Benchmarking Diagnosis Algorithms With a Diverse Set of IC Deformations

T. Vogels, T. Zanon, R. Desineni, R. D. Blanton, W. Maly,
J. G. Brown, J. E. Nelson, Y. Fei, X. Huang, P. Gopalakrishnan, M. Mishra, V. Rovner, and S. Tiwary

Dept. of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
`blanton@ece.cmu.edu`

Abstract

Diagnosis algorithms for integrated circuits (ICs) are typically developed and evaluated using a limited number of logic-level models of defect behaviors. However, it is well-known that real IC defects exhibit behavior well outside these models. Consequently, the utility of IC diagnosis methodologies may be uncertain. In this paper, a simulation-based benchmarking strategy is developed that uses circuit-level models to describe the complex nature of real defects. Specifically, we have proposed a simple yet powerful strategy using a small circuit and a set of bounded deformations (i.e., defects) for measuring the effectiveness of diagnosis techniques. Evaluation of several simple and commercial diagnosis algorithms indicates that this form of diagnosis benchmarking is viable.

1. Introduction

In this first section, we motivate the need for benchmarking to improve the increasing need for effective diagnosis.

1.1. Motivation

Among the many consequences of rapidly introducing new IC technologies [1] is the increased emphasis on achieving high levels of manufacturability (*e.g.*, [2]). This necessity, combined with the significant complexity of advanced IC products, elevates the task of yield learning to the top of the IC industry agenda. As a result, the test domain is not only asked: *"Does the device under test (DUT) meet the required specs?"* but also: "W*hy does the DUT fail the given test set?*" Of course, both questions are not new to the test community. What seems to be new, however, is the emerging importance of the second question.

So far, reasons for yield loss have been investigated with minimal involvement from the test domain. Utilization of test results has been limited to flagging defective parts and, in some cases, to indicate possible locations of the defect. Defect scanners and physical failure analysis (PFA) labs have carried out the bulk of the yield learning effort.

In the era of a billion transistors on a single chip, multi-layer interconnect obstructing visual inspection, and many other PFA obstacles, it is virtually impossible to approach yield learning without better utilizing information available from test. Hence, the question: *"W*hy does the DUT fail the given test set?*"* implies the growing importance of diagnosis in the deep sub-micron era testing.

1.2. Traditional IC diagnosis

The majority of traditional test-based diagnosis tasks, also referred to as fault diagnosis or logic-level diagnosis in the literature, have focused on identifying the location of the defect causing the detected IC malfunction. A large number of theoretical and practical approaches dealing with the localization of defects in a faulty IC have been published in the literature. Almost all of these approaches can be very broadly classified into two categories — *cause-effect diagnosis* [3]-[15] and *effect-cause diagnosis* [16]-[30]. Both approaches to localization essentially involve finding the best match between the defective behavior of a malfunctioning IC to the faulty behavior predicted by a logic-level fault model. Cause-effect diagnoses pre-compute fault-model behaviors, while effect-cause approaches compute the fault-model behaviors dynamically.

It is important to stress here that all of the cited diagnosis approaches have adopted defect localization as an end objective that PFA takes as an input for subsequent analysis. It is also worth noting that, according to PFA experts, for up to 50% of the investigated cases, the defect could not be found regardless of the localization method utilized. This important observation (unsupported by rigorously collected evidence) is alarming from a yield-learning perspective.

1.3. IC failure characterization

Co-ordinates of the defect location are useful if one can use them to navigate PFA equipment to the IC's defective

regions and also when some repair technique can be applied to enhance yield. However, in the case when neither repair nor PFA is feasible, the defect location becomes irrelevant. What would be more useful in such a case would be the physical characteristics of the defect (*e.g.*, information about which IC layers are involved, or its size, shape, and conductivity). If such characteristics were to be collected for a sufficiently large number of samples, they would enhance yield learning.

1.4. IC diagnosis quality

For the reasons described above, we have focused upon "*defect characterization oriented diagnosis*", *i.e.*, IC test-based diagnosis whose objective is to determine a defect's physical characteristics, not just its location. This research led to the results published in [31]-[38]. Most relevant for the subject of this paper is the diagnostic method proposed and evaluated in the traditional way [34]. Specifically, the evaluation involves checking which of the assumed set of defects, inserted into a circuit under test, could be detected (and identified) by the proposed algorithm[1] through simulation.

The shortcoming of [34] and similar work is that we "declare victory" optimistically, when we understand that 50% of the time PFA cannot locate the defect. One significant source of this undeserved optimism stems from the widely-accepted practice of using unrealistic models of the defective circuit in the assessment of the effectiveness of the proposed diagnostic algorithms. "Unrealistic" refers to the limited spectrum of misbehaviors taken into account during the assessment. In other words, "Mother Nature" produces a much larger variety of circuit misbehaviors than current diagnosis simulations are able to mimic. Consequently, some new diagnosis algorithms that have demonstrated good performance in assessment experiments may be useless when confronted with the physics of the real world. The objective of this paper is to take the first step towards more realistic measurements of the diagnostic accuracy of IC diagnosis methodologies with special attention paid to defect characterization.

1.5. Benchmarking diagnostic algorithms

The key idea investigated in this paper is the concept of an IC diagnosis benchmark. To be useful, such a benchmark should have a simulation environment capable of mimicking portions of real sub-micron ICs affected by a large variety of circuit misbehavior.

We have identified the need for a benchmarking activity by asking two simple questions. First, how can our results, described in [34], be compared to other diagnosis ideas without having a common and consistent basis? Second, how will this or any other diagnosis cope with unanticipated tester results?

To address these questions, we developed the concept of a "benchmarking experiment" that could generate "tester results" for a variety of defects, test sets, and test-application approaches. We have also realized that other developers of diagnosis software could use our benchmark to evaluate their approach by checking whether or not their diagnosis methodology delivers expected results.

