Yield Learning with Layout-aware Advanced Scan Diagnosis

Jayanth Mekkoth, Murali Krishna, and Jun Qian
Cisco Systems Inc., San Jose CA USA

Nagesh Tamarapalli, Wu-Tung Cheng, Jan Tofte, and Martin Keim
Mentor Graphics Corp., Wilsonville OR USA

Abstract

Even after ramping up the yield to mature level from initial level, yield is seldom constant and is continually varying. Excursions occur due to variances in the large number of tools, materials, and people involved in the complex IC fabrication. Quickly identifying and correcting the root causes of yield excursions is extremely important to achieving consistent, predictable yield, and maintaining profitability.

This paper presents a case study of yield learning through a layout-aware advanced scan diagnosis tool to resolve a significant yield excursion for an IC containing 1 Million logic gates, manufactured at 130 nm technology node.

1. Introduction

With each technology node it has been observed that initial and mature yields are trending downwards and yield ramp is taking longer. Even after yield is ramped up to a mature level, it continually varies as excursions occur due to variances in the large number of tools, materials, and people involved in the complex IC fabrication. Quickly identifying and correcting the root causes of yield excursion is extremely important to achieving consistent, predictable yield, and maintaining profitability.

Given the increasingly large number of devices, metal layers, and the buried defects, traditional fault isolation techniques such as in-line inspection are proving to be less effective. For the current nanometer technology designs, scan-based diagnosis plays a crucial role in accelerating the yield learning by quickly isolating the failures and identifying the defect type(s). The information from scan-based diagnosis is used in performing physical failure analysis to uncover the defect mechanism and ultimately in determining the root cause of the defect(s) and taking corrective measures such as changes to the process and/or design.

This paper presents a case study of yield learning through advanced scan diagnosis to resolve a significant yield excursion for an IC containing 1 Million logic gates, manufactured at 130 nm technology node.

2. Layout-aware Advanced Scan Diagnosis

Scan based diagnosis is recently being applied to yield learning [1]-[6] in addition to the conventional fault isolation of one-of-a-kind failures [7]-[14]. For the purpose of yield learning, the scan diagnosis tool needs to have high performance such that statistically significant failure data can be processed. In addition, suspect ranking and defect type classification are essential. Finally, given the prevalent use of embedded compression hardware, yield learning based on production test failures requires that the diagnosis tool be capable of diagnosing the compressed pattern failures directly.

The diagnosis tool utilized in this study is suitable for both one-off analysis as well as yield learning. It is capable of diagnosing with compressed patterns and utilizes a location-based approach to identify the suspects, classify their defect type and assign a score for each suspect indicating how well the failures observed on the tester can be explained by the suspect behavior. The suspects are finally presented in a ranked order based on their score. The flow of this approach broadly consists of the three steps outline below. First, for each failing pattern, each single gate pin location with failure (the location has opposite logic value from good machine value) is simulated to see whether the failing pattern can propagate the failure to all observable points with same behavior as observed on the ATE. Second, if the scan diagnosis tool finds any correlation, the failing pattern is explained by that location. Third, after this step, a heuristic method finds a set of minimal faults that can explain all the failing patterns [15]-[17]. In addition, the diagnostic tool partitions the failure data into several independent *symptoms*. For each symptom, the suspects that explain the failing patterns in that symptom are listed in the decreasing order of their score along with their defect type classification.

Below details about the defect type classification performed by the diagnosis tool are presented.

STUCK – A stuck-at 0 or stuck-at 1
OPEN/DOM – This category is used for suspect nodes that fail both a 0 and 1 state during different patterns. This suspect could be caused by an open along a single net or a dominant bridge between two or more nets.
BRIDGE_2WAY – A bridge between two nets. A logic-high / logic-low on either net causes both signals to pull to a high / low value.
BRIDGE_3WAY – A bridge affecting 3 nets.
EQ# – The suspect is equivalent to another previously listed suspect.
INDETERMINATE – The suspect always fails as a low or a high state, but causes some patterns that passed on the tester to fail in simulation and therefore cannot be classified as stuck-at.

By default the defect type classification is based on logical analysis. However, layout information can be provided to the tool to improve the classification as well as resolution. For example, net pairs that have increased possibility of bridging can be extracted from the layout and supplied to the diagnosis tool. Figure 1 below illustrates the various features that are targeted with bridge extraction for this study. The four types of bridges: Side-to-side, Corner-to-corner, End-of-line, and Via-to-Via; were derived from manufacturing Design Rule Check (DRC) and Design-for-Manufacturing (DFM) rules and known bridge causing layout features.

