A Multiple Fault Diagnosis Method Based on a Single Fault Simulation

Yoseop Lim
Department of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
yoseop@soc.yonsei.ac.kr

Joohwan Lee
Department of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
eldsich@soc.yonsei.ac.kr

Eunsei Park
Department of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
espark@soc.yonsei.ac.kr

Sungho Kang
Department of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
shkang@yonsei.ac.kr

Abstract - *With the increasing complexity of VLSI devices, more complex faults have appeared. Most of previous fault diagnosis methods considered a single defect assumption. However, for present technologies and chip sizes, defects have tendency to be clustered. So, we propose a multiple fault diagnosis method using a fault selection table. The proposed method can diagnose multiple defects based on a single fault simulation. In spite of a multiple fault diagnosis, the number of candidate faults is drastically reduced. Experimental results for ISCAS85 and full-scan version of ISCAS89 benchmark circuits prove the efficiency of the proposed method.*

Keywords: Fault Diagnosis, Fault Simulation, Multiple Faults, Stuck-at Fault.

1 Introduction

With the increasing complexity of VLSI devices, the demand for fault diagnosis has also increased. Fault diagnosis is the process that deduces the location of the defect which caused the failures. An accurate fault diagnosis can identify both design and process errors, thus improving yield. Therefore, it is very important to develop an efficient fault diagnosis methodology in order to improve device quality and reduce a production cost.

Most of previous fault diagnosis methods considered a single defect assumption. However, for present technologies and chip sizes, defects have tendency to be clustered. Preferably, multiple defects on a failing chip reflect the reality more correctly.

In recent years, several papers on multiple-fault diagnosis have been published [1-3]. In [1], the authors use single-location-at-a-time (SLAT) patterns to perform the diagnosis for failure responses caused by multiple faults. The SLAT pattern attributes each failure response to a single fault. Using SLAT patterns, the algorithm tries to find all single-fault locations. Each candidate fault can explain some failing patterns. After finding these candidate faults, the algorithm seeks to sort them into the minimum cardinality groups that could explain all the failing patterns. However, in reality, when multiple faults are present in the circuit, not every fault could have an SLAT test.

In [2], the authors first apply a path-tracing algorithm and find sites with possible candidate faults. Then, they sequentially try each candidate site in the circuit, simulating about 6000-10000 random vectors to see how many failing POs can be corrected by each candidate simulation. The candidates are ranked by their correcting potential. The selected good candidates are injected into the circuit, and the iteration proceeds.

In [3], the authors propose an incremental approach based on multiple-fault simulation with several heuristics to speed up the diagnostic process. However, the algorithms proposed in [2-3] have exponential time complexity due to the inherent nature of the multiple-fault-simulation problem. They may be impractical for big industrial designs.

In this paper, we propose a multiple fault diagnosis method based on a single fault simulation. The proposed algorithm can correctly diagnose multiple faults with the minimum candidate faults set.

In Section 2, we propose our diagnosis algorithm. In Section 3, for ISCAS circuits, we show experimental results. Section 4 summarizes our conclusions.

2 The Proposed Diagnosis Algorithm

The proposed diagnosis algorithm does not have an exponential time complexity problem because it uses a single fault simulator. The simulation time of a single fault simulator is linear with respect to the size of a circuit. And for each fault simulation, a single fault simulation is faster than multiple fault simulation. So, it may be practical for big industrial designs. On the other hand, the simulation time is exponentially increased with a multiple fault simulator. They may be infeasible for large circuits. Another advantage using single fault simulation is easy to implementation.