A trivial but key component of this idea is the generation of "tester responses" (through simulation) that are "contaminated" by realistic and complex IC misbehaviors. Here, we use the term "contaminated" to stress that the proposed benchmarking methodology, like a real tester, should produce unanticipated circuit misbehaviors. Moreover, it should allow for more realistic assessments as well as an "apples-to-apples" comparison of diagnosis methodologies.

Most of the questions raised in the first section are addressed in the remaining parts of this paper. Specifically, we provide an overview of the method used to create the benchmark data in section 2. We then describe the details of the defects utilized and other characteristics of the benchmark data (sections 3 and 4), which is followed by diagnosis results of various algorithms (section 5). In section 6, we end the paper with our vision for future use and extensions of the diagnosis benchmarking effort along with conclusions.

2. Benchmarking Framework

The best-possible diagnostic benchmark is a large collection of defective ICs, each having well-understood, completely-characterized, and fully-documented deformations[2], which could then be tested with a given test set. However, for many reasons, such a benchmark does not exist. Second-best is an experiment in which a substantial number of diagnosed ICs would be subjected to successful PFA. The results of PFA, in turn, would be used to either confirm or reject the results of diagnosis. But this is not a very realistic option either. (The SEMATECH experiment [39], comes closest to these options.) The third option is to build a simulation environment in which a collection of well-understood and efficient models of defective ICs is used to mimic the testing procedure to generate the desired tester responses. This third option is the framework that we utilize. Similar approaches, albeit with different objectives, have been previously presented, *e.g.* in [40] and [41].

2.1. Structure of the framework

Figure 1 depicts the structure of the simulation framework. It is composed of the Test Set Definition Unit (TSDU), the Collection of Defective Circuit Models (CDCM), the

---

[1] Similar approaches have been applied in almost all of the diagnosis-oriented papers cited.

[2] Deformation is defined as any deviation from the nominal IC [32]. Deformations can be bounded or boundless, the former is typically referred to as a "defect". In this paper, deformation and defect will be used interchangeably.

Simulation Engine (SE), and the Parser of Simulation Results (PSR).

It is envisioned that the user of the framework would be able to design a test set and then generate "tester responses" for all or a subset of the available models of defective circuits. It is also assumed that the user is not able to modify the circuit models or test-application conditions. The only "knob" available to the user in a benchmarking experiment would be the test set.

2.2. Simulation of circuit misbehavior

Simulation fidelity of a circuit malfunction is the key to successful implementation of our benchmarking strategy. There are, however, two fundamental obstacles severely limiting achievable fidelity of defective circuit simulations. The first obstacle involves modeling sub-micron size deformations that cause circuit malfunction. This challenge must be traded off with the size of simulated circuit, that is, the circuit should be large enough to realistically mimic the interaction between the defective portion of the circuit and the rest of the IC. The second obstacle is that accurate simulation techniques, on all levels of design abstraction, have been developed for defect-free ICs. Thus, existing simulators become inefficient, if not useless, if applied outside of the domain of operation for which they have been developed. Also, the limited portfolio of software developed for simulation of defective circuits, such as CARAFE [42][43], VLASIC [44] or CODEF [45][46] does not provide the necessary accuracy or required speed of computation.

Taking into account the above tradeoffs and the utility of the available software, it was decided that the simulation of defective circuits should be conducted in the following way:

1. It should rely on a single and widely-available circuit-level simulator such as SPICE.
2. Each deformation should be encapsulated in the form of a separate SPICE input file.
3. The spectrum of simulated deformations should be as large as possible.
4. The mapping between deformation geometry and its circuit-level implementation should be accomplished in any conceivable way including automatic and manual generation of models.

2.3. Benchmark circuit

The right choice for the benchmark circuit size is not a trivial task because of the already mentioned tradeoff of
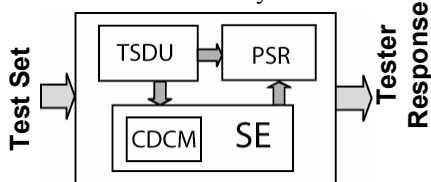

Figure 1: Structure of the simulation framework.

computational complexity versus simulation fidelity. Equally important are the circuit's structure, style of layout, input/output count, *etc*. Analysis of various options quickly led to the conclusion that there is no single ideal solution that would satisfy a majority of the objectives and constraints. Therefore, the first circuit utilized for our benchmark experiment was selected to represent a typical

Gate Type	# of Gates	Net Type	# of Nets
Inverter	27	Primary input	14
Nand2	47	Primary output	8
Nor2	10	Power	2
Nand3	8	Inter-gate signals	109
Nor3	3	Intra-gate signals	85
Total	95	Total	196

Table 1: Key characteristics of the chosen benchmark circuit.
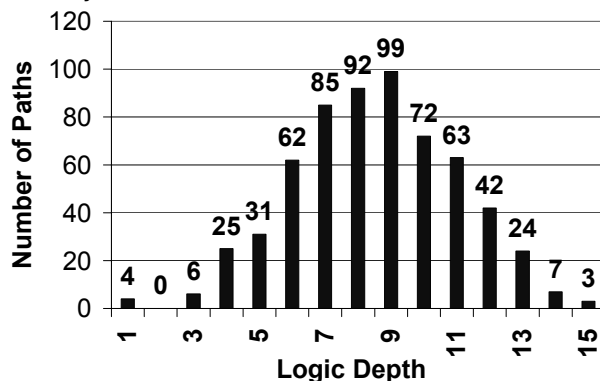

Figure 2: Histogram of logic depth for 615 paths.

portion of an IC design that enabled, as much as possible, a comprehensive characterization.

Hence, we have chosen a 4-bit ALU similar to the '181 [47] defined for TTL. The ALU was synthesized from a Verilog description using Synopsys Design Compiler [48] with a small library of 0.18μm CMOS standard cells that only includes an inverter and 2- and 3-input NAND and NOR gates. Table 1 and Figure 2 summarize key characteristics of the benchmark circuit.

