A Test Pattern Quality Metric for Diagnosis of Multiple Stuck-at and Transition faults *

Sarmad Tanwir
Virginia Polytechnic Institute
and State University
Blacksburg, VA, USA
sarmadt@vt.edu

Michael Hsiao
Virginia Polytechnic Institute
and State University
Blacksburg, VA, USA
hsiao@vt.edu

Loganathan Lingappan
Intel Corporation
Folsom, CA, USA
loganathan.lingappan@intel.com

ABSTRACT

The test patterns computed for detecting the manufacturing defects in the electronic circuits are generally insufficient for diagnosis. The test set compaction and failure log truncation lead to loss of critical failure observations that diagnosis might depend on. In this context, it is beneficial to know the diagnostic usefulness of failures so that we can log the more useful failures instead of logging the initial failures. In this paper, we evaluate three metrics to gauge such diagnostic usefulness in real-time by observing the circuit responses on the tester. We implement a pattern selection framework for failure logging and compare the results with those achieved by logging the initial failures. Using one of our proposed metrics, we were able to improve the diagnosis quality for a significant number of faulty instances of ISCAS'89 and IWLS'05 benchmarks having 1-7 inserted stuck-at and transition faults.

Keywords

Diagnosis, failure data, failure sampling, real-time, transition faults, stuck-at faults, metric, pattern selection

1. INTRODUCTION

Diagnosis of defective VLSI circuits plays an important role in improving the yield of the chip manufacturing processes. During diagnosis, we analyze the test, layout and design data to identify suspect locations of manufacturing defects, to reduce the cost of the physical failure analysis (PFA).

Figure 1 provides a simplified overview of the test and diagnosis process. A test suite consisting of a good mix of random and deterministic patterns targeted at stuck-at and transition faults is applied to the manufactured chips and the failures are logged. These failures that are mismatches with the good circuit response are then analyzed by offline diagnosis software tools to find a set of suspect locations. In

*Supported in part by NSF Grant 1422054 and Intel Corp.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '17, May 10-12, 2017, Banff, AB, Canada

© 2017 ACM. ISBN 978-1-4503-4972-7/17/05...\$15.00

DOI: <http://dx.doi.org/10.1145/3060403.3060450>

Figure 1: Testing and Diagnosis Process Flow

an ideal scenario, we will uncover the real defects by physically analyzing a small number of suspect locations. However, during the low yield regime, we observe a large number of failures in most of the faulty circuits, all of which cannot be logged due to memory limitations. Therefore, we must discard a significant portion of the log which likely includes information that is critical for diagnosis.

Recently, the test data explosion due to the increasing design complexity and continual technology scaling has sparked interest in methods that can minimize the failure data without a large trade-off on diagnosis quality. The authors of [6] have used machine learning techniques to predict a termination point for failure data collection. Using their proposed method, they could achieve a data reduction of 30.4% while ensuring about 90% accuracy. The main drawback of their method is that a large population of full failure data is needed for learning. In their implementation, they learnt from 90% of initial full failing data to minimize about 10% of the remaining data by 30%. Their method only improves the accuracy (not the resolution) and is not practical for adoption by the industry.

In [4] and subsequent papers, the authors propose saving a predetermined subset of response bits. They rank the output bits on the basis of the single stuck-at and transition fault detection counts and log the top ones that cover all modeled faults. Their ranking is predetermined and cannot be changed according to the observed failures. Moreover, fault model changes requires re-ranking, which is computationally expensive for real-time. Also, one particular output response ranking may not work for all failing chips that fail differently.

The authors of [3] have used rule mining to reduce failure log sizes. Their method is dependent on available diagnosis results and circuit design; however, during the low yield regime, the diagnosis data is usually not good enough to support such learning. In [2] and [1] the authors have proposed a static and dynamic N-cover algorithm for failure data minimization. They compute an N-cover requirement corresponding to the total failing frequency of an output,

which is a monotonically increasing function computed by repeating diagnosis on a small sample of failing chips. Once this function is available, the authors use it and the total failing frequencies of the outputs to find the actual number of instances to be logged. A concern with this method is that a single N-cover curve may not be applicable to all failing outputs of the circuit. Secondly, it is highly dependent on the population of defects in the initial sample. Also, the fact that different chips fail differently presents a challenge to the generalization of the technique.