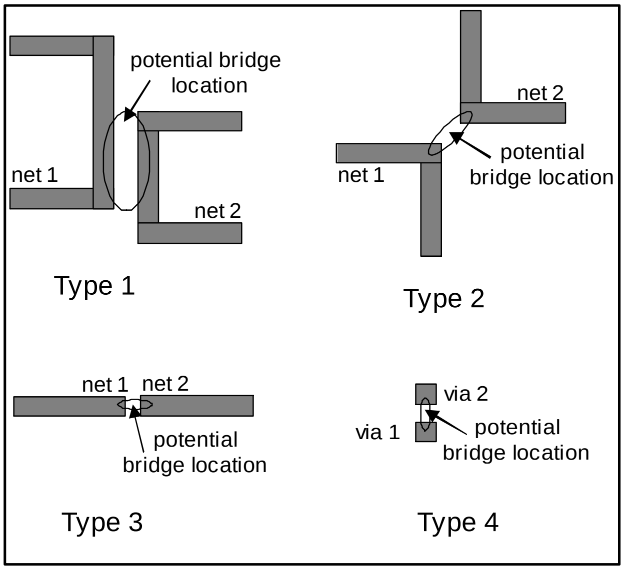

Figure 1 Features for bridge extraction

Opens can also be extracted in a similar way. For example, all single-via locations in signal lines can be extracted, since single vias are a key contributor to (resistive) open kind of defects.

3. Case Study Information

Table 1 below illustrates design and test information about the case study presented in this paper. The design is manufactured at 130 nm with Cu interconnect, has 1 million logic gates with 8 clock domains, and 297 embedded memories. As can be seen from the table, the design is mostly full scan. The test set is composed of single stuck-at test with 2399 patterns attaining high test coverage of 99.76%.

# Logic gates	1 Million
# Memories	297
# Clocks	8
# Scan cells	76K
# Scan chains	16
Technology	130 nm, Cu
# Stuck-at test patterns	2399
Stuck-at test coverage	99.76%

Table 1 Case study design and test information

Yield learning effort was initiated due to an excursion wherein in the yield was significantly less than has been seen until that point of time. Analyzing the tester failures revealed that most of the fallout was occurring for a particular stuck-at test pattern, indicating a systematic issue.

4. Yield Learning

This section explains the yield learning steps taken to identify the root cause of the excursion. These steps involved performing scan-based diagnosis, adaptive Automatic Test Pattern Generation (ATPG) to improve diagnosis resolution, and physical failure analysis to identify the defect mechanism and ultimately the root cause.

4.1 Scan-based Diagnosis

Using the failure information for the single failing test pattern, the design netlist, and the stuck-at pattern set, scan-based diagnosis was performed. Initial diagnosis did not consider the layout information and resulted in a total of 28 ranked suspects. The top 8 suspects of these are illustrated in Figure 2.

```
1       99     INDET.       0
/tc_inst_0/tc_phi_acc_mux_inst_0/U6687/A1 (AO22X1)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n3110)

2       99     EQ1          0
/tc_inst_0/tc_phi_acc_mux_inst_0/U6687/A0 (AO22X1)
(/tc_inst_0/tc_phi_acc_mux_inst_0/sl_kri_sync_fifo_hi_rdata
[43])

3       99     INDET.       0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10161/Y (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n8566)

4       99     EQ3          0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10161/A (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6960)

5       99     EQ3          0
/tc_inst_0/tc_phi_acc_mux_inst_0/U6687/Y (AO22X1)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6960)

6       99     EQ3          0
/tc_inst_0/tc_phi_acc_mux_inst_0/U4024/D (OR4X1)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n8566)

7       96     INDET.       0
/tc_inst_0/tc_phi_acc_mux_inst_0/U2206/Y (BUFX12)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n1538)

8       96     EQ7          0
/tc_inst_0/tc_phi_acc_mux_inst_0/U2206/A (BUFX12)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6962)
```

Figure 2 Top suspects from the initial diagnosis run

As can be seen from the figure, the initial diagnosis run did not result in good resolution or identification of the defect type (indeterminate type means that type of defect could not be narrowed down). This was primarily due to the lack of enough failure information, which essentially consisted of single failing stuck-at test pattern. The next section outlines adaptive ATPG that was undertaken to increase the amount of failure information and thus the diagnosis resolution.