Placement and routing were performed using Silicon Ensemble from Cadence [49], assuming the availability of a 0.18 μm, 5-metal-layer CMOS process. (The design, however, only required 4 metal layers.) In addition to the logic gates, a number of "filler gates" were placed to assure uniform pattern density, a requirement for most modern IC processes. Key characteristics of the resulting layout are shown in Table 2 and Figure 3. The netlist in SPICE format was extracted using a simple approach based on the public domain layout tool Magic [50]. Transistor device models for circuit simulation were obtained for TSMC's 0.18μm CMOS process from the publicly accessible MOSIS webpage [51].

The circuit characteristics of the ALU implementation indicate it is far from its optimal version. It is not very

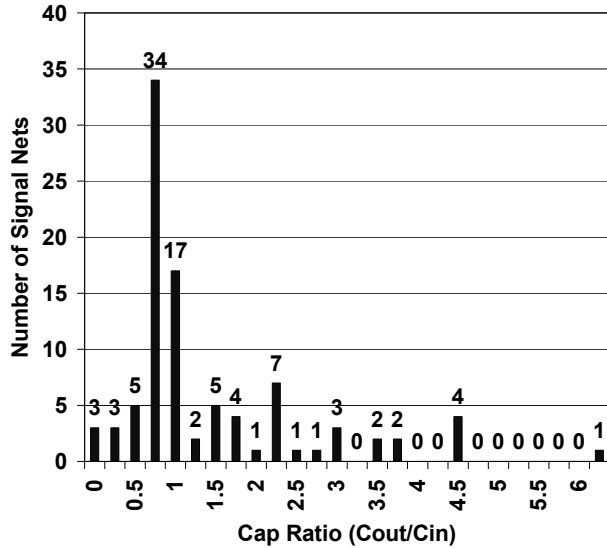| Poly	Metal 1	Metal 2	Metal 3	Metal 4
8.27%	29.63%	11.07%	8.7%	2.12%

Table 2: Metal layer coverage of benchmark layout.

Figure 3: Histogram of signal load capacitance Cout to driving gate input capacitance Cin.

dense and its timing characteristics are sub par. But this implementation choice was purposeful in that it represents the typical design style utilized by industry to meet time-to-market constraints.

3. Spectrum of deformations

One of the critical elements of the benchmark is the choice of IC structure deformations that should (and can) be accurately modeled on the circuit level of design abstraction.

In the following description, we categorize all our bounded deformations (*i.e.,* defects) as either simple single spots of extra metal (all of them are of "6 PM" shape according to the classification proposed in [32] and [33]) or as defects having complex geometry.

3.1. Simple metal spot defects

Even in today's very complex integrated circuits, simple metal spot defects that cause shorts between two or more metal lines can be considered the dominant source of IC failures. Therefore, roughly 90% (1000 spot defects) of the deformations included in our benchmark consist of this type. A complete probability (yield impact) analysis for metal shorts was conducted for our benchmark circuit to obtain a realistic set of shorts from the large pool of potential metal shorts.

The probability of a metal short depends on two factors: (1) the critical area [52][53] of the short and (2) the size distribution of the bounded deformation. First, the critical area for all metal shorts (using micro-event

Type	Number of Shorts	Normalized Probability of Occurrence
2-line	507	86.28%
3-line	377	11.63%
4-line	105	1.90%
≥5-line	11	0.19%
Single stuck-at-1	56	3.31%
Single stuck-at-0	87	5.73%
Multiple stuck-at-1	15	0.23%
Multiple stuck-at-0	154	3.41%
Short between V_{DD} & GND	12	10.47%
Signal line shorts	676	76.85%
Non-feedback	381	55.38%
Feedback	619	44.62%
Inter-gate feedback	*48*	*3.11%*
Intra-gate feedback	*275*	*33.29%*
Both	*296*	*8.22%*

Table 3: Characteristics of the 1000 selected metal-line shorts caused by spot defects.

analysis [54]) was extracted using SiCat[3]. A total of 3,878 metal shorts were obtained from spot defects with a circular shape and radius ranging between 0.15μm and 2μm. Then, a "reasonable" defect size density distribution $D(r)$ for spot defects was modeled using the common power-law theory for spot defect radius r [53][55]:

$$D(r) = \frac{2(p-1)D_0 X_0^{(p-1)}}{p+1} \frac{1}{r^p} \qquad r \geq X_0$$

where p (power parameter), D_0 (defect density), and X_0 are each experimentally determined. The distribution parameters for our benchmark experiment are chosen as follows: p=3, D_0=0.1 cm^{-2} and X_0 much smaller than the minimum feature size of the layout. This set of parameters describes defect size distribution of an average, but stable IC manufacturing line [56].

Finally, the shorts were ranked according to their probability of occurrence using the simple critical area-based yield model. The 1000 most-likely shorts were included in our diagnosis benchmark experiment. All 1000 shorts were modeled as 10 Ω resistors in the SPICE netlist.

To gauge the variety of the metal shorts, we extracted several characteristics and listed the results in Table 3 and Table 4. The data in Table 3 is based on the number of signal lines involved in a given short and the logic-level relation among the shorted lines (if any). Table 3 reveals that almost 50% of the chosen shorts involve only two signal lines. However, 3-line and 4-line shorts also contribute a significant number to the total. Please note that the probability numbers in the rightmost column sum to 100% within each set of shorts considered.

---

[3] PDF Solutions, Inc., San Jose, CA.

The second set of data given in Table 3 is based on the involvement of a power line in an instance of a short. A short is classified as stuck-at-0(1) if it involves GND(VDD) and one other signal line. A short is labeled a multiple stuck-at-0(1) if more than one signal line and GND(VDD) are shorted by the defect. A short that does not include either VDD or GND is referred to as a signal-line short.