In another recent paper [5], the authors have experimented with various metrics for the online selection of failure logs, some of which require additional meta-data. Their best techniques may be computationally expensive and may not be suitable for making log selection decisions in real-time for large circuits. Another limitation of their work is that they have evaluated their proposed metrics only on circuits with single stuck-at faults.

In this paper, we evaluate three such metrics that may be used to gauge the diagnostic usefulness of test patterns in real-time. In order to reduce the cost, we don't use any internal signals and rely only on the scan and primary outputs. In our experiments, using one of our proposed metrics, we could improve the diagnosis quality for 29 and 25 benchmark circuits having multiple stuck-at and transition faults respectively.

The paper is organized as follows. In Section II, we explain in detail the evaluated metrics. In Section IV and V, we explain our evaluation framework and the results of our experiments. Finally, we present our conclusions in Section VI.

Figure 2: Counting total and unique output responses

2. THE EVALUATED METRICS

We evaluate and compare three different metrics on faulty circuits having one or more stuck-at and transition faults:

2.1 Total Number of Failing Outputs

When using this metric, we intend to maximize the total number of failing outputs in the selected failing responses logged. Figure 2 shows two output responses of a hypothetical circuit to failing test patterns p_1 and p_2 . These responses r_1 and r_2 consist of the failing outputs $\{o_3, o_5, o_6, o_8, o_{10}\}$ and $\{o_2, o_5, o_6\}$ respectively. The total number of failing outputs in these two responses is $|r_1| + |r_2| = 5 + 3 = 8$. The motivation for this metric is that a larger number of failing outputs offers a larger set of observable points.

2.2 Number of Unique Failing Outputs

When using this metric, we intend to keep as many unique failing outputs in the log as possible. Using the same responses r_1 and r_2 , the number of unique failing outputs is the size of their union i.e $|r_1 \cup r_2| = |\{o_2, o_3, o_5, o_6, o_8, o_{10}\}| = 6$. The motivation for this metric is that a larger number of

Figure 3: Overall Evaluation Process

unique failing outputs offers a greater opportunity to prune the search among failing vectors.

2.3 Sum of Total and Unique Failing Outputs

This is a hybrid metric, where we try to maximize both the total and unique failures. For the above example responses r_1 and r_2 , the sum of total and unique failing outputs is $|r_1| + |r_2| + |r_1 \cup r_2| = 5 + 3 + 6 = 14$.

2.4 Algorithm for Log Selection

We use a greedy algorithm for selecting failing responses on the basis of the metrics enumerated above. First, we let the log fill up with the initial failing responses. Then, with each new failing pattern, we determine how much the metric would improve when its response replaces each of the existing responses. We replace an existing response with the new response such that we achieve the maximum gain. In case of no gain, we discard the new response.

3. EVALUATION FRAMEWORK

We use 29 full scan circuits from the ISCAS'89 benchmarks and 8 full scan circuits from the Opencores library in the IWLS'05 benchmarks to test the effectiveness of the considered metrics.

3.1 Test Generation

We use a state-of-the-art commercial tool to generate two separate test sets for stuck-at and transition faults. For the transition faults, we generate double-capture or broadside patterns. Table 1 lists the characteristics of the generated tests. Column 1 lists the benchmark names. Columns 2 and 3 list the number of generated tests and fault coverages for stuck-at faults respectively. Columns 4 and 5 list the same for the transition faults.

3.2 Generation of Faulty Circuits

For our evaluation experiments, we create two sets of faulty circuits each for stuck-at and transition fault models. The first set consists of 300 circuits with a single stuck at fault each. The second set also consists of 300 faulty circuits except that each circuit has 2 to 7 inserted faults. We determine at random the faults and the number of faults inserted in these circuits.

