Physically-Aware Analysis of Systematic Defects in Integrated Circuits

Wing Chiu Tam, ECE Dept., Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213;
Email: wtam@andrew.cmu.edu; Advisor: Prof. Shawn Blanton; ACM member no.: 5412827; Category: Graduate

I. PROBLEM AND MOTIVATION

The relentless scaling of CMOS technologies tremendously decreases device sizes and increases complexity of the resulting circuits. As more and more devices are placed in the decreasing silicon real estate, our ability to observe and understand defects is rapidly diminishing. IC test, traditionally used to screen for failures for quality control purposes, now serves an added function of understanding and monitoring defects in the process. By diagnosing and mining the test fail data, valuable insight can be obtained concerning the reasons that chips fail. After the defects are identified and their failure mechanisms are understood, actions can be taken to prevent their future occurrence to improve yield. In addition, specific tests can be generated to screen for these defects to improve product quality. **My research addresses the problem of information extraction from volume diagnosis data in order to provide feedback to the designers for product yield improvement.** More specifically, this work tackles issues in this area by performing analysis of IC diagnosis data in the context of the design layout. The methodologies developed include evaluating the effectiveness of DFM (design-for-manufacturability) rules [1] in systematic-defect prevention, identifying and understanding systematic defects [2] to guide process improvement effort, formulate new DFM rules [3] or identify targets for test [4, 5], and creating a framework for fast and accurate defect simulation [6] to validate these and other emerging test/yield learning methodologies (*e.g.*, [7]).

II. DFM RULE EVALUATION

In this section, the methodology for DFM rule evaluation is described.

A. Background and Related Work

DFM (design for manufacturability) refers to a wide spectrum of activities that are performed on the design prior to manufacturing in order to improve yield. DFM is necessary in nano-scale technology because of the formidable challenges encountered in manufacturing. To have acceptable yield, potential manufacturing difficulties must be anticipated and then remedied using appropriate design compensations. One approach is to remove/reduce the offending layout features that are susceptible to failures. To achieve this goal, these features must be communicated to the designers. The common approach is to use DFM rules. DFM rules are very similar to regular design rules, except that many of them are often not strictly enforced. This is because the cost-benefit trade-off for adhering/violating a particular rule is not well understood [8]. This prevents designers from reaping the full benefit of DFM. Prior work [4, 9-11] in this area focuses on identifying or testing for systematic issues, our work focuses on understanding the economic benefit of applying a particular set of DFM rules and ranking them according to their relative contribution to yield enhancement.

B. Approach and Uniqueness

To achieve this goal, a methodology (called RADAR --- Rule Assessment of Defect-Affected Regions) has been developed to evaluate DFM rules using diagnosis-implicated regions of failed ICs [1]. Fig. 1 shows the flow diagram of the RADAR methodology. The flow starts with diagnosing a large number of IC failures. This diagnosis-driven approach is substantially different from traditional approach in which test structures are used to evaluate DFM rules [12, 13]. The outcome of diagnosis is a set of suspect signal lines or cells where the defect is believed to reside. The corresponding area in the layout is examined to collect measures of effectiveness (*i.e.*, the number of times a rule is violated/adhered) for each DFM rule of interest. In addition, for control purposes, the nets for the rest of the circuit, where failure does not occur, are sampled and examined in the same way. Using the information collected, statistical analysis is then performed to determine: (1) if there is strong evidence to indicate that the violation of a DFM rule is associated with the observed failures, (2) which rule is more important in terms of their ability to prevent failures (*e.g.*, if violations of rule A cause more failures than violations of rule B, then rule A is more important than rule B), and (3) the amount of yield loss that would be recovered if a certain percentage of violations of a specific rule is corrected.

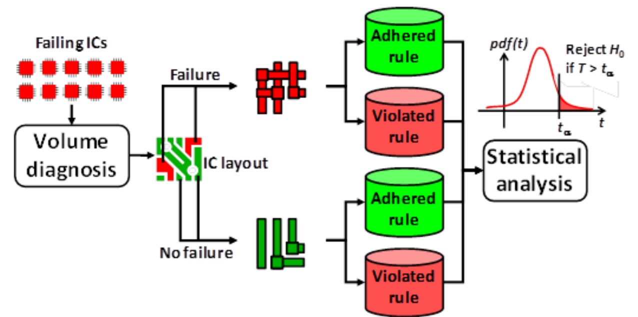

Fig. 1: Overview of the RADAR methodology.