Layer	Normalized Probability of Occurrence
Metal-1	43.99%
Metal-2	29.01%
Metal-3	22.63%
Metal-4	4.37%

Table 4: Probability of shorts in metal layers.

We also determined if shorted signal lines cause structural feedback that can lead to oscillations or unwanted sequential behavior [57]. The large number of feedback shorts (619 compared to 381 non-feedback shorts) is expected due to the nature of spot defects and the close proximity of gate I/O. The last row in Table 3 labeled "Both" gives the number of shorts with more than two signal lines, where both inter- and intra-gate feedback exist within the multi-line short.

The numbers in Table 4 represent the probabilities of a metal short affecting a specific metal layer in our design.

3.2. Defects with complex geometry

We added another 108 defects of complex geometry to the 1000 shorts. The added categories of defects are as follows.

Metal Clusters. Clusters of unwanted, extra metal are found in all metal layers of even advanced IC technologies [36]. Metal clusters can be large and small, sparse and dense, and may have boundaries of various shapes. In our experiment, we included ten metal clusters with three in Metal 1, two in Metal 2, and four in Metal 3. For each metal layer, we placed the same small rounded cluster of
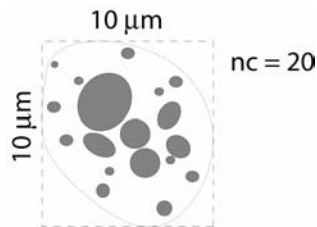

Figure 4: Cluster of metal defects.

20 components (nc=20) in three random locations as shown in Figure 4. In addition, we placed a large elongated cluster in Metal 3 as shown in Figure 5. In SPICE, all multi-line shorts caused by clusters of metal defects were modeled as a set of 10Ω resistors.

Metal Stringers. Metal stringers, i.e., very narrow connections of metal and/or other material (e.g., titanium nitride) located in tiny grooves in between metal lines may
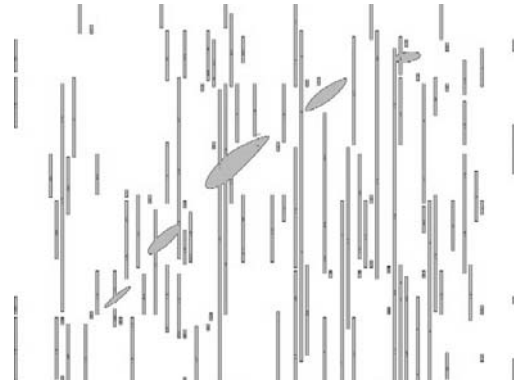

Figure 5: A cluster defect shorts metal lines in Metal 3.

occur in the development phase of a new process. They appear in pattern-specific locations in the metal layer [58] or in places caused by random imperfections due to CMP. In order to model these metal stringers, a layout manipulation algorithm, similar to the algorithm proposed in [58], was written using SiCat. This algorithm was used to determine locations of stringers in Metal layers 2 and 3. For example, Figure 6 shows the affected nets in metal 3 of our benchmark. In SPICE, they were modeled as complex multi-line bridges with values of resistance ranging between 100Ω-10kΩ.

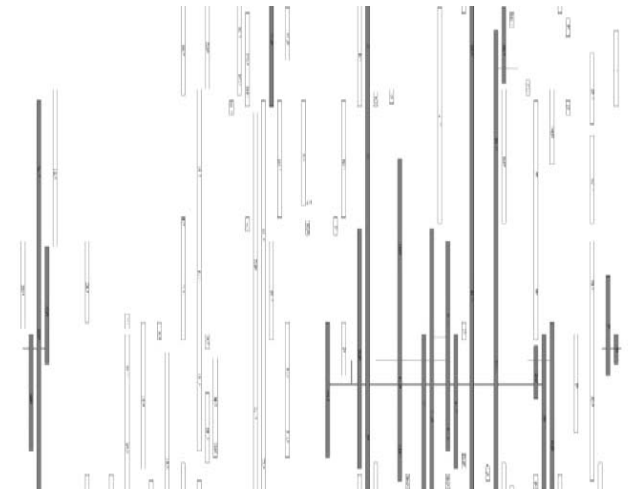Figure 6: Circuit nets (highlighted in bold) affected by stringers in Metal 3.

Poly Stringers. "Poly" (polysilicon) stringers, like metal stringers, should not occur in a well-developed process. But they can occur in products fabricated with immature technologies. Poly stringers can cause resistive shorts between the inputs of one or more gates. For our experiment, we assume that a poly stringer deformation affects each defective gate at the same time. Each resulting resistive bridge among the gate inputs were modeled using resistors ranging from 100Ω-10MΩ in SPICE. A total of seven poly stringers were added to our set defects.

"Monster" Poly Defect. Even modern processes can generate large deformations of a complex geometry [36] albeit very infrequently. This type of defect is represented in our benchmark by a single region of extra poly material as shown in Figure 7. This "monster" defect causes shorts, opens, and missing transistors. It is modeled in SPICE as a manually-generated network of resistors and transistors.

