2(A) p_1(\Phi) + p_2(\Phi) p_1(A) = 0.5049
\]

\[
p(AB) = p_2(B) p_1(A) = 0.49005
\]

Other probabilities can be calculated similarly. The results after applying all three tests are (after rounding):

\[
p(AC) = p(AD) = 0.2499; \quad p(ABC) = p(ABD) = 0.2426.
\]

These probabilities are viewed as the scores of multiplets, higher score represents higher probability of occurrence. Note that there are two top-ranked multiplets \( (A,C) \) and \( (A,D) \). They are given more credibility over \( (A,B,C) \) and \( (A,B,D) \) because A appears to be more possible than B for its appearance in two tests. In this example the iSTAT reports two multiplets instead of four given by SLAT. Therefore, iSTAT reduces the number of multiplets, and better diagnostic resolution is obtained.

Besides the exact matching tests, there are two other possible types of tests: passing and complex. A passing test does not produce any failing output and a complex test produces failing outputs that do not exactly match the predicted response of any single fault. Both passing and complex tests are accounted in iSTAT such that the responses of all the tests are accounted in the final multiplet scores. This makes the evidence reflected by the final scores more accurate and credible.

Although iSTAT has shown a lot of strength on increasing the diagnostic resolution over previous per-test techniques, there still exist some problems that need to be addressed. First, iSTAT relies on user’s confidence on decision making, and it needs more evidence to provide user with stronger confidence on the true failures. For example, the score calculation may lead to accumulative error. For two multiplets with scores of 0.200001 and 0.20000, respectively, we cannot tell with much confidence which should be preferred because the difference might be caused by accumulative error incurred in the calculation. In order to make user more confident on the correct decision, we need to find more evidence to support the true failures.

Second, iSTAT gives evidence on which multiplet is more likely to include a true fault site. However, it doesn’t tell that within a multiplet, which fault is more likely to be the true fault site. Therefore, if the top-ranked multiplet contains a large number of fault candidates, iSTAT becomes less accurate.

Third, while iSTAT reduces the candidate size compared to SLAT, it may lead to a higher probability of aliasing, i.e., the true failing sites are not included in the top-ranked multiplets. This can be demonstrated through experiments in Section 4. It can be seen that in most cases, the true failing sites can be successfully included by SLAT multiplets. However, the percentage of these successful cases of iSTAT is lower, indicating that the scoring algorithm and the choice of top-ranked multiplets can lead to a misleading result, where the true fault sites are not included in the top-ranked multiplets.

Finally, the experiments in [8] are carried out on only 20 faults, which is inadequate to provide results from a statistical point of view.

3. Exploit Both Per-test and Per-output Information: STSOAT

3.1 The SOAT Algorithm

We first examine the correspondence between tests and faulty output pins in iSTAT, which is illustrated in Figure 1 through a “matching matrix”. The first three rows in Figure 1 represents three matching tests with corresponding matching faults as shown in Table 1, assuming the circuit has ten outputs. For each matching fault, its predicted outputs match exactly the response of the test. There are two additional rows, corresponding to a passing test and a complex test, respectively. For each test, we list the corresponding outputs, with faulty and fault-free pins represented by shaded and clear boxes, respectively. The matrix can be referred to as a “test matching matrix” since the index is a list of tests.

Next, we notice that if the positions of tests and outputs in Figure 1 are exchanged, a new matching matrix can be created, see Figure 2. In this new matrix, the index is a list of “matching” outputs, contrast to the matching tests in Table 1. Each output may be associated with a number of matching faults. In the presence of each matching fault, the predicted pass/fail status of all tests on that output matches exactly the observed status. Such an output is referred to as a “matching” output. In case of no matching, an output is either “passing” or “complex”, which is defined similarly as those in Figure 1. This matrix can be referred to as “output matching matrix”. We note that the matching faults in Figure 2 have no correlation with those in Figure 1. Therefore, it provides additional knowledge on fault candidates, which can be used for diagnosis.

It is obvious that the two matching matrices are mutually transposed. However, as can be seen in Figure 2, faults are evaluated again based on the matching, passing and complex outputs, other than tests. This provides additional evidence for scoring, which is non-
redundant since it is independent of the evidence given by test matching matrix. Because this new scoring is based on outputs, we refer to it as a “per-output” diagnosis or “Single Output At-a-Time” (SOAT).

Because the architecture of output matching matrix is identical to the test matching matrix used in iSTAT, we can directly apply the scoring strategy of iSTAT on it, as illustrated in Section 2. This will produce a list of new scored multiplets. The processing is still applied separately on matching, passing and complex outputs.

3.2 The STSOAT Approach

Obviously, SOAT can be used as stand-alone approach. However, since the information needed by SOAT is no more than that of iSTAT, a better choice might be to combine the two and produce a new approach that is superior to both of them. This is possible since by merging iSTAT and SOAT, more evidence is brought into the scoring procedure, which can potentially make the scoring results more accurate. We refer to this approach as Single Test and Single Output At-a-Time (STSOAT). The procedure is outlined as follows.