To detect association of rule violations and yield loss, hypothesis testing is used. This technique is commonly used in the medical communities to determine if certain chemicals/activities have an effect on the health of an individual (*e.g.*, is smoking associated with lung cancer?). To determine the relative importance of a rule, the relative risk is estimated for each rule. In the context of RADAR, relative risk is simply the ratio of the failure rate given rule violation to the failure rate given rule adherence. From the definition, it is clear that relative risk measures the increase in likelihood of

failure when a rule is violated. Thus, calculating the relative risks can reveal the relative importance of each rule. Additionally, the failure rate given rule violation is also estimated for all the DFM rules of interest using an EM (expectation-maximization) approach. After the relative risk and failure rate given rule violation are estimated, the failure rate given rule adherence can be estimated as well since relative risk is simply a ratio of the two failure rates. Having the estimates of these two failure rates, in turn, allows the estimation of yield loss before and after addressing a certain amount of rule violations. The difference measures the resulting yield impact of addressing the violation and thus can also be used as a measure to indicate importance of a rule in a given set of rules.

C. *Results and Contributions*

RADAR helps the designers better understand the cost-benefit trade-off and allows them to adjust future designs to improve yield at the same technology node. Table 1 shows the result of applying RADAR to a commercial Nvidia GPU. For this experiment, via-enclosure rules (*e.g.*, M8.ENC.1R, M7.ENC.2R) and metal-density rules (*e.g.*, M23.WD.4R) are evaluated. The rule names are shown in column 1. Hypothesis testing using Pearson's χ^2 test (columns 2) shows that a majority of the DFM rules is associated with yield loss. (A green entry in column 2 denotes association while a red entry denotes independence). The associated *p*-values (*i.e.*, the probability of observing more extreme data than the current one) are shown in column 3. Both relative risk and yield impact are then calculated for each rule, which are shown in columns 4 and 5, respectively. The rules have been sorted in decreasing order of relative risk. The order shows that contact enclosure rules are more effective than density rules. This suggests that via issues may cause more yield loss than planarity/density problems for this particular process/design. It is also clear from columns 4 and 5 that the ordering of yield impact for a given set of rules is not the same as the ordering of relative risk. This is expected because yield impact takes into account the amount of rule violations specific to a design.

TABLE 1: SUMMARY OF DFM RULE EVALUATION RESULTS FOR AN NVIDIA GPU DESIGN.

Rule name	Pearson's χ^2 statistics	*P*-value	Relative risk	Yield impact
M8.ENC.2R	55.24	0.0000	117.69	0.009316
M7.ENC.2R	547.47	0.0000	69.95	0.010739
M2.ENC.2R	3264.86	0.0000	31.95	0.016067
M4.ENC.2R	983.60	0.0000	28.90	0.009543
M2.ENC.1R	4169.01	0.0000	26.34	0.000159
M5.ENC.2R	799.76	0.0000	24.18	0.012041
M6.ENC.2R	682.20	0.0000	24.06	0.012955
M3.ENC.1R	2812.44	0.0000	22.53	0.001236
M23.WD.4R	57.28	0.0000	8.19	0.000154
M67.WD.4R	0.00	0.9740	0.72	0.000000
M78.WD.4R	0.20	0.6542	0.47	0.000000

III. SYSTEMATIC DEFECT IDENTIFICATION

In this section, the methodology for systematic defect identification is described.

A. *Background and Related Work*

Defects inevitably occur in a semiconductor manufacturing process and can be classified into two categories, namely random and systematic defects. The former arise because of random contamination particles that affect the structure of a circuit, thus altering its functionality. In contrast, systematic defects happen because of design-process interactions, *i.e.*, part of the design is sensitive to manufacturing in certain process corners and has an increased likelihood of failure. Unlike random defects, systematic defects can happen repeatedly in places with similar layout geometries. Identification of systematic defects immediately leads to yield improvement. Since systematic defects are becoming a significant source of failures in nano-scale technologies, the need to identify them becomes more important than ever.