3.3 Test Application and Response Collection

We apply the computed tests to the generated faulty circuits. The commercial tool is used to compute the responses that make up the failure files. From the full failure files, we create two shortened failure files. In the first file, we keep

Table 1: Test Statistics

Bench	Stuck-at		Transition	
	# Tests	FC	# Tests	FC
s00027	6	100.00%	8	100.00%
s00208	34	100.00%	37	89.71%
s00298	28	100.00%	33	85.65%
s00344	19	100.00%	39	95.55%
s00349	18	99.91%	43	95.39%
s00382	29	100.00%	38	83.76%
s00386	69	100.00%	55	84.12%
s00400	28	98.50%	40	82.33%
s00420	71	100.00%	87	86.79%
s00444	31	98.54%	43	82.36%
s00510	61	100.00%	80	91.80%
s00526	62	100.00%	68	70.48%
s00641	37	100.00%	65	96.89%
s00713	36	99.01%	73	93.60%
s00820	104	100.00%	125	84.76%
s00832	104	99.20%	128	83.65%
s00838	148	100.00%	183	85.61%
s01196	129	100.00%	204	99.94%
s01238	145	97.37%	216	97.16%
s01423	59	99.26%	91	91.11%
s01488	116	100.00%	143	93.43%
s01494	115	99.57%	143	93.07%
s09234	196	94.63%	381	90.14%
s13207	288	99.28%	437	91.50%
s15850	184	98.32%	310	89.07%
s35932	55	91.06%	85	89.33%
s38417	165	99.78%	267	98.90%
s38584	198	96.89%	423	92.47%
aes_core	377	99.95%	658	99.96%
des3_area	128	100.00%	263	99.98%
des3_perf	128	99.99%	206	100.00%
spi	339	99.97%	761	99.15%
sysc_aes	154	99.39%	361	95.16%
sysc_des	92	99.97%	155	99.35%
wb_dma	190	99.41%	1994	99.01%

the failing outputs corresponding to the initial $k = 5$ failing patterns. In the second file, we store the failing outputs of the selected k failing patterns, using the proposed metrics.

3.4 Diagnosis

We perform diagnosis on the truncated and selected failure files. A definitive improvement in accuracy and resolution in a large number of benchmarks would imply that the metric is good for improving diagnosis. In our comparison, we look at two measures.

3.4.1 Accuracy

We define accuracy as the number of actual (inserted) faults reported by the diagnosis tool among the suspects divided by the total number of actual faults existing (inserted) in the circuit. For example, if 6 faults were inserted in a circuit and only 2 of those ended up in the list of suspects. The accuracy of diagnosis is $2/6 \times 100\% = 33.33\%$.

3.4.2 Average Inverse Hit Index

In the context of our experiments, the hit index is the in-

Figure 4: Comparison for diagnosis quality for circuits with single stuck-at faults

dex of an inserted fault in the ranked list of suspects. If a fault does not end up in the suspects, we assign it a hit index of ∞ . For each faulty circuit instance, we take the reciprocal of the hit indices and calculate the average. The higher this average, the higher is the diagnosis quality. For example if three faults were inserted in the circuit and their corresponding hit indices were $\{1, \infty, 9\}$. The average inverse hit index would be $\frac{1/1 + 1/\infty + 1/9}{3} = \frac{1 + 0 + 0.11}{3} = 0.37$.

4. EXPERIMENT RESULTS

Tables 2 and 3 provide the results for about 300 faulty instances for each benchmark having 2-7 stuck-at and transition faults respectively. Column 1 of Table 2 lists the benchmark circuits. Column 2 and 6 list the average accuracy and inverse hit index for the baseline results obtained using the 5 initial failing patterns (and all passing patterns from the test set). Columns 3, 4 and 5 state the diagnosis accuracy achieved when we optimized the log by total failing outputs, unique failing outputs and both total and unique failing outputs respectively. Whereas, columns 7, 8 and 9 provide the average inverse hit index for these three cases.

As can be seen from the table, optimizing the total number of failures did improve the accuracy and resolution for some benchmarks. However, for most benchmarks, we see a drop in the diagnosis quality. On the other hand, when we optimize the number of unique failing outputs, we see an improvement in diagnosis quality for 29 out of 35 benchmarks for stuck-at faults and 25 out of 28 benchmarks for transition faults. When we optimize both, the result is not as good because we see a deterioration in diagnosis quality for more number of benchmarks.