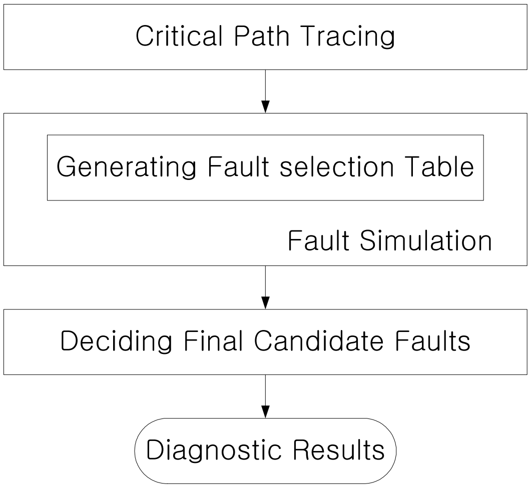
Critical Path Tracing
Generating Fault selection Table
Fault Simulation
Deciding Final Candidate Faults
Diagnostic Results

Figure 1. Initial Fault Selection Table

Figure 1 shows the proposed diagnosis algorithm. The algorithm first applies a path-tracing algorithm to decide initial candidate faults. Then, it simulates initial candidate faults and generates a fault selection table at the same time. After fault simulation, it decides final candidate faults using the fault selection table.

The fault selection table is stored with two criteria to decide final candidate faults. One is the fault score that quantitatively presents a similarity between the simulation result of the candidate fault and the fail-log of a faulty chip. The other is VI flag. VI presents that the simulation result of the candidate fault and the fail-log of a faulty chip are identical. VI (Vector-wise Intersection) is defined in [4]. VI is the most weighted criterion for fault diagnosis in [4] and is an import factor to generate the explain fails table in [1].

The fault scoring algorithm is proposed in [5]. It expands concepts of Vector-wise Intersection, Intersection, Non-prediction and Mis-prediction defined in [4] and quantitatively presents a similarity between the simulation result of the candidate fault and the fail-log of a faulty chip. If both the simulation result and the fail-log have erroneous outputs, the fault score is increased by 1. On the other side, if only one result has an erroneous output, the fault score is decreased by 1. Moreover, in VI case, the fault score is increased by the number of primary outputs.

After the fault simulation, the final candidate faults are decided with the fault selection table. Deciding the final candidate faults consists of the following steps:

1. For initial candidate faults and test patterns, calculate the fault score and the number occurrences of VI. Calculate the total fault scores of faults that are the sum of fault scores for all test patterns.

2. Select candidate faults that have the highest total fault scores in the top VI faults.

3. Check test patterns that are explained by final candidate faults to uncalculated hereafter. Also, check final candidate faults to uncalculated henceforth.

4. Recalculate total fault scores of faults that are the sum of fault scores for unexplained test patterns. Number occurrences of VI for unexplained test patterns.

5. Iterate Step 2~4. If the number of VI or the number of unexplained test pattern in Step 4 is 0, complete these steps and report final candidate faults.

In Figure 2, we illustrate this process. There are 5 initial candidate faults and 5 test patterns. Figure 2 displays the fault selection table after the completion of Step 1. ▣ represents Vector-wise Intersection. Because Vector-wise Intersection is the case that both the tester response and the simulation response have faulty responses, the patterns that are scored on the Vector-wise Intersection are failing patterns. The numbers above ▣ are fault scores that are calculated by the manner that is proposed in [5]. ▣ and fault scores are not changed. Only total fault scores and the number of VI for unexplained test patterns are recalculated in iterations.

	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5
Test Pattern 1	10	10	0	8	4
	▣	▣			
Test Pattern 2	10	2	1	5	7
	▣				
Test Pattern 3	5	3	10	7	10
			▣		▣
Test Pattern 4	3	10	10	10	2
		▣	▣	▣	
Test Pattern 5	0	10	2	10	4
		▣		▣	
#VI	2	3	2	2	1
Total Fault Score	28	35	23	40	27

Figure 2. Initial Fault Selection Table

In Step 2, the total fault score of fault 4 is larger than that of fault 2. However, the number of VI for fault 2 is 3 and it is the largest. So, fault 2 is selected to final candidate faults.

	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5
Test Pattern 1	10	10	0	8	4
	▣	▣			
Test Pattern 2	10	2	1	5	7
	▣				
Test Pattern 3	5	3	10	7	10
			▣		▣
Test Pattern 4	3	10	10	10	2
		▣	▣	▣	
Test Pattern 5	0	10	2	10	4
		▣		▣	
#VI	1	-	1	0	1
Total Fault Score	16	-	12	12	18

Figure 3. Fault Selection Table after Fault 2 Selection

Figure 3 shows the fault selection table after Step 3~4. In Step 3, test patterns 1, 4, 5 are checked because they are explained by fault 2. Also, fault 2 is checked. In Step 4, the numbers of VI and total fault scores are recalculated.