In the last several years, there has been a growing amount of work on systematic defect identification. In [10, 14, 15], expected failure rates are computed for layout features/nets. This information is combined with volume diagnosis data to identify outliers (*i.e.,* systematic defects). Work in [16] applies clustering techniques to test-response data collected from a large number of failing ICs to identify common failure signatures. Lastly, the work described in [17, 18] performs extensive lithography simulation on the entire layout to identify hotspots. Layout snippets containing these hotspots are extracted and clustered to form DFM rules.

The aforementioned work has the goal of identifying systematic defects, and undeniably, they have a certain degree of success, as evidenced by experiment results. Unfortunately, past approaches [10, 14-16] fall short of automatically extracting similar layout features that may underlie many of the failures. The work in [17, 18], while capable of automatic layout feature extraction, relies entirely on simulation, which is known to be inaccurate and too conservative. Relying on simulation alone also means that no systematic defects are identified for failing mechanisms that are not modeled.

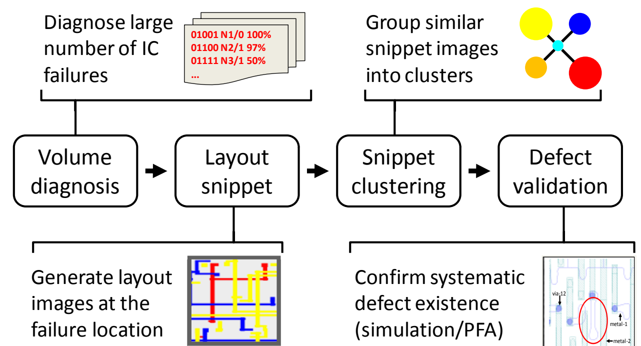

Fig. 2: Overview of the LASIC methodology.

B. *Approach and Uniqueness*

To identify systematic defects, a diagnosis-driven methodology (called LASIC --- Layout Analysis for Systematic IC-defect identification using Clustering) is developed [2]. The LASIC flow is illustrated in Fig. 2. LASIC uses the following four steps: (1) diagnosis, (2) layout-snippet extraction, (3) snippet clustering, and (4) validation. Similar to

the RADAR flow, LASIC starts by applying diagnosis to a large sample of chips, or is continually applied to failures. The outcome of diagnosis is a set of suspect signal lines or cells where the defect is believed to reside. Snippet images are then extracted from the corresponding layout region for each suspect. Clustering, such as *K*-means, is then applied to group similar snippet images together for identifying any commonalities. (To speed up the clustering process, the snippet images are represented using discrete cosine transform (DCT) so that only the dominant DCT coefficients are used in the clustering process.) Layout features identified by LASIC can then be analyzed using a process simulator (such as the lithography tool Optissimo [19]) to confirm the existence of a systematic defect, and/or physically analyzed using Physical Failure Analysis (PFA).

In contrast to [10, 14-16], LASIC identifies problematic layout features automatically. In addition, since the methodology analyzes real IC fails, it does not suffer from the false negatives that plague the approaches described in [17, 18].

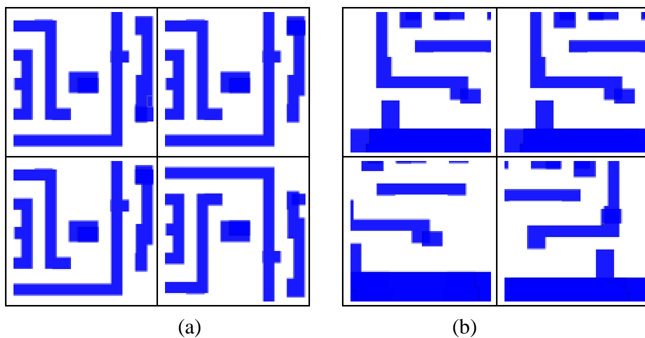

Fig. 3: Illustration of clustered snippet images (a) from one cluster and (b) a second cluster.

