Machine Learning-Based Volume Diagnosis

Seongmoon Wang Wenlong Wei
{swang,wwei}@nec-labs.com NEC Labs. America, Princeton, New Jersey

Abstract

In this paper, a novel diagnosis method is proposed. The proposed technique uses machine learning techniques instead of traditional cause-effect and/or effect-cause analysis. The proposed technique has several advantages over traditional diagnosis methods, especially for volume diagnosis. In the proposed method, since the time consuming diagnosis process is reduced to merely evaluating several decision functions, run time complexity is much lower than traditional diagnosis methods. The proposed technique can provide not only high resolution diagnosis but also statistical data by classifying defective chips according to locations of their defects. Even with highly compressed output responses, the proposed diagnosis technique can correctly locate defect locations for most defective chips. The proposed technique correctly located defects for more than 90 % (86 %) defective chips at 50x (100x) output compaction. Run time for diagnosing a single simulated defect chip was only tens of milli-seconds.

1 Introduction

Reducing time to ramp up yield is crucial for profitability of semiconductor companies. Even after the process has been stabilized, process excursion can also reduce yield. Feature-related systematic defects are increasing fast. At 45nm technology and below, traditional yield learning techniques such as in-line inspection and memory bitmapping become less effective due to small features and large number of metal layers. Scan-based volume diagnosis methods have attracted great attentions recently [14, 7, 6] as alternative yield learning techniques. Volume diagnosis uses manufacturing test data to locate defects. Hence, for successful volume diagnosis, the capability to process large volume of data in reasonable time is a key requirement. Another required feature is obtaining accurate statistical data from failing data. Volume diagnosis should quickly provide defect pareto prior to conducting physical failure analysis.

The volume of data that should be processed for volume diagnosis can be huge, especially during the ramp-up period. Hence, only part of output responses are sampled for diagnosis or output responses should be compressed by output compaction. Since the volume of output responses for a large SoC (system-on-chip) often exceeds memory capacity of the tester, output compaction is widely used to reduce output test data volume [10, 13, 12]. Almost all output compression techniques employ lossy compression. Hence diagnostic resolution can be severely affected by loss of information when output responses are highly compressed.

Most scan diagnosis techniques [14, 5, 12] that support compressed output responses consist of three steps. The first step is to find scan cells that capture errors. If output responses are compressed, it is difficult to accurately find error capturing scan cells. In the second step, a set of candidate faults are identified from error capturing scan cells, which are identified in the first step. Due to ambiguity of the first step, sets of candidate faults for compressed output responses are larger. Finally, the collected candidate faults are ranked from the most likely one to the least likely one.

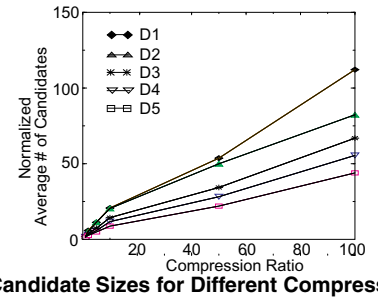

Figure 1. Candidate Sizes for Different Compression Ratios

We have conducted extensive experiments to understand relationship between candidate sizes and compression ratios. To achieve desired compression ratio Cx, we inserted a simple XOR tree after outputs of C scan chains, i.e., C scan chains are connected to an XOR tree. For diagnosis, we have implemented the SLAT algorithm [2]. Results of the experiments are summarized for 5 different industrial designs, $D1, \ldots, D5$, in Figure 1. The X-axis represents the compression ratio and the Y-axis represents the number of candidate defects, which is normalized to the number of candidate defects for the no-compression case, i.e., observing output responses directly without any compactor. Roughly, the number of candidate defects increases linearly as the compression ratio increases for all circuits while the slope varies circuit by circuit. For all circuits except $D5$, numbers of candidate defects for 100x compression are more than 50 times (up to 120 times) larger than those of candidate defects for no-compression. Although Rajski et al. [12] show that high resolution diagnosis is possible with data produced by their 1000x output compactor, these results are obtained under the assumption that all defects manifest themselves as single stuck-at faults. It has been observed that there exist many defects that do not manifest themselves as single stuck-at faults. The SLAT algorithm [2] employs fault model independent diagnosis to address this issue.