"Nasty Poly-Spot Defect (NPSD)". Spots of extra poly in modern processes, especially processes that use local interconnect, may result in a complex deformation of the circuit's network. For example, a spot of extra poly can
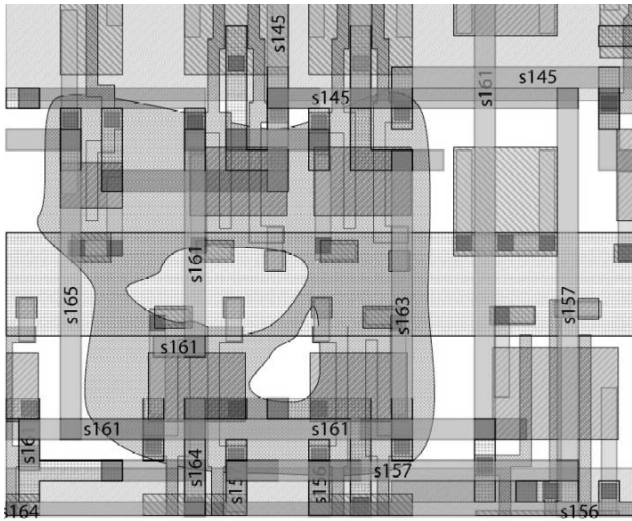

Figure 7: Complex poly defect causing shorts, opens, and missing transistors.

cause a short between one or two poly regions and a contact, while at the same time obstruct the contact between the source (or drain) and the Metal-1 interconnect layer [31]. In our benchmark, we included 50 such defects. They were automatically generated from the circuit layout using a custom algorithm that utilizes SiCat. In SPICE, these deformations were modeled as low-resistance shorts between poly and a contact. In addition, all the transistors with an obstructed source or drain contact were removed from the circuit netlist.

Resistive Contacts. Contacts to either the drain or source (due to lithography imperfections or other reasons) may become highly resistive. We arbitrarily selected a small region containing sources and drains of five transistors in the layout for introducing resistive contacts. For SPICE simulations, each contact was modeled as a 100kΩ resistor.

Resistive Vias. Vias, like contacts, despite their geometrical simplicity, are very prone to process deviations and defects. We selected 3 via sites, all on the critical path. Each via was made highly resistive using resistances from 200Ω to 5kΩ. The electrical model utilized for resistive vias is identical to the one chosen for resistive contacts except for the decreased resistor values.
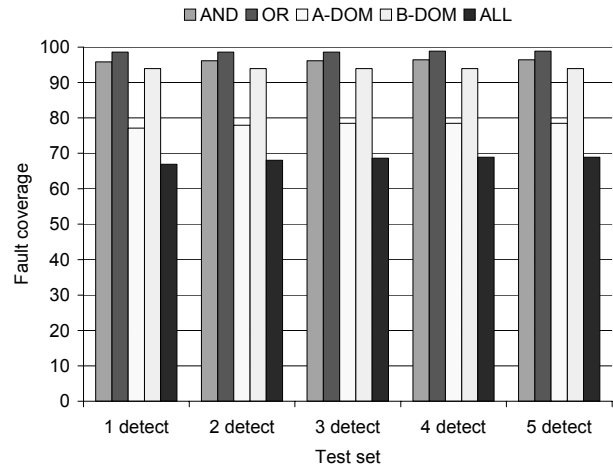

Figure 8: Fault coverage for two-line bridges.

4. Test

In this section, we describe the test sets utilized and the generation of the "tester response" using SPICE.

4.1. Selected test sets

To test the benchmark, we used several N-detect test sets with N=1, 2, 3, 4, and 5 [59][60]. We generated the N-detect tests using the ATPG tool Atalanta [61] in conjunction with our in-house fault simulator [37]. We used five different test sets to understand the test-related characteristics of the defects in our benchmark. We evaluated the quality of each test set against the single-stuck line (SSL) fault model, various two-line bridging fault models, and the transition fault model. For bridges, we used all the two-line bridges that resulted from our critical area analysis of the layout.

4.2. Test set characteristics

All five test sets achieve 100% SSL fault coverage since there are no redundant faults. While the 1-detect test set achieves 97.18% coverage of transition faults, all test sets 2-detect and higher achieve 100% transition fault coverage. For calculating coverage of two-line bridges, we used all the wired bridge models (wired-AND, wired-OR, and wire-dominant where one signal line A dominates the other line B and vice-versa). The bridge fault coverages are shown in Figure 8. For each test set, the last column "ALL" represents the intersection of the first four columns. In other words, we only consider a bridge between a pair of lines to be detected when each of the four bridge types are detected by the respective test set. Although the coverages reported in Figure 8 do not account for resistive or feedback bridges that can cause oscillation or sequential behavior, they do provide an estimate of the quality of the test sets with respect to two-line bridges.

4.3. Test simulation

The circuit simulator SPICE was used to predict the behavior of defective circuits. For each defect instance, we generated a "tester response" for every test set for three different voltages: nominal (1.8V), low (0.9V), and high (2.4V).

For each test set, we performed three types of measurements: slow voltage (structural) test, IDDq test, and "delay" (at-speed) test. In our SPICE setup, results of voltage test are determined from the voltage levels of SPICE waveforms of ALU outputs after all outputs have settled. Delay test samples output voltage waveforms 2.55ns after test application. An IDDq test fails if the measured current exceeds a threshold of 0.7µA. Similar to the style of [39], Figure 9 shows the number of defects that fail each type of test for a 1-detect set and a 5-detect set for each value of supply voltage. For example, for a nominal supply voltage of 1.8V, we have 1049 and 1051 defects that fail slow, IDDq and at-speed test for 1- and 5-detect test sets, respectively. However, 13 defects pass all tests for V_{DD}=1.8V. Only a low-voltage test (V_{DD}=0.9V) is able
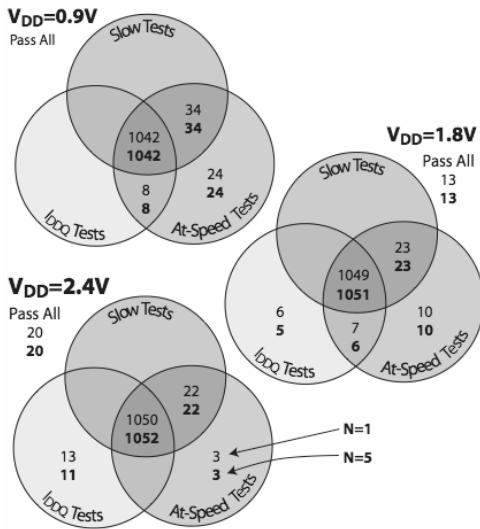

Figure 9: Number of defects detected by each test type for 1- and 5-detect test sets.

to detect all defects.