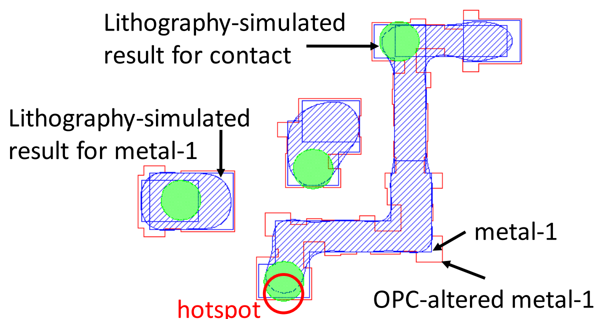

Fig. 4: Illustration of a hotspot identified using lithography simulation.

C. *Results and Contributions*

The methodology has been implemented and tested on two industrial data. Specifically, two commercial designs, namely a LSI testchip in 130nm technology and an Nvidia graphic processing unit (GPU) in 90nm technology, with real failure data are used to validate the methodology. Fig. 3 shows example images from two different metal-1 clusters after LASIC has been applied. Fig. 3a shows four snippet images from a cluster in metal-1, while Fig. 3b shows four snippet images from another metal-1 cluster. Fig. 3 shows that geometries in the same cluster resemble each other but are not exactly the same while geometries in different clusters exhibit substantial differences. This example clearly demonstrates that LASIC is able to extract from diagnosis data images that have similar layout features.

The layout snippets that correspond to the snippet images are fed to Optissimo for lithography simulation. Fig. 4 shows an example of a lithography-induced hot spot (*i.e*., insufficient contact coverage) identified by Optissimo. There are altogether 20 hotspots identified in the metal-1 layer. If these hotspots are confirmed to cause significant yield loss, then DFM rules can be formulated to prevent their future occurrence. For example, the insufficient coverage of the contact in Fig. 4 can be resolved by increasing the metal-1-to-contact enclosure requirement.

IV. Mixed-Signal Defect Simulation

In this section, the methodology for mixed-signal defect simulation is described.

A. *Background and Related Work*

All test-fail driven yield/test learning methodologies should be validated to ensure their correctness. Unfortunately, this is easier said than done because of unavailability of test fail data and the corresponding PFA data, which are often foundry confidential. To circumvent this difficulty, the concept of virtual failure data creation was proposed in [20]. The idea is to simulate failures by injecting faults/defects in the circuit. The virtual data generated can then be used to evaluate the effectiveness and accuracy of the yield/test learning methodology under investigation. This approach also has the advantage that the "correct" answer (*i.e.,* the injected defect) is known and can be precisely controlled and thus it enables debugging and refinement of the methodology.

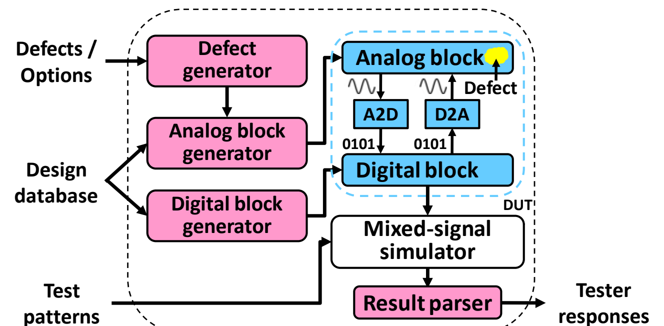

Fig. 5: Block diagram of the SLIDER framework.

B. *Approach and Uniqueness*

While previous work (*e.g.,* [20-23]) has considered this topic before, the innovation in my work here centers on three aspects: (i) accuracy resulting from defect injection taking place at the layout level, (ii) speedup resulting from careful and automatic partitioning of the circuit into digital and analog domains for mixed-signal simulation, and (iii) complete automation that involves defect injection, design partitioning, netlist extraction, mixed-signal simulation, and test-data extraction [6]. The defect simulation framework developed is called SLIDER (Simulation of Layout-Injected Defects for Electrical Responses).

Fig. 5 shows the key components of SLIDER. It is composed of an analog block generator (ABG), a digital block generator (DBG), a mixed-signal simulator (MSS), and a result parser (RP). The blocks in pink are implemented in SLIDER. The MSS used in SLIDER is Cadence AMS designer [24, 25]; however any other mixed-signal simulator can be easily used instead.