Fig 4 depicts the improvement in diagnosis resolution for circuits having single faults using the three metrics. It shows the ratios of the number of circuits that were diagnosed better with the application of the metric to the circuits diagnosed better using the initial failures. As may be observed, the most significant improvement in diagnostic resolution was achieved by optimizing the unique failing outputs that yielded better diagnosis for **28x**, and **12x** more circuits having single stuck at and transition faults respectively.

5. CONCLUSION

In this paper, we evaluated three log selection metrics for improving VLSI diagnosis. Using the metric that optimizes the number of unique failing outputs in the failure log, we

Table 2: Metric Comparison for Circuits with Multiple Stuck-at Faults

Bench	Avg Accuracy				Avg Inv Hit Index			
	Init	Selective			Init	Selective		
		T	U	B		T	U	B
s00027	36.8	38.1	37.8	38	0.3	0.3	0.3	0.3
s00208	28.4	30.7	30.2	30.6	0.2	0.2	0.2	0.2
s00298	33.8	30.9	32.4	31.6	0.3	0.3	0.3	0.3
s00344	40	35.5	39.9	37	0.3	0.3	0.3	0.3
s00349	38.9	34.6	38.7	34.7	0.3	0.3	0.3	0.3
s00382	30.6	32.4	34.5	32.5	0.2	0.2	0.2	0.2
s00386	35.8	32.9	38.8	36.1	0.3	0.3	0.3	0.3
s00400	34.2	33.1	34.9	34.1	0.2	0.2	0.2	0.2
s00420	26.8	31.7	30.5	31.7	0.2	0.2	0.2	0.2
s00444	32.2	32.9	34	32.7	0.2	0.2	0.3	0.2
s00510	33.5	32.4	35.6	33.6	0.3	0.3	0.3	0.3
s00526	33.3	32.3	34.6	32.4	0.3	0.3	0.3	0.3
s00641	41.7	37.2	43.5	39.4	0.4	0.3	0.4	0.3
s00713	40.5	34.2	42.7	37.1	0.3	0.2	0.3	0.2
s00820	31.7	31.8	33.2	32.9	0.3	0.3	0.3	0.3
s00832	30.8	32.9	33.9	33.2	0.3	0.3	0.3	0.3
s00838	34.8	35.7	38.1	36.8	0.3	0.2	0.3	0.3
s01196	40.8	35.8	44.3	37.3	0.4	0.3	0.4	0.3
s01238	38.3	35	40.9	36.8	0.3	0.3	0.3	0.3
s01423	43.2	36	44.1	36.6	0.3	0.3	0.3	0.3
s01488	34	32.6	34.2	32.8	0.3	0.3	0.3	0.3
s01494	34.2	33	37.1	36.4	0.3	0.3	0.3	0.3
s09234	39.5	35.8	44.5	38.9	0.3	0.3	0.3	0.3
s13207	40.4	43.7	46.6	43.8	0.3	0.3	0.3	0.3
s15850	41.9	38.4	45.6	39.4	0.3	0.2	0.3	0.3
s35932	36.5	34.8	35.7	34.7	0.3	0.2	0.2	0.2
s38417	43.2	37.4	46.6	39.3	0.4	0.3	0.4	0.3
s38584	42.5	40.6	47.7	41.8	0.3	0.3	0.4	0.3
aes_core	45.9	38.8	54.3	42.7	0.4	0.3	0.5	0.4
des3_area	35.3	28.8	34.7	29.9	0.3	0.2	0.3	0.2
des3_perf	46	40.8	50.5	41.4	0.4	0.3	0.5	0.3
spi	35.6	39.9	39.5	39.1	0.3	0.3	0.3	0.3
sysc_aes	42.8	37.1	43.6	38.7	0.4	0.3	0.4	0.3
sysc_des	36.7	34.3	36.6	35.1	0.3	0.3	0.3	0.3
wb_dma	43.6	42.5	46.8	41.2	0.3	0.3	0.3	0.3

were able to achieve a significant improvement in both accuracy and resolution for a majority of ISCAS'89 and IWLS'05 circuits having single and multiple stuck-at and transition faults. Increasing the number of unique failing outputs is effective because the internal signals explaining those are likely to be different. Especially for circuits with multiple faults, different unique failing outputs may correspond to different activated faults. This helps us capture a greater number of symptoms, thus improving the accuracy.