4.2 Adaptive ATPG

Production test set is typically optimized to detect as many defective die as possible as fast as possible. This results in highly compacted test patterns, wherein detection of faults is more important than distinguishing them, ordered based on their effectiveness in terms of test coverage. Such test patterns are not necessarily optimal for diagnosis, where the objective is to distinguish the failures so that they can be isolated to few locations.

In the specific case under consideration, as noted above, a single stuck-at test pattern failed on the tester. Diagnosis with this failing test pattern resulted in many suspects with inconclusive defect type. Analysis of the top few suspects, through fault injection and layout analysis indicated that the defect is non-stuck type as some of the patterns passing on the tester were failing in simulation. Based on the layout of the top suspects a bridge defect was suspected and adaptive ATPG was used to create additional test patterns to detect the bridges.

The concept of adaptive ATPG, illustrated in Figure 3, is to augment the original test set by iteratively creating fine-tuned test patterns where certain fault models or areas of the die flagged by diagnosis with the previous test set are targeted during ATPG. The idea is that even if resolution needs improvement, diagnosis data can still give valuable information such as pointers to a specific part of the logic, or exclusion of simple defect types such as stuck–at. Subsequent iteration of pattern generation using multiple–detect and bridging–fault ATPG targeting the logic associated with the reported suspect candidates is a good start. Any additional patterns failing on the tester will help the diagnosis tool narrow down the list of suspect candidates and determine the defect types. With good results from diagnosis, effective failure analysis can be performed to locate the defect(s).

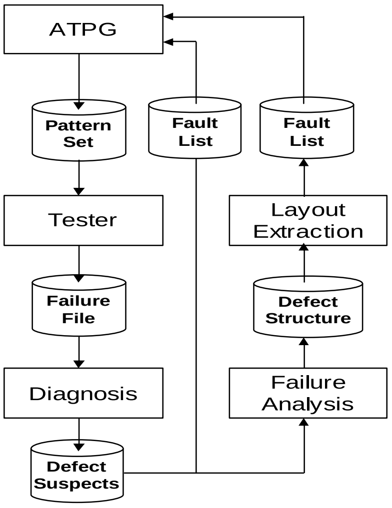

Figure 3 Adaptive ATPG

Adaptive ATPG can also be used to reduce test escapes. ATPG diagnosis and failure analysis results from defective devices can give clues about structures prone to systematic defects. If the failure analysis of defective devices shows systematic defects, it is likely that similar systematic defects may occur at other locations where the existing pattern set may not detect the defects. If for example a bridging type of defect is associated with a particular layout structure, bridge extraction rules can be created to extract bridge-pairs exhibiting similar layout structures throughout the device. Adaptive ATPG can target the extracted bridge-pairs. The updated pattern set may result in detection of additional defective devices exhibiting such a systematic defect type.

Adaptive ATPG was used to generate multiple-detect and targeted bridging patterns as illustrated in Table 2. Note that in multiple-detect ATPG, each fault is dropped from the fault list only after it is detected a specified number of times. Detecting a fault multiple times with varying excitation and propagation conditions statistically increases the possibility of detecting bridge faults around the targeted fault. The targeted bridging ATPG on the other hand explicitly targets specified bridge locations and creates patterns to detect them. Just like the conventional ATPG targeted bridge ATPG drops a fault from the list after it is detected once. As can be seen from Table 2, additional 12 and 7 4-detect stuck-at and transition patterns respectively were created. The stuck-at 4-detect ATPG targeted all the faults in the logic cone of the failing flops, while the transition ATPG targeted just the initial suspects.

For targeted bridge ATPG, layout analysis of the suspects was performed to identify the fault list. Based on this analysis, a pair of nets is determined to have elevated likelihood of bridging. Note that automatic extraction of bridges using the rules illustrated in Section 2, did not extract the pair of nets identified through analysis. This indicates that either the parameters used for extraction or the type of extraction rules did not cover suspected bridge site. As will be explained in more detail later, failure analysis revealed a new systematic issue that was not covered by the rules illustrated in Section 2.