The inputs to the framework are the design database, a description of the defects to be injected into the layout or a set of options to direct the defect generation process, and test patterns. The design database must contain both the layout and the logic-level netlist. The defect description specifies the geometry of one or more defects (*i.e.*, coordinates, size, polygon, layer, *etc.*). If the defect description is not specified, then options must be specified to direct the defect generation process. The supported defect types include: open, resistive via, signal bridge, supply bridge, and various kinds of cell defects that include the “nasty polysilicon defect” [26], transistor stuck-open, and transistor stuck-closed. The test patterns mimic the tester stimulus that would be applied to the circuit during production testing.

Using these inputs, SLIDER starts by automatically partitioning the design into its respective analog and digital blocks, which is accomplished by ABG and DBG, respectively. The analog block contains all the cell instances and nets that are required to produce accurate circuit-level simulation of defect behavior, while the digital block contains everything that remains. Defects are then injected into the analog block at the layout level. The modified layout is then extracted to produce the circuit-level netlist of the analog block. The analog block, the digital block, and their connections form the design under test (DUT), as illustrated in blue in Fig. 5. A transient analysis is then performed on the DUT by the MSS using the test patterns as inputs. To capture the DUT’s response, RP samples the output waveform at the frequency defined by the test clock. During simulation, the MSS automatically takes care of the cross-domain signals through the use of connection modules [25], an analog-to-digital (digital-to-analog) convertor for a signal that crosses from the analog (digital) domain to the digital (analog) domain.

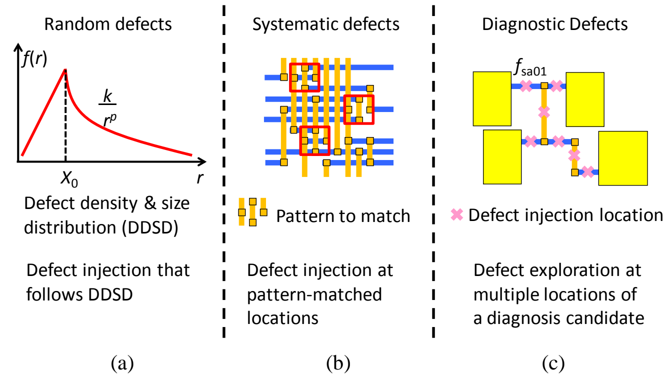

Fig. 6: Three defect generation modes are supported in SLIDER that include: (a) random, (b) systematic, and (c) diagnostic.

Besides allowing the users to directly specify the defects directly, SLIDER supports several defect-generation modes, as shown in Fig. 6. SLIDER can generate defects that mimic random contaminations that follow a defect density and size distribution (DDSD) (Fig. 6a). SLIDER can also generate systematic defects that share a common layout pattern (Fig. 6b). In this mode, SLIDER provides a layout pattern (which can be user-defined or automatically generated) to a commercial pattern matching tool [27] to find all locations that mimic the given pattern. The identified locations then serve as defect injection sites as illustrated in Fig. 6b. Finally, Fig. 6c illustrates the third mode of SLIDER. This mode allows the user to explore different defect-injection scenarios at a given location for improving diagnosis.

C. Results and Contributions

Because of the scalability enhancement, SLIDER can be applied in other settings that include diagnosis resolution improvement, defect localization, and fault model evaluation. Three set of experiments are performed to demonstrate scalability and usage of SLIDER.

First, a run-time experiment is performed. Specifically, eleven benchmark circuits are placed-and-routed using Cadence Encounter. For each circuit, 100 random defects are layout-injected and simulated. The results are averaged and summarized in Fig. 7. Fig. 7a compares the total number of gates and the number of gates in the analog block for each circuit. It is clear that the size of the analog block does not scale, even when the circuit size increases tremendously. This property provides the basis for the scalability of SLIDER, *i.e.*, the amount of effort needed for circuit-level simulation remains relatively constant. Fig. 7b shows the total runtime normalized by the pattern count. It is clear that the normalized runtime is in the order of seconds for all the benchmark circuits. Lastly, Fig. 7c shows that SLIDER adds a negligible overhead to the total runtime, which is dominated by the simulation time.