5. Diagnosis

Our objective is to provide a framework for assessing different diagnosis algorithms. To determine whether or not the proposed benchmark delivers the desired characteristics, we conducted the following diagnosis experiment. We implemented and compared results of five different diagnosis algorithms. In the following subsections, we describe the diagnosis algorithms considered for this experiment, their diagnosis results for

our benchmark, and our observations derived from these results.

5.1. Diagnosis algorithms

Two of the five selected diagnosis algorithms are commercial tools and are referred to as "Tool A" and "Tool B". Without going into the details of these two tools, it suffices to say that they both focus upon defect localization, that is, identification of potentially faulty signal lines. We implemented two diagnosis algorithms based on classical stuck-at diagnosis. While the first of these two algorithms, called "SSL/failing", uses only failing (voltage) test patterns, the second ("SSL/all") uses both failing and passing test patterns. These two algorithms are used to represent classical localization-based diagnosis. The rationale behind using "SSL/all" in addition to "SSL/failing" is to see the effect of using passing patterns on the overall quality of diagnosis. The quality of diagnosis is ambiguous and depends really upon the intended objective. We will define diagnosis quality, as used in the context of this paper, later in Section 5.2. The fifth diagnosis algorithm, labeled "PBI", is described in [34]. PBI (Progressive Bridge Identification) was developed to diagnose two-line bridging defects only. PBI was selected as an example of a defect-specific diagnosis algorithm. We used it to measure its performance and selectivity for tester responses generated by non-bridge defects.

5.2. Diagnosis results

The five diagnosis algorithms were used to diagnose the tester responses for all the simulated defects. We will henceforth refer to the result of diagnosis as *diagnosis callout*. In order to compare the callouts of different diagnosis algorithms, we will next define *Diagnosis Quality*.

As mentioned earlier, diagnosis quality depends upon the intended objective, which can either be localization or defect characterization. Here, we will define diagnosis quality in terms of localization alone.

Assuming that the set of signal lines affected by a defect is represented by {Real_lines} and the set of signal lines in a diagnosis callout is represented by {Reported_lines}, the Quality of Diagnosis can be quantified using the following four categories:

- *Exact diagnosis*: {Real_lines} exactly matches {Reported_lines}.
- *Contained*: {Real_lines} is a proper subset of {Reported_lines}.
- *Partial*: The intersection between {Real_lines} and {Reported_lines} is non-empty and the diagnosis is neither exact nor contained.
- *Empty*: {Reported_lines} is empty.

- *Misleading*: {Reported_lines} is non-empty and the intersection between {Real_lines} and {Reported_lines} is empty.
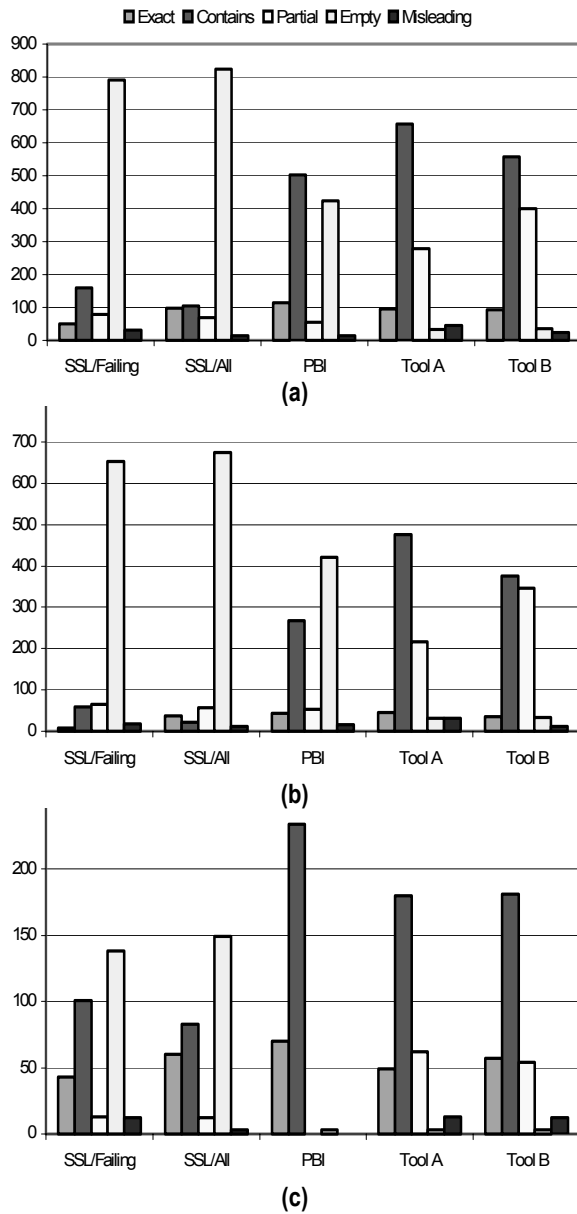

(a)

(b)

(c)

Figure 10: Results of diagnosis for different diagnosis algorithms for V$_{DD}$=1.8V: (a) all benchmark defects, (b) non-classical defects only, and (c) classical defects only.

Based on the above, callouts of the five diagnosis algorithms are categorized in Figure 10(a). The results shown in Figure 10(a) are for tester responses generated at the nominal voltage (1.8V) for a slow-speed, 5-detect test set. The results for other voltages are quite similar and are therefore omitted.