Built-in self-test (BIST) typically employs multiple input signature registers (MISRs) to achieve very high compression. Techniques proposed in [9, 11, 16] improve diagnostic resolutions by collecting multiple signatures for a test sequence; multiple signatures are obtained by applying the same test sequence multiple times, each with a different polynomial of the MISR or using multiple MISRs with different polynomials. For high resolution, a large number of signatures are required. The technique proposed in [8] generates parity data in addition to MISR signatures. Since this technique requires additional parity data and a separate signature for each test pattern, achieving high compression is difficult with this technique.

978-3-9810801-5-5/DATE09 © 2009 EDAA

Software-based self-diagnosis techniques proposed in [3, 4] enable defective chips to diagnose themselves. The critical problem with these techniques is that diagnostic results are not credible because diagnostic software is run on defective chips. If the defect(s) is activated by the diagnosis routine, then the diagnostic result becomes garbage. This situation is analogous to obtaining a diagnostic opinion from a mentally ill doctor. Using complex hardware to improve diagnosis resolution can also face the mentally-ill-doctor dilemma. If hardware additionally inserted to improve diagnostic resolution takes 10 % of chip area, then there are 10 % of chances that diagnosis results are corrupted by defects in the inserted hardware.

As discussed above, pinpointing the defect location from highly compressed output responses is extremely difficult. This paper proposes an entirely different diagnosis approach, which is based on machine learning. Machine learning has been used to diagnose human diseases, machinery faults, powerline faults, etc. Unlike traditional methods, the proposed method does not require the procedure to identify failing scan cells, which is in nature very inaccurate with highly compressed output data. The proposed technique can provide not only high resolution diagnosis but also statistical data by classifying defective chips according to locations of their defects (a standard formulation of machine learning is the classification problem). The classification data can be used to identify systematic yield problems and guide the sampling step to select a few defective chips for failure analysis. The proposed diagnosis technique is fault model independent and hence can locate defects that do not manifest themselves as single stuck-at faults. The proposed technique requires no additional hardware. Since a diagnosis process of the proposed method is merely evaluating several decision functions, run time complexity of the proposed method is several orders of magnitude lower than traditional diagnosis methods.

The rest of this paper is organized as follows: Section 2 presents the key ideas and motivation of the proposed technique. Section 3 describes the training procedure. The procedure for diagnosis is described in Section 4. Experimental results are presented in Section 5. Section 6 gives conclusions.

2 Motivation and Key Ideas

Consider the stuck-at-1 (s-a-1) and the stuck-at-0 (s-a-0) fault at circuit line l. Although test patterns that detect the s-a-1 fault never detect the s-a-0 fault and vice versa, there will be several common scan cells that capture errors of both the s-a-1 and the s-a-0 fault. Likewise, some of scan cells that capture errors of the stuck-at faults at l will capture errors of bridging defects at l. Even if conditions to activate defects are different, once activated, fault effects of defects at the same circuit line will propagate through similar paths and be captured into some common scan cells. This commonality in failing scan cells among defects at the same circuit line can be extended to defects in the same fanout free region.

In Figure 2, assume that test pattern p_i detects the s-a-1 fault f_a at circuit line l_a, which is the output of fanout free region FFR_x. The activated fault effect at l_a propagates to scan outputs so_{22}, so_{56}, and so_{99} through internal circuit lines. Assume that p_i sensitizes the path from l_b to l_a (the output of FFR_x), i.e., the fault effect at l_b propagates to l_a. If a defect at l_b, no matter what type of defect it is, is activated by p_i, then the fault effect of the defect at l_b propagates to exactly the same scan cells in which the fault effects of f_a are captured.