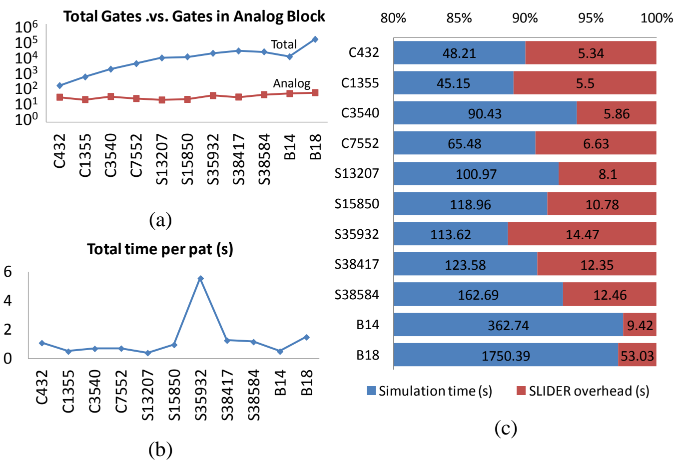

Fig. 7: Runtime results that demonstrate the scalability of the SLIDER framework.

In the second experiment, the accuracy of LASIC is evaluated using the virtual failure data generated by SLIDER. Specifically, two failure populations, each of size 100 are generated for both random and systematic defects, using DDSD and systematic defect generation modes of SLIDER,

respectively. The failures are then randomly sampled from both populations to form a single population. This process is repeated to get failure populations with different proportions of random and systematic defects. The goal is to evaluate how much random "noise" can be tolerated by LASIC. The experiment is performed using eight ISCAS benchmarks. Four types of defects are injected into these benchmarks: metal-2 bridge, metal-2 open, metal-3 bridge, and metal-3 open. The result is averaged over the eight benchmarks. In addition, to isolate the "noise" that occurs due to inaccuracies/ambiguities in diagnosis, this experiment is performed for two scenarios: ideal and actual diagnosis. Ideal diagnosis means that diagnosis correctly pinpoints the faulty line with no ambiguity while actual diagnosis can be inaccurate (*i.e.*, omitting the defect completely) or low in resolution (*i.e.*, additional suspects are reported along with the actual defect location). Comparing the two scenarios provides insight about how LASIC performs under various scenarios. After applying LASIC to the virtual data, clusters of layout images of the suspect net are generated. The cluster that contains the underlying pattern of the injected systematic defects is called the *correct cluster*. For each scenario, the averaged rank of the correct cluster is plotted against the percentage of systematic defects injected. The data is summarized in Fig. 8 for both ideal and actual diagnoses.

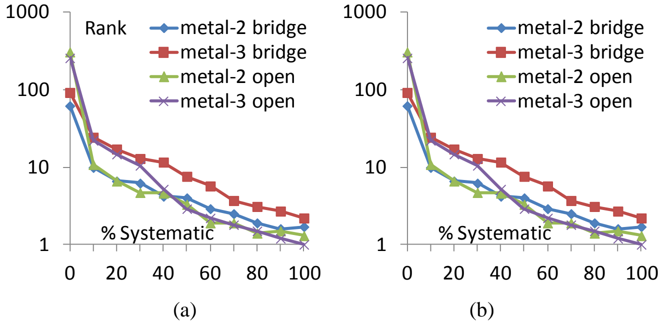

Fig. 8: LASIC evaluation for (a) ideal and (b) real diagnoses.

It is clear from Fig. 8 that the average rank of the correct cluster improves (a smaller numeric value of the rank indicates that the cluster is ranked higher) as the percentage of systematic defects increases for both ideal and actual diagnoses. This result is expected because the size of the correct cluster should increase as the number of systematic defects present increases. It is also evident that the technique is effective in both ideal and actual diagnoses because the correct cluster is present in the top 40 ranked clusters (top 3%) even when the population only consists of 20% systematic defects. As expected, performance of LASIC is worse for actual diagnosis due to inaccuracies/ambiguities in diagnosis. More experiments involving multiple systematic sources of defects and random defects are also performed. It is found that LASIC is able to identify over 90% of systematic issues in the top 55 clusters. The interested reader is referred to [28] for more details.