5.3. Observations

As can be seen from Figure 10(a) both "SSL/failing" and "SSL/all" perform poorly for most of the defects, with the callout being empty in a majority of cases. This is an expected and good result – expected because very few of the considered defects mimic stuck-at behavior, and good because this result indicates that our benchmark does in fact represent a real-life situation where most defects do not behave as stuck-at faults. The importance of passing patterns is indicated by the increase in the number of exact diagnoses and the corresponding decrease in the contained cases for "SSL/all" over "SSL/failing". Passing patterns assist in removing spurious lines in the diagnosis callout.

The bridge-selective algorithm PBI fares better than the first two diagnosis algorithms simply because of the significant number of signal line shorts in the database of defects (Section 3.1). The large number of empty diagnoses in the first three algorithms is a good indicator of how defect-specific diagnosis algorithms completely ignore defect behaviors outside their range of assumed misbehaviors.

Both "Tool A" and "Tool B" are commercial tools and are much more conservative in terms of identifying potentially faulty signal lines. They are conservative because they identify a large portion of the IC as potentially defective if an explanation for a defect's misbehavior cannot be precisely found. As a result, they have a larger number of partial and contained diagnoses and a smaller number of exact diagnoses as compared to PBI.

To more precisely measure the performance of these diagnosis algorithms, we partitioned our list of defects into two categories, namely, classical and non-classical defects. All non-feedback two-line shorts (including shorts to either supply) are categorized as classical. We refer to these shorts as classical because of the ubiquity of test and diagnosis algorithms related to this class of defects. The remaining defects are categorized as non-classical. The diagnosis experiment for these two defect categories was repeated. The results are shown in Figure 10(b) and Figure 10(c), respectively.

The chart in Figure 10(b) indicates that both "Tool A" and "Tool B" perform much better than others in terms of having fewer empty diagnoses for non-classical defects. As shown in Figure 10(c), PBI performs the best among all five algorithms for classical defects – it has the most number of exact and contained diagnoses and zero cases of empty diagnosis. As was mentioned earlier, this is because PBI was intended for the diagnosis of these classical defects.

6. Conclusions

Since test will become a significant source of valuable information in yield learning activities, the importance of test-based diagnosis is growing rapidly. In this paper, we

have proposed a simple, yet powerful strategy using a small circuit and a set of bounded deformations (*i.e.,* defects) to measure the effectiveness of diagnosis techniques.

We have generated more than 1000 instances of possible malfunctions of the benchmark. We demonstrated that complex relationships between defects and test results can occur in even a small circuit. Simulation results seem to mimic many symptoms observable on a real tester. In addition, we have shown that there exists a diverse spectrum of circuit misbehaviors for simple spots of extra metal.

We also demonstrated that even a relatively simple benchmark can discriminate between various diagnosis methodologies. Hence, we claim that benchmarking of diagnosis methodologies against more realistic models is feasible, and should be mandated in the R&D of IC diagnosis assessments. We also observed that the relationships between layout, defect geometry, and test results can be a very powerful tool in building an understanding of defect-yield loss relationships. Therefore, we plan to use this benchmarking approach in our research and *we are prepared to distribute our benchmarking framework freely* to assist anyone willing to confront new and existing diagnosis challenges with a portion of physical reality.

Acknowledgements

References

[1] Semiconductor Industry Association, "*International Technology Roadmap for Semiconductors*", 2003 Edition. http://public.itrs.net/Files/2003ITRS/Home2003.htm.

[2] W. Maly, "High Levels of IC Manufacturability: One of the Necessary Prerequisites of the 1997 SIA Roadmap," *Proc. of IEDM*, San Francisco, CA, pp. 759- 762, Dec. 1998.

[3] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, Piscataway, NJ, 1990.

[4] J. M. Galey *et al.*, "Techniques for the Diagnosis of Switching Circuit Failures," *IEEE Trans.. on Communications and Electronics*, vol. 83, pp. 509-514, Sept. 1964.

[5] J. P. Roth, "Diagnosis of Automata Failures: A Calculus and a Method," *IBM Journal of Research and Development*, vol. 10, no. 4, pp. 278-291, July 1966.

[6] J. Richman and K. R. Bowden, "The Modern Fault Dictionary," *Proc. of International Test Conference*, pp. 696-702, Sept. 1985.

[7] P. G. Ryan, S. Rawat and W. K. Fuchs, "Automated Diagnosis of VLSI Failures," *Proc. of VLSI Test Symposium*, pp. 187-192, April 1991.

[8] I. Pomeranz and S. M. Reddy, "On the Generation of Small Dictionaries for Fault Location," *Proc. of International Conference on Computer-Aided Design*, pp. 272-279, Nov. 1992.

[9] S. D. Millman, E. J. McCluskey and J. M. Acken, "Diagnosing CMOS Bridging Faults with Stuck-At Fault Dictionaries," *Proc. of International Test Conference*, pp. 860-870, Oct. 1990.

[10] P. G. Ryan, W. K. Fuchs and I. Pomeranz, "Fault Dictionary Compression and Equivalence Class Computation for Sequential Circuits," *Proc. of International Conference on Computer-Aided Design*, pp. 508-511, Nov. 1993.

[11] V. Boppana, I. Hartanto and W. K. Fuchs, "Full Fault Dictionary Storage Based on Labeled Tree Encoding," *Proc. of VLSI Test Symposium*, pp. 174-179, April 1996.

[12] V. Boppana and W. K. Fuchs, "Fault Dictionary Compaction by the Elimination of Output Sequences," *Proc. of International Conference on Computer Aided Design*, pp. 576-579, Nov. 1994.

[13] B. Chess and T. Larrabee, "Creating Small Fault Dictionaries," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 3, pp. 346-356, March 1999.

[14] D. B. Lavo and T. Larrabee, "Making Cause-Effect Cost-Effective: Low Resolution Fault Dictionaries," *Proc. of International Test Conference*, pp. 278-286, Oct. 2001.