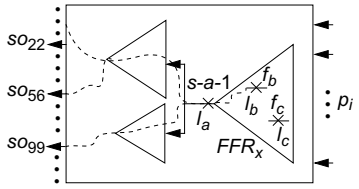

Figure 2. Faults Belonging to the Same Fanout Free Region

In Figure 2, assume that defect f_c is not activated by p_i. However, if there are other test patterns that activate f_c and sensitize the path from l_c to l_a, then many of scan cells that capture the fault effect of f_a when p_i is applied will also capture the fault effect of f_c. When output responses are compressed by space compaction, most fault effects of defects that are located in the same fanout free region will propagate to same output compactors and observed at same scan cycles.

As described in the above paragraph, defects that are located in the same fanout free region will have strong correlations in scan cells that capture errors. This property allows machine learning algorithms to classify defective chips according to their defect fanout free regions. Training is performed with compressed output responses that are produced by different faulty circuits, which are made by injecting faults into each class (fanout free region) in the circuit, and fault simulating them with a given set of test patterns. Sizes of classes determine diagnostic resolution of the proposed method. According to our extensive experiments, most fanout free regions are small (include only 2-4 gates). Large fanout free regions can be split into smaller areas to enhance diagnosis resolution.

Even though the proposed method locates defect areas rather than defect circuit lines, since defined areas are small, we can easily pinpoint defects with scanning electro microscopy (SEM), E-beam inspection, or other inspection equipment. Even with traditional diagnosis methods, it is necessary to inspect candidate defects to verify if a defect really exists at one of candidate circuit lines (unless the diagnosis returns only one candidate defect with very high certainty). Locating one highly suspicious area is more useful than locating several suspicious circuit lines. If the number of candidate circuit lines is large and these circuit lines are at a distance to each other, then inspection task will be very time consuming.

Statistical information can be obtained from the classification results without further processing data. Statistical information can be used to tentatively quantify systematic defects and random defects prior to detailed inspection such as SEM, e.g., classes (fanout free regions) that have large populations will contain systematic defects while classes that have small populations will contain random defects. Since defect inspection is time consuming, the number of dies that can be inspected should be limited. Classification data can be used to guide the sampling process such that defective chips that best represent other chips in each class are selected for inspection.

3 SVM Training

In this paper, defective chips are classified using an publicly available machine learning software package called *MiLDe* [1]. MiLDe is an integrated development environment with a suite of machine learning tools. However, in this paper we use only the Support Vector Machine (SVM) tool of MiLDe. The SVM is a training algorithm for classification and regression [15]. In the proposed diagnosis method, training data are prepared from output responses generated from

faulty versions of the design, which are created by injecting stuck-at faults into each of fanout free regions.

Note that target objects of the proposed diagnosis technique are defect fanout free regions (see Section 2). Since most fanout free regions are small, a typical million gate design can have hundreds of thousands of fanout free regions. The total memory usage of an SVM tool is determined by the number of decision functions. Hence, building decision functions for all fanout free regions for a million gate design at once can blow up memory. To avoid memory blow-up and also reduce run time complexity of diagnosis, we divide the entire design into many sub-circuits (partitions), each of which is much larger than even the largest fanout out free region, according to the output compaction structure of the design and conduct training for each sub-circuit separately. Then fanout free regions are identified for each partition.

Each fanout free region is defined as a class for training; all faulty versions of the design that have faults in the same fanout free region belong to the same class. A faulty version of the design is created by injecting a **single** stuck-at fault. Note that faults that are injected to make training data are independent of defects that occur in real defective chips. Several faulty versions of the design are created for each class. Output responses of each faulty version are collected by fault simulating it with test patterns to be applied during test application.

4 Volume Diagnosis

If a chip fails during test application, output responses produced by the failed chip are transferred for diagnosis. First, scan slices that capture errors are identified. Then the partitions that produced any error are identified from those scan slices. Assume that there exits only one defect (fanout free region). Since there is only one defect fanout free region, any partition that produced errors contains the defect fanout free region. Hence, we can arbitrarily select one partition among the partitions that produced errors to locate the defect fanout region. The training results of the partition, which was identified from the error scan slices, is loaded along with the output responses of the failed chip. The defect fanout free region is located by simply finding the fanout free region (class) for which decision function gives the largest value.