6. REFERENCES

[1] S. Bodhe, M. E. Amyeen, C. Galendez, H. Mooers, I. Pomeranz, and S. Venkataraman. Reduction of diagnostic fail data volume and tester time using a dynamic n-cover algorithm. In *2016 IEEE 34th VLSI Test Symposium (VTS)*, pages 1–6. IEEE, 2016.

Table 3: Metric Comparison for Circuits with Multiple Transition Faults

Bench	Avg Accuracy				Avg Inv Hit Index			
	Init	Selective			Init	Selective		
		T	U	B		T	U	B
s00027	23.9	23.5	23.9	23.3	0.16	0.16	0.16	0.16
s00208	15.4	16.3	14.4	15.4	0.10	0.10	0.09	0.09
s00298	23.4	23.1	25.0	22.9	0.15	0.16	0.17	0.16
s00344	30.0	29.9	34.7	30.3	0.19	0.19	0.22	0.20
s00349	26.6	27.1	30.1	27.1	0.17	0.17	0.19	0.17
s00382	29.9	29.6	31.6	31.0	0.20	0.20	0.21	0.21
s00386	24.1	24.3	24.3	25.0	0.15	0.16	0.16	0.16
s00400	25.4	24.5	26.3	25.5	0.18	0.17	0.18	0.18
s00420	19.1	18.6	18.5	17.6	0.11	0.10	0.10	0.10
s00444	24.2	24.8	25.2	25.5	0.16	0.16	0.16	0.17
s00510	20.4	22.5	22.2	22.5	0.12	0.14	0.14	0.14
s00526	22.8	22.7	23.4	24.0	0.14	0.14	0.14	0.15
s00641	40.4	29.6	40.6	32.0	0.20	0.15	0.21	0.17
s00713	38.7	33.3	41.2	35.0	0.16	0.15	0.19	0.17
s00820	21.8	22.1	23.7	22.2	0.12	0.12	0.14	0.13
s00832	20.9	20.7	21.4	21.2	0.12	0.13	0.13	0.13
s00838	15.2	17.0	15.9	16.7	0.09	0.09	0.09	0.09
s01196	40.5	34.0	41.0	36.8	0.23	0.21	0.25	0.22
s01238	41.5	38.1	46.4	40.6	0.25	0.23	0.28	0.25
s01423	28.1	24.1	30.6	26.1	0.17	0.15	0.19	0.16
s01488	24.1	25.1	24.0	24.6	0.14	0.15	0.15	0.15
s01494	24.4	24.5	26.3	25.0	0.13	0.14	0.14	0.14
s09234	35.9	31.8	43.5	34.5	0.16	0.16	0.19	0.17
s13207	35.3	31.6	41.1	30.9	0.15	0.15	0.18	0.15
s15850	38.4	32.8	43.0	33.4	0.18	0.15	0.20	0.16
s38417	42.6	32.5	43.9	34.3	0.21	0.17	0.22	0.17
s38584	37.3	33.0	45.3	35.2	0.21	0.18	0.24	0.19

[2] S. Bodhe, M. E. Amyeen, I. Pomeranz, and S. Venkataraman. Diagnostic fail data minimization using an-cover algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(3):1198–1202, 2016.

[3] C. Bolchini, E. Quintarelli, F. Salice, and P. Garza. A data mining approach to incremental adaptive functional diagnosis. In *2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, pages 13–18. IEEE, 2013.

[4] K.-J. Lee, W.-C. Lien, and T.-Y. Hsieh. Test response compaction via output bit selection. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(10):1534–1544, 2011.

[5] S. Tanwir, S. Prabhu, M. Hsiao, and L. Lingappan. Information-theoretic and statistical methods of failure log selection for improved diagnosis. In *Test Conference (ITC), 2015 IEEE International*. IEEE, 2015.

[6] H. Wang, O. Poku, X. Yu, S. Liu, I. Komara, and R. D. Blanton. Test-data volume optimization for diagnosis. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 567–572. IEEE, 2012.