We first apply both iSTAT and SOAT using test and output matching matrices, respectively, and we keep all the multiplets with scores, i.e., a set of iSTAT multiplets and a set of SOAT multiplets. This is different from iSTAT, where only top-ranked multiplets are selected for failure analysis. We then create a list of distinct fault sites from iSTAT multiplets and another from SOAT multiplets, referred to as “iSTAT fault list” and “SOAT fault list”, respectively. An example is shown in Table 2, where we list the multiplets and corresponding fault list for both iSTAT and SOAT.

![Table 2. Multiplets and corresponding fault lists generated by iSTAT and SOAT.](image)

Next we assign a score to each fault site in the list as follows. For each fault site, we find all the multiplets containing it, and for every multiplet a partial score is obtained. The partial score equals the score of the multiplet divided by the number of faults in the multiplet. The sum of all the partial scores is the score assigned to the fault site. For example, for fault A in Column 3 of Table 2, it appears in two multiplets with scores 0.5 and 0.3, respectively. Since both multiplets contain two fault sites, the score given to A is calculated as $0.5/2+0.3/2=0.4$. In Columns 1 through 4 of Table 3, we show the corresponding scores of the faults in Table 2.

Finally, the two fault lists from iSTAT and SOAT are combined to form a new ordered list, or STSOAT fault list. The scores are first added together and then renormalized to 1.0, as illustrated in Columns 5 through 7 of Table 3. Note that fault site A is given the highest score in STSOAT because it receives highest scores in both iSTAT and SOAT fault lists. In this specific example, the scores given to faults are relatively close.

After the STSOAT fault list is obtained as shown in Table 3, physical failure analysis has to be carried out on this ranked fault list to locate the true defects. This can be done using several strategies. For example, we can set a threshold and only the faults with a score higher than the threshold are examined in the order of their scores, from highest to lowest. (For the example in Table 3, if the threshold is set to be 0.2, then we will first examine fault site A and then B in failure analysis.) Another method is that we choose the faults from top of the ranked list until the sum of their scores exceeds a certain threshold. A more conservative strategy is to examine all the faults starting from the one with the highest score until a true failure site is identified. In all these methods, STSOAT list can help to reduce the number of candidate faults (or equivalently, improve the diagnostic resolution), and it also helps to increase the failure coverage, as shown in the experimental results in Section 4.

![Figure 1. A test matching matrix used in iSTAT.](image)

![Figure 2. An output matching matrix.](image)
4. Experimental Results

We run experiments on six largest ISCAS 89 benchmark circuits. For each circuit, we randomly generate over 300 various types of unmodeled faults (the single stuck-at faults are used as modeled faults), among them are multiple stuck-at faults (2 or 3), different types of bridging faults (OR, AND, XNOR and Dominance), gate replacement faults (NOR to NAND, NAND to NOR, etc.), gate polarity inversion and other complex net faults (multiple bridging, etc.). These fault types can represent the complex behavior of the physical defects. We inject each fault at a time and produce the multiplets using SLAT, iSTAT and SOAT, respectively. We then create the STSOAT fault list using the multiplets of iSTAT and SOAT as illustrated in Section 3. The results are then averaged over all the diagnosis runs.

We first examine if the true failing sites are included in the results given by these methods. We report in Figure 3 the percentages of the cases that at least one true failing site is included by SLAT, highest-ranked iSTAT multiplets and STSOAT fault list, respectively. It can be seen that in most cases, the true failing sites can be successfully detected by SLAT. However, the percentage of these successful cases using iSTAT is much lower, indicating that the scoring algorithm and the choice of highest-ranked multiplets can lead to misleading results. The result given by STSOAT should be at least as good as that of SLAT since it includes all the fault sites. Moreover, in all the cases the STSOAT is even slightly better than SLAT. As discussed in Section 3, this is because STSOAT brings additional information to the scoring of the fault sites.

![Figure 3. The percentages of cases that at least one true failing site is included using iSTAT, SLAT and STSOAT.](image)

Next, we investigate that to what extent the STSOAT can improve the diagnostic resolution. We compare the number of fault sites that need to be checked before a true failure site is identified. If both of them provide the same number, we count the number of faults need to be checked until all true failure sites are identified. We use this strategy to make a comparison between iSTAT and STSOAT possible since the latter does not have multiplets in its results. A smaller number indicates a higher diagnostic resolution. We show the average number of fault sites that are checked (after rounding) and the corresponding improvement of STSOAT over iSTAT in Table 4. It can be seen that STSOAT can provide a much higher diagnostic resolution, the improvement is over 60%. One more observation is that the numbers shown here are quite small, which indicates the true failures can be quickly identified after carrying out failure analysis on only 3 or 4 fault sites. This also makes STSOAT a practical approach.

5. Conclusion

We have proposed a new diagnosis method that can improve the diagnostic quality of per-test diagnosis. The method is based on the use of both tests and outputs information and a fault list scoring algorithm. It does not require additional test information. Experimental results show that the new method can improve the diagnostic quality in both diagnostic resolution and failure coverage compared to other work.

Acknowledgement

The author thanks Dr. Peilin Song of IBM Corp. for his valuable comments on the earlier draft of this paper.

Reference