If there are multiple defects, there can be more than one defect fanout free region and these defect fanout free regions may distribute across multiple partitions. Hence, selecting one partition arbitrarily may not locate all defect fanout free regions. A straightforward solution is to repeat the procedure described in the above paragraph for every partition that produced errors. If there are a large number of partitions that produce errors, this can be very time consuming. We are currently investigating efficient algorithms to select best partitions to extend the proposed diagnosis technique to multiple defect cases.

5 Experimental Results

We conducted experiments to verify the feasibility of the proposed volume diagnosis method with the 2 largest ITC'99 benchmark circuits and 4 industrial designs. Experimental results are shown in Table 1. Since defective silicon chips were not available to us, we instead used simulation in the experiments. To simulate defective silicon chips, which fail tests, we made faulty designs by injecting faults into original designs like we made faulty versions of designs for training. To prevent confusion between faulty versions of designs we made for training and faulty versions of designs that we made to simulate defective silicon chips, we call the former *faulty designs for training* and the latter *simulated defect chips* in the rest of the paper. In contrast to faulty designs for training where only single stuck-at faults are injected, bridging faults as well as stuck-at faults were injected into simulated defect chips. For each design, we selected three circuit partitions randomly for the experiments. Then we made up to 20 (many fanout free regions were too small to inject 20 different single stuck-at faults) different faulty designs for training for each fanout free region in the three randomly selected partitions. To make simulated defect chips, 100 randomly selected faults were injected into each of the three selected partitions, i.e., we made 100 simulated defect chips for each of the selected partitions. We injected 50 single stuck-at faults and 50 AND/OR bridging faults (a few of them were bridged across two different fanout free regions). We avoided inserting any fault into the circuit lines where stuck-at faults are already inserted in the corresponding faulty design for training.

Table 1. Experimental Results

CKT name	*50x*					*100x*				
	time (sec)		*diag success %*			*time (sec)*		*diag success %*		
	train	*diag*	*1st*	*2nd*	*+3rd*	*train*	*diag*	*1st*	*2nd*	*+3rd*
b18	185	1.4	89.5	2.1	93.7	160	1.2	84.7	3.1	88.8
#FF=3308	210	1.5	83.0	4.3	89.4	175	1.4	90.3	3.2	94.6
#pat=1459	175	1.1	91.7	4.2	96.9	181	0.6	85.3	4.2	90.5
b19	196	0.6	90.1	0	94.6	152	1.0	91.0	4.5	95.5
#FF=6618	148	0.6	88.2	5.4	93.5	205	0.7	93.1	2.3	95.4
#pat=1579	159	0.6	86.0	3.2	89.2	134	0.7	87.1	1.1	89.2
D1	181	2.4	90.2	4.3	94.6	45.8	5.3	79.0	4.9	85.2
#FF=2455	273	2.8	87.8	3.3	91.1	936	2.6	84.3	1.1	86.5
#pat=2156	554	5.0	81.8	4.5	87.5	1300	4.4	77.3	4.0	82.7
D2	119	1.4	78.4	14.6	92.5	1427	3.4	70.9	2.3	73.3
#FF=6796	2100	4.4	79.6	3.2	82.8	4613	8.5	75.4	5.4	80.4
#pat=976	1608	2.3	77.2	2.2	79.3	2014	8.0	78.5	1.1	82.8
D3	934	3.8	84.4	5.2	89.6	2731	2.4	75.5	3.1	79.6
#FF=5014	1603	7.8	83.0	5.3	89.4	3547	5.5	84.9	5.4	91.4
#pat=545	1264	3.9	88.3	5.3	94.7	4311	10.8	82.5	5.2	88.7
D5	95.9	1.1	84.2	3.2	89.5	4122	6.8	85.6	2.1	87.6
#FF=64K	1042	8.3	80.6	5.1	85.7	4122	6.8	84.2	3.2	89.5
#pat=1510	987	3.6	89.7	1.0	90.7	5408	13.9	70.1	2.1	73.2
Average	657	2.9	85.2	4.2	90.3	1977	4.7	82.2	3.2	86.4