In the third experiment, the accuracy of RADAR is evaluated using the SLIDER-generated virtual failure data. A set of eight virtual DFM rules is handcrafted. In addition, for the evaluation of each DFM rule, two patterns are defined, namely the adherence pattern and the violation pattern. The adherence pattern defines the scenario when the rule is adhered, while the violation pattern defines the scenario when the rule is violated (*i.e.*, the rule itself). Both patterns have to be defined for virtual failures to be created in both scenarios. Similar to the LASIC experiment, two failure populations, each of size 100, are generated for both adherence and violation patterns for each rule, using the systematic-defect generation mode of SLIDER. Again, the failures are then randomly sampled from both populations to form a single population. This process is repeated to generate failure populations with a different proportion of adherence and violation patterns. The goal is to evaluate how the statistic used in hypothesis testing and the relative risk change as the proportion of violated-rule defects varies. Seven ISCAS designs are used in the evaluation. The results are averaged over the designs.

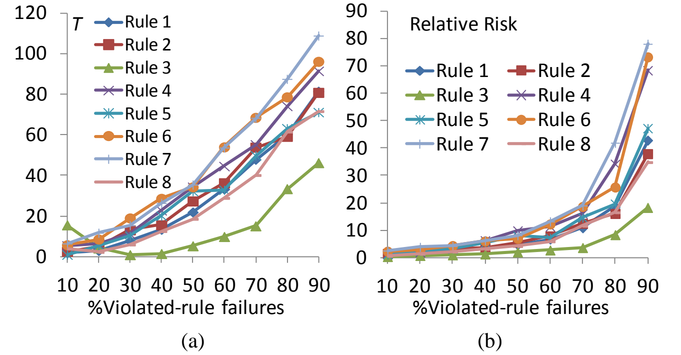

Fig. 9: Hypothesis testing evaluation showing the trend for each DFM rule in terms of (a) statistic T in Pearson χ^2 test and (b) the relative risk when the amount of violated-rule failures varies.

To examine how the effectiveness of RADAR changes as the amount of violated-rule failures varies, the statistic T (for Pearson χ^2 test) and the relative risk are plotted against the percentage of violated-rule failures in the failure population, as shown in Fig. 9a and 9b, respectively. Fig. 9a clearly shows that, as the percentage of violated-rule failures increases, the statistic T increases, indicating the increasing association of rule violations and failures, as expected. The associated p-value (not shown in Fig. 9) also decreases, indicating the increasing confidence that the association is valid. Fig. 9b shows that the relative risk increases as the amount of violated-rule failure increases, which indicates that the failure rate given violation is increasingly higher than the failure rate given adherence, as expected. More comprehensive experiments have been performed, which show that the yield impact estimation of RADAR has an accuracy of 73.56%. The interested reader is referred to [29] for more details.

As a side note, the initial results were different from what are shown Fig. 8 and Fig. 9 when LASIC and RADAR were first evaluated. Bugs were discovered and both techniques were improved. This would be difficult without the use of the virtual failure data generated by SLIDER.

References

[1] W. C. Tam and R. D. Blanton, "To DFM or Not To DFM?," *Design Automation Conference*, 2011.

[2] W. C. Tam, O. Poku, and R. D. Blanton, "Systematic Defect Identification through Layout Snippet Clustering," *International Test Conference*, p. 13.2, 2010.

[3] W. C. Tam, O. Poku, and R. D. Blanton, "Automatic DFM Rule Discovery through Layout Snippet Clustering," *International Workshop on Design for Manufacturability and Yield*, 2010.

[4] K. Dongok, *et al.*, "Testing for Systematic Defects Based on DFM Guidelines," *International Test Conference*, pp. 1-10, 2007.

[5] X. Yu, *et al.*, "Controlling DPPM through Volume Diagnosis," *IEEE VLSI Test Symposium*, pp. 134-139, 2009.

[6] W. C. Tam and R. D. Blanton, "SLIDER: A Fast and Accurate Defect Simulation Framework," *IEEE VLSI Test Symposium*, pp. 172 - 177, 2011.