#4-detect stuck-at test patterns	12
#4-detect transition patterns	7
#Targeted bridging patterns	2

Table 2 Adaptive ATPG results

The additional patterns created using adaptive ATPG were applied to the failing dice and 13 of the 19 4-detect patterns and both targeted bridging patterns caused failures on the tester. Diagnosis with these additional failures resulted in the type of the defect being classified as 2-way bridging, though there were 8 suspects with a score of the 100. The top 5 suspects are shown Figure 4.

```
1 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10161/A (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6960)

1 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10160/A (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6918)

2 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10161/A (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6960)

2 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10160/Y (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n8565)

3 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10161/Y (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n8566)

3 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10160/A (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6918)

4 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10161/Y (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n8566)

4 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U10160/Y (BUFX8)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n8565)

5 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U2206/A (BUFX12)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6962)

5 100 BRIDGE_2WAY 0
/tc_inst_0/tc_phi_acc_mux_inst_0/U2208/A (BUFX12)
(/tc_inst_0/tc_phi_acc_mux_inst_0/n6920)
```

Figure 4 Top suspects from diagnosis run with adaptive patterns

4.3 Failure Analysis

Layout analysis of the top few suspects revealed that the net pairs listed in suspect 4 have physical proximity and higher likelihood of bridging. Figure 5 below depicts the layout view of the suspected bridge defect site for nets listed in suspect 4. The interconnect segments belonging to the net pair, highlighted with an oval in the figure, are routed in metal layer 5 and are separated by distance d2. These segments are surrounded by thick metal stacks, separated by distance d1, in the lower metal layers 1 through 4.

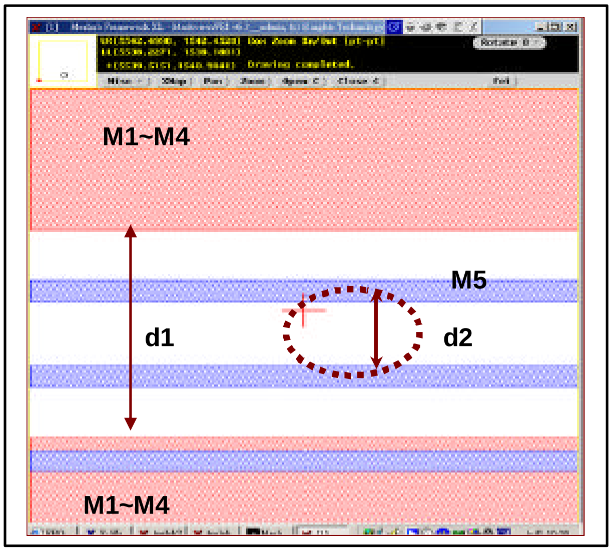

Figure 5 Layout view of suspected bridge defect site

Physical failure analysis uncovered a bridge defect between the interconnect segments shown in Figure 5. This defect was systematic and was affecting a large number of die. The root cause was determined to be the metal residue left on metal layer 5, due to the severe non-planar surface caused by dishing during chemical mechanical polishing (CMP) of the underlying four layers of metal. Dishing in the lower layers elevated the possibility of bridging even though the metal lines in layer 5 are separated by more than the recommended minimum distance.

4.4 Corrective Measures

In order to remove the metal residue due to dishing in lower layers, process was changed to perform extra polishing. In addition, a new bridge extraction rule, illustrated in Figure 6, was added the deck of bridge extraction rules. Note that the original rules did not extract the net pair exhibiting the defect even though they run in parallel. This is because they are separated by a distance much higher than what was expected to cause elevated bridge defect possibility due to parallel run. However, in this specific systematic issue even though the nets are farther, they had increased bridge defect possibility due to dishing caused by CMP of wide metal stacks in the lower layers.

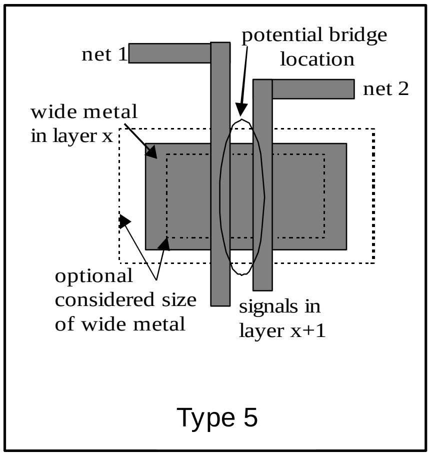

Figure 6 New bridge extraction feature

With extra polishing, yield was again restored to anticipated levels. In addition, as mentioned before, quality of the outgoing parts can also be improved by adding test patterns detecting the sites identified by the new bridge extraction rule.