After repeating step 2, fault 5 is selected into final candidate faults. Because fault 5 has the highest total fault score in faults of which VI is the maximum.

After repeating Step 3~4, fault 1 is selected. Finally, all test patterns are explained and final candidate fault selection algorithm stops. Final candidate faults are {Fault 1, Fault 2, Fault 5}.

3 Experimental Results

For these experiments, ISCAS85 and ISCAS89 benchmark circuits are synthesized with the STD150 library. Test patterns are generated by Synopsys TetraMAX. We randomly select two faulty sites and insert the stuck-at faults. Each benchmark circuit is tested 30 times.

Table 1. The Diagnosis Results of ISCAS85 Circuits with Two Faulty sites

Circuits	SLAT [1]		The proposed algorithm	
	Detected faults	Candidate faults	Detected faults	Candidate Faults
c432	1.67	9.77	1.67	4.23
c499	1.23	31.27	1.37	3.33
c880	1.90	9.57	1.93	2.70
c1355	1.13	19.93	1.27	3.70
c1908	1.20	10.07	1.23	2.30
c2670	1.83	21.97	1.87	2.67
c3540	1.77	9.53	1.80	3.20
c5315	1.83	14.03	1.87	3.00
c6288	1.70	27.23	1.77	3.67
c7552	1.83	37.83	1.87	3.30
Average	1.61	19.12	1.66	3.21

The diagnosis results of ISCAS85 circuits are shown in Table 1. In order to verify efficiency, the SLAT algorithm [1] is tested as well. There is one difference between the SLAT algorithm described here and those described in the SLAT paper: Here, multiplets contain stuck-at faults, while the original SLAT multiplets consisted of only faulty circuit nodes. For each diagnosis, Table 1 reports the number of detected faults that are inserted faulty circuits and the number of candidate fault in top-ranked multiplets.

Table 1 illustrates that the number of faults which were diagnosed by the SLAT algorithm is 1.61, on average, and by the proposed algorithm is 1.66. The diagnostic accuracy of the proposed algorithm is almost the same as or slightly better than that of the SLAT algorithm. But the number of candidate faults of the proposed algorithm is much smaller than the number of the SLAT algorithm. The average number of candidate faults is 19.12 for the SLAT algorithm and 3.21 for the proposed algorithm. It is almost a six-fold difference. It means the proposed algorithm is more efficient.

Table 2. The Diagnosis Results of ISCAS89 Circuits with Two Faulty Sites

Circuits	SLAT [1]		The proposed algorithm	
	Detected faults	Candidate faults	Detected faults	Candidate Faults
s1196	1.77	11.53	1.87	2.47
s1238	1.77	13.87	1.90	2.77
s1488	1.70	12.00	1.83	3.23
s1494	1.77	12.47	1.83	2.87
s5378	1.90	8.67	1.93	2.93
s9234	1.93	11.97	1.93	4.00
s13207	1.83	15.63	2.00	2.80
s15850	1.97	10.80	1.97	3.20
s35932	1.93	11.80	1.93	3.57
s38594	1.97	9.30	2.00	2.70
Average	1.85	11.80	1.92	3.05

The proposed algorithm is greatly efficient for not only combination circuits but also large full scan circuits. Table 2 illustrates diagnostic results for the full scan version of ISCAS89 circuits. The diagnostic accuracy of the proposed algorithm is almost the same as or slightly better than that of the SLAT algorithm like combinational circuits. The average number of candidate faults is 11.80

for the SLAT algorithm and 3.05 for the proposed algorithm. It is almost a 4 times difference.