[7] Y.-T. Lin, *et al.*, "Physically-Aware N-Detect Test Pattern Selection," *Design, Automation and Test in Europe*, pp. 634-639, 2008.

[8] M. Nowak and R. Radojcic, "Are There Economic Benefits in DFM?," *Design Automation Conference*, pp. 767-768, 2005.

[9] R. Turakhia, *et al.*, "Bridging DFM Analysis and Volume Diagnostics for Yield Learning - A Case Study," *IEEE VLSI Test Symposium*, pp. 167-172, 2009.

[10] J. Jahangiri and D. Abercrombie, "Value-Added Defect Testing Techniques," *IEEE Design & Test of Computers,* vol. 22, no. 3, pp. 224-231, 2005.

[11] K. Dongok, *et al.*, "Defect Diagnosis Based on DFM Guidelines," *IEEE VLSI Test Symposium*, pp. 206-211, 2010.

[12] A. A. Kagalwalla, *et al.*, "Design-of-Experiments Based Design Rule Optimization," *Prof. SPIE,* vol. 8327, pp. 832709-8, 2012.

[13] Y. Zhang, *et al.*, "32nm Design Rule and Process Exploration Flow," *Proc. SPIE,* vol. 7122, pp. 71223Z-12, 2008.

[14] K. Martin, *et al.*, "A Rapid Yield Learning Flow Based on Production Integrated Layout-Aware Diagnosis," *International Test Conference*, pp. 1-10, 2006.

[15] M. Sharma, *et al.*, "Efficiently Performing Yield Enhancements by Identifying Dominant Physical Root Cause from Test Fail Data," *International Test Conference*, pp. 1-9, 2008.

[16] L. M. Huisman, M. Kassab, and L. Pastel, "Data Mining Integrated Circuit Fails with Fail Commonalities," *International Test Conference*, pp. 661-668, 2004.

[17] J. Ghan, *et al.*, "Clustering and Pattern Matching for an Automatic Hotspot Classification and Detection System," *Proc. SPIE,* vol. 7275, pp. 727516-11, 2009.

[18] N. Ma, *et al.*, "Automatic Hotspot Classification Using Pattern-Based Clustering," *Proc. SPIE,* vol. 6925, pp. 692505-10, 2008.

[19] "The Optissimo User Manual," 6.0 ed. San Jose, CA: PDF Solutions Inc., 2001.

[20] T. Vogels, *et al.*, "Benchmarking Diagnosis Algorithms with a Diverse Set of IC Deformations," *International Test Conference*, pp. 508-517, 2004.

[21] A. Jee and F. J. Ferguson, "Carafe: an Inductive Fault Analysis Tool for CMOS VLSI Circuits," *IEEE VLSI Test Symposium*, pp. 92-98, 1993.

[22] W. Meyer and R. Camposano, "Fast Hierarchical Multi-Level Fault Simulation of Sequential Circuits with Switch-Level Accuracy," *Design Automation Conference*, pp. 515-519, 1993.

[23] W. C. Tam, O. Poku, and R. D. Blanton, "Automated Failure Population Creation for Validating Integrated Circuit Diagnosis Methods," *Design Automation Conference*, pp. 708-713, 2009.

[24] "The Virtuoso AMS Designer Environment User Guide," ed. San Jose, CA: Cadence Design Systems Inc., 2009.

[25] K. Kundert and O. Zinke, The Designer's Guide to Verilog-AMS, 1 ed.: Springer, 2004.

[26] R. D. Blanton, *et al.*, "Fault Tuples in Diagnosis of Deep-Submicron Circuits," *International Test Conference*, pp. 233-241, 2002.

[27] "The Calibre Pattern Matching User Manual," 1.0 ed. Wilsonville, OR.: Mentor Graphics Corporation, 2010.

[28] W. C. Tam and R. D. Blanton, "Design-for-Manufacturability Assessment for Integrated Circuits using RADAR," *submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2012.

[29] W. C. Tam and R. D. Blanton, "LASIC: Layout Analysis for Systematic IC-Defect Identification using Clustering," *submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2012.