5. Conclusions

In this paper we presented a case study of yield learning for a 130nm device with 1 Million logic gates. Significant fallout was observed for a particular test pattern in the production stuck-at test pattern set. Adaptive ATPG, layout-aware scan diagnosis, and layout analysis were used to narrow down the suspected defect site. Physical failure analysis revealed a systematic bridge defect caused by metal residue left over because of dishing caused by CMP of underlying metal layers. Increasing the polishing resulted in restoring the yield back to the anticipated levels. Furthermore a new rule was added to the deck of bridge extraction rules such that, tests can be targeted to increase the quality of the outgoing parts, and diagnosis has the extra information to isolate the defects.

6. References

[1] W. Maly, et. al., "Deformation of IC Structure in Test and Yield learning," Proc. Intl. Test Conf. 2003, pp.856-865.

[2] J.B. Khare, W. Maly, S. Griep, D. Schmitt-Landsiedel, "Yield-oriented computer-aided defect diagnosis," IEEE Trans. On Semi. Manufacturing, May 1995, Volume 8, Issue 2, pp. 195-206.

[3] C. Hora, R Segers, S. Eichenberger, and M. Lousberg, "An Effective Diagnosis Method to Support Yield Improvement," Proc. Int'l Test Conf. 2002.

[4] A. Leininger, P. Muhmenthaler, W.-T. Cheng, N. Tamarapalli, W. Yang, H. Tsai, "Compression Mode

Diagnosis Enables High Volume Monitoring Diagnosis Flow”, ITC 2005, Paper 7.3

[5] D. Appello et al., “Understanding Yield Losses in Logic Circuits”, IEEE Design and Test of Computers, Vol 21, No. 3, pp. 208-215, May-June 2004

[6] C. Eddleman, N. Tamarapalli, and W.-T. Cheng, “Advanced Scan Diagnosis based Fault Isolation and Defect Identification for Yield Learning”, Proc. Of ISTFA 2005

[7] R. Desineni and R.D. Blanton, “Diagnosis of Arbitrary Defects using Neighborhood Function Extraction”, Proc. Of VLSI Test Symposium 2005

[8] R. D. Blanton, “Failure Diagnosis using Fault Tuples”, in Proc. Of IEEE Latin American Test Workshop, pp. 253-257, Feb 2001

[9] J. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured Circuts,” IEEE Design and Test of Comp., vol. 6, no, 4, 1989, pp. 49-60.

[10] T. Bartenstein, D. Heaberlin, L.Huisman and D. Sliwinski, "Diagnosing Combinational Login Designs using the Single Location At-A-Time(SLAT) Paradigm,” in Proc. Intl. Test Conf., 2001, pp. 287-296.

[11] W.-T. Cheng, K.-H. Tsai, Y. Huang, N. Tamarapalli, and J. Rajski, “Compactor independent direct diagnosis,” Proc. of Asian Test Symp., pp. 15-17, 2004.

[12] D. Bodoh, A. Blakely, and T. Garyet, “Diagnostic Fault Simulation for the Failure Analysis,” Proc. Int’l Symp. for Test and Failure Analysis 2004.

[13] H. Balachandran, J. Parker, D. Shupp, K. Butler, C. Force, J. Smith, “Correlation of logical failures to a suspect process step,” Proc. Intl. Test Conf., 1999, pp. 458-466.

[14] A. Kinra, H. Balachandran, R. Thomas and J. Carulli, “Logic mapping on a microprocessor,” Proc. Intl. Test Conf., 2000, pp. 701-710.

[15] J. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured Circuts,” IEEE Design and Test of Comp., vol. 6, no, 4, 1989, pp. 49-60.

[16] T. Bartenstein, D. Heaberlin, L.Huisman and D. Sliwinski, "Diagnosing Combinational Login Designs using the Single Location At-A-Time(SLAT) Paradigm,” Proc. Intl. Test Conf., 2001, pp. 287-296.

[17] D. B. Lavo, I. Hartanto, T.Larrabe, "Multiplets, Models, and the Search for Meaning: Improving Per-Test Fault Diagnosis,” Proc. Intl Test Conf., 2002, pp.250-259.