We used only the radial basis function (RBF) kernel with the same parameter values ($\gamma = 5 \times 10^{-3}$ and $C = 10$) for all circuits [1]. The parameter values that gave the highest diagnosis success rate were different for different circuits (different partitions even in the same circuit). However, perfectively tuning parameter values for each partition will be unrealistic due to its prohibitive run time. Hence, we selected parameter values that gave generally good results for most partitions and used them for all different partitions and circuits.

In the column labeled *CKT name*, *#FF* shows the number of scan cells in the circuit and *#pat* shows the number of test patterns applied. For all cases, we used simple XOR trees to compress output responses. The columns under the heading *50x* (*100x*) shows results obtained by using 50x (100x) space compaction, i.e., inputs of a simple XOR tree are connected to outputs of 50 (100) scan chains. The columns under the

heading *diag success %* show the percent of simulated defect chips for which the proposed diagnosis technique correctly located defect fanout free regions; the column *1st* (the column *2nd*) shows the percentage of the fanout free regions for which decision functions gave the (second) largest value. For on average 85.2 % (82.2 %) simulated defect chips, the fanout free region for which decision function gave the largest value was the defect fanout free region at 50x (100x) compaction, i.e., diagnosis success. The column *+3rd* shows the sum of the first (the column *1st*), the second (the column *2nd*), and the third (the column *3rd*) match. When the third match is considered as well, success rates of the proposed method are 90 % or higher for most circuits except D2. Although there are some exceptions (e.g., b19), diagnosis success rates are higher for most cases with 50x compaction than 100x. The reason that diagnosis success rates for 100x compaction are higher than those for 50x compaction for b19 is merely a coincidence (note that we randomly selected three partitions for each circuit).

Experiments for every case shown in Table 1 was run on a 3.2 GHz Intel Xeon processor running Linux. For most cases, time taken for diagnosing 100 simulated defect chips (the columns *diag* under the heading *time (sec)*) is only a few seconds. In other words, average diagnosis time per a single simulated defect chip is only tens of milli-seconds (considering that a typical test session can last tens of seconds, this is several orders of magnitude shorter than typical test application time). This means that the proposed volume diagnosis technique requires virtually no extra time. Training time is generally determined by the size of the partition being trained. Since sizes of partitions vary, training time also varies greatly among different cases from 45.8 to 4613 seconds.

Another advantage of the proposed technique is that even failed diagnoses do not corrupt defect statistics. In almost all cases where diagnosis failed, i.e., the fanout free regions for which the decision function returned the largest value was not the defect fanout free region, even that largest value was much smaller than 1. In other words, even in a case where we were not able to correctly locate the defect fanout free region, we were informed that the diagnosis failed. Therefore, we can avoid mixing wrong diagnosis results with correct diagnosis results. Even if we discard 15 % failed diagnosis results, since defect statistics can be built with 85 % of correct diagnosis results, it will give enough information for yield learning.

6 Conclusions

In this paper, a novel volume diagnosis method is proposed. The proposed technique uses machine learning techniques instead of traditional cause-effect and/or effect-cause analysis. The proposed diagnosis technique exploits the fact that errors of faults in the same fanout free regions propagate through common paths and observed at common scan cells. The proposed technique has several advantages over traditional cause-effect and effect-cause diagnosis methods, especially for volume diagnosis. In the proposed method, since the time consuming diagnosis process is reduced to merely evaluating several decision functions, run time complexity is much lower than traditional cause-effect and effect-cause diagnosis methods. Diagnosis time for the proposed method is negligible compared to typical test application time. The proposed technique can provide not only high resolution diagnosis but also statistical data by classifying defective chips according to locations of their defects. Even with highly compressed output responses, the proposed diagnosis technique can correctly locate defect locations for most defective chips. Since success or failure of each diagnosis is clearly known, even failed diagnoses do not corrupt defect statistics. Hence, more reliable statistical data can be obtained.