Table 3. The Diagnosis Results of ISCAS85 Circuits with Three Faulty Sites

Circuits	SLAT [1]		The proposed algorithm	
	Detected faults	Candidate faults	Detected faults	Candidate Faults
c432	2.03	11.93	1.97	5.10
c499	1.30	19.33	1.50	4.43
c880	2.33	14.17	2.43	4.00
c1355	0.97	7.37	1.17	4.23
c1908	1.43	11.27	1.57	3.40
c2670	2.30	30.43	2.37	4.07
c3540	2.43	14.57	2.60	4.87
c5315	2.57	15.43	2.57	3.97
c6288	2.33	56.03	2.47	4.80
c7552	2.40	31.57	2.67	4.80
Average	2.01	21.21	2.13	4.37

We executed more experiments. We randomly selected three faulty sites and inserted the stuck-at faults. Table 3 illustrates diagnostic results for ISCAS85 with three faulty sites. Table 3 illustrates that the number of faults which were diagnosed by the SLAT algorithm is 2.01, on average, and by the proposed algorithm is 2.13. The average number of candidate faults is 21.21 for the SLAT algorithm and 4.37 for the proposed algorithm. It is almost a five-fold difference. Results of the proposed algorithm are much better in c2670 and c6288.

Table 4. The Diagnosis Results of ISCAS89 Circuits with Three Faulty Sites

Circuits	SLAT [1]		The proposed algorithm	
	Detected faults	Candidate faults	Detected faults	Candidate Faults
s1196	2.23	13.63	2.63	3.57
s1238	2.30	17.00	2.63	4.37
s1488	2.27	16.17	2.57	4.80
s1494	2.27	14.50	2.60	4.13
s5378	2.73	14.23	2.83	4.20
s9234	2.73	17.83	2.77	5.43
s13207	2.72	12.41	2.76	4.21
s15850	2.77	14.10	2.80	4.53
s35932	2.53	12.67	2.53	3.77
s38594	2.87	13.90	2.93	4.17
Average	2.54	14.64	2.71	4.32

Table 4 illustrates diagnostic results for the full scan version of ISCAS89 circuits with three faulty sites. The proposed algorithm is greatly efficient for large full scan circuits. Table 4 illustrates that the number of faults which are diagnosed by the SLAT algorithm is 2.54, on average, and by the proposed algorithm is 2.71. The diagnostic accuracy of the proposed algorithm is almost the same as or slightly better than that of the SLAT algorithm.

4 Conclusions

In this paper, we propose a multiple fault diagnosis method based on a single fault simulation. The diagnosis time of our algorithm is linear with respect to the size of a circuit because our algorithm is based on a single fault simulation. Our algorithm does not have exponential time complexity due to the inherent nature of the multiple-fault-simulation problem. So, it is practical for big industrial designs. Moreover, the number of candidate faults is drastically reduced. Experimental results for ISCAS85 and full-scan version of ISCAS89 benchmark circuits prove the efficiency of the proposed algorithm.

Acknowledgment

The authors would like to appreciate CAD tool support from IC Design Education Center (IDEC).

References

[1] L. M. Huisman, "Diagnosing Arbitrary Defects in Logic Designs Using Single Location at a Time (SLAT)," *IEEE trans. Comput.-Aided Des. Intergr. Circuits Syst.*, vol. 23, no. 1, pp. 91-101, 2004.

[2] J. B. Liu and A. Veneris, "Incremental Fault Diagnosis," *IEEE trans. Comput.-Aided Des. Intergr. Circuits Syst.*, vol. 24, no. 2, pp. 240-251, 2005.

[3] Z. Wang, M. Marek-Sadowska, K. Tsai and J. Rajski, "Analysis and Methodology for Multiple-Fault Diagnosis," *IEEE trans. Comput.-Aided Des. Intergr. Circuits Syst.*, vol. 25, no. 3, pp. 558-575, 2006.

[4] S. Venkataraman and S. Drummonds, “ Poirot : Applications of a Logic Fault Diagnosis Tool, ” *IEEE Design & Test of Computers*, pp. 19-30, 2001.

[5] Yoseop Lim, Joohwan Lee, Hyungjun Cho and Sungho Kang, “ An Efficient Diagnosis Algorithm for Multiple Stuck-at Faults, ” *Proc. of SOC Design Conference*, pp. 394-397, 2006.