Experimental results clearly demonstrate feasibility. The proposed technique correctly located defects for more than 90 % defective chips when 50x output compaction was employed. Even with 100x compaction, the proposed diagnosis technique was able to locate defects correctly for more than 86 % defective chips. Run time for diagnosing a single simulated defect chip was only a few milli-seconds for most cases.

References

[1] *http://ml.nec-labs.com/software.php?project=milde*.

[2] T. Bartenstein, D. Heaberlin, L. Huisman, and D. Sliwinski. Diagnosing Combinational Logic Designs Using the Single Location At-a-Time (SLAT) Paradigm. In *Proceedings International Test Conference*, pages 287–296, 2001.

[3] P. Bernardi, E. Sánchez, M. Schillaci, G. Squillero, and M. S. Reorda. An Effective Technique for Minimizing the Cost of Processor Software-Based Diagnosis in SoCs. In *Proceedings Design Automation and Test in Europe Conference and Exhibition*, pages 412–417, 2006.

[4] L. Chen and S. Dey. Software-Based Diagnosis for Processors. In *Proceedings IEEE-ACM Design Automation Conference*, pages 259–262, 2002.

[5] W.-T. Cheng, M. Sharma, T. Rinderknecht, L. Lai, and C. Hill. Signature Based Diagnosis for Logic BIST. In *Proceedings International Test Conference*, pages 1–9, 2007.

[6] M. Keim, P. Muhmenthaler, H. Tang, M. Sharma, J. Rajski, C. Schuermyer, and B. Benware. A Rapid Yield Learning Flow Based on Production Integrated Layout-Aware Diagnosis. In *Proceedings International Test Conference*, pages 1–10, 2006.

[7] A. Leininger, P. Muhmenthaler, W.-T. Cheng, N. Tamarapalli, W. Yang, and H. Tsai. Compression Mode Diagnosis Enables High Volume Monitoring Diagnosis Flow. In *Proceedings International Test Conference*, pages 1–10, 2005.

[8] J. C.-M. Li, H.-M. Lin, and F.-M. Wang. Column Parity Row Selection (CPRS) BIST Diagnosis Technique: Modeling and Analysis. *IEEE Trans. on Computers*, 56(3), Mar. 2007.

[9] C. Liu and K. Chakrabarty. Identification of Error-Capturing Scan Cells in Scan-BIST with Applications to System-on-Chip. *IEEE Trans. on Computer-Aided Design of Integrated Circuit and System*, 23(10):1447–1459, Oct. 2004.

[10] S. Mitra and K. S. Kim. X-Compact: An Efficient Response Compaction Technique for Test Cost Reduction. In *Proceedings International Test Conference*, pages 311–320, 2002.

[11] J. Rajski and J. Tyszer. Diagnosis of Scan Cells in BIST Environment. *IEEE Trans. on Computers*, 48(7), July 1999.

[12] J. Rajski, J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab. X-Press Compactor for 1000x Reduction of Test Data. In *Proceedings International Test Conference*, pages 1–10, 2006.

[13] J. Rajski, J. Tyszer, C. Wang, and S. M. Reddy. Convolutional Compaction of Test Responses. In *Proceedings of International Test Conference*, pages 745–754, 2003.

[14] H. Tang, S. Manish, J. Rajski, M. Keim, and B. Benware. Analyzing Volume Diagnosis Results with Statistical Learning for Yield Improvement. In *Proceedings European Test Symposium*, 2007.

[15] V. Vapnik. *Support Vector Method for Function Approximation, Regression, Estimation, and Signal Processing*. Wiley Interscience, Reading, M.A., 1998.

[16] Y. Wu and S. M. I. Adham. Scan-Based BIST Fault Diagnosis. *IEEE Trans. on Computer-Aided Design of Integrated Circuit and System*, 18(2):203–1188, Feb. 1999.