[15] S. D. Millman and J. M. Acken, "Diagnosing CMOS Bridging Faults with Stuck-At, IDDq, and Voting Model Fault Dictionaries," *Proc. of IEEE Custom Integrated Circuits Conference*, pp. 409-412, May 1994.

[16] S. Venkataraman and S. B. Drummonds, "POIROT: A Logic Fault Diagnosis Tool and Its Applications," *Proc. of International Test Conference*, pp. 253-262, Oct. 2000.

[17] M. Abramovici and M. A. Breuer, "Multiple Fault Diagnosis in Combinational Circuits Based on Effect-Cause Analysis," *IEEE Trans. on Computers*, vol. C-29, no. 6, pp. 451-460, June 1980.

[18] J. A. Waicukauski and E. Lindbloom, "Logic Diagnosis of Structured VLSI," *IEEE Design and Test of Computers*, pp. 49-60, Aug. 1989.

[19] H. Cox and J. Rajski, "A Method of Fault Analysis for Test Generation and Fault Diagnosis," *IEEE Trans. on Computer-Aided Design*, vol. 7, no. 7, pp. 813-833, July 1988.

[20] M. Marzouki, J. Laurent and B. Courtois, "Coupling Electron-Beam Probing with Knowledge Based Fault Localization," *Proc. of International Test Conference*, pp. 238-247, Oct. 1991.

[21] J. Savir and J. P. Roth, "Testing for, and Distinguishing Between Failures," *Proc. of 12th Fault Tolerant Computing Symposium*, pp. 165-172, June 1982.

[22] S. Venkataraman, I. Hartanto and W. K. Fuchs, "Dynamic Diagnosis of Sequential Circuits," *Proc. of VLSI Test Symposium*, pp. 198-203, April 1996.

[23] S. Venkataraman *et al.*, "Rapid Diagnostic Fault Simulation of Stuck-At Faults in Sequential Circuits using Compact Lists," *Proc. of Design Automation Conference*, pp. 133-138, June 1995.

[24] P. G. Ryan, S. Rawat and W. K. Fuchs, "Two-Stage Fault

Location," *Proc. of International Test Conference*, pp. 963-968, Oct. 1991.

[25] S. B. Drummonds *et al.*, "Bridging the Gap Between Logical Diagnosis and Physical Analysis," *IEEE International Workshop on Defect Based Testing*, April 2002.

[26] S. Chakravarty and Y. Gong, "An Algorithm for Diagnosing Two-Line Bridging Faults in Combinational Circuits," *Proc. of Design Automation Conference*, pp. 520-524, June 1993.

[27] J. Wu, G. S. Greenstein and E. M. Rudnick, "A Fault List Reduction Approach for Efficient Bridge Fault Diagnosis," *Proc. of Design, Automation and Test in Europe Conference*, pp. 780-781, March 1999.

[28] Y. Sato *et al.*, "A Persistent Diagnostic Technique for Unstable Defects," *Proc. of International Test Conference*, pp. 242-249, Oct. 2002.

[29] T. Bartenstein *et al.*, "Diagnosing Combinational Logic Designs Using the Single Location At-a-Time (SLAT) Paradigm," *Proc. of International Test Conference*, pp. 287-296, Oct. 2001.

[30] D. B. Lavo, I. Hartanto and T. Larrabee, "Multiplets, Models, and the Search for Meaning: Improving Per-Test Fault Diagnosis," *Proc. of International Test Conference*, pp. 250-259, Oct. 2002.

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

[32] W. Maly *et al.*, "Deformations of IC Structure in Test and Yield Learning," *Proc. of International Test Conference*, pp. 856-865, Oct. 2003.

[33] W. Maly *et al.*, "A Yield Modeling and Test Oriented Taxonomy of Deep Submicron Technology Induced IC Structure Deformations," *Proc. of International Symposium for Testing and Failure Analysis*, Nov. 2003.

[34] T. Vogels *et al.*, "Progressive Bridge Identification," *Proc. of International Test Conference*, pp. 309-318, Oct. 2003.

[35] R. Desineni *et al.*, "A Multi-Stage Approach to Fault Identification Using Fault Tuples," *Proc. of International Symposium for Test and Failure Analysis*, pp. 496-505, Nov. 2003.

[36] T. Zanon *et al.*, "Analysis of IC Manufacturing Process Deformations: An Automated Approach Using SRAM Bit Fail Maps," *Proc. of International Symposium for Test and Failure Analysis*, pp. 232-241, Nov. 2003.

[37] K. N. Dwarakanath and R. D. Blanton, "Universal Fault Simulation Using Fault Tuples," *Proc. of Design Automation Conference*, pp. 786-789, June 2000.

[38] R. Desineni, K. N. Dwarakanath, and R. D. Blanton, "Universal Test Generation Using Fault Tuples," *Proc. of International Test Conference*, pp. 812-819, Oct. 2000.

[39] P. Nigh *et al.*, "An Experimental Study Comparing the Relative Effectiveness of Functional, Scan, IDDq, and Delay-Fault Testing," *Proc. of VLSI Test Symposium*, pp. 459-463, April-May 1997.

[40] Y. Kwon and D. M. H. Walker, "Yield Learning via Functional Test Data", *Proc. of International Test Conference*, pp. 626-635, Oct. 1995.

[41] Z. Stanojevic *et al.*, "Computer-Aided Fault to Defect Mapping (CAFDM) for Defect Diagnosis", *Proc. of International Test Conference*, pp. 729-738, Oct. 2000.

[42] A. L. Jee and F. J. Ferguson, "Carafe: An Inductive Fault Analysis Tool for CMOS VLSI Circuits," *Proc. of VLSI Test Symposium*, pp. 92-98, April 1993.

[43] A. L. Jee, "Carafe: An Inductive Fault Analysis Tool for CMOS VLSI Circuits," *M.Sc. Thesis*, Dept. of Computer Engineering, UC Santa Cruz, June 1991.

[44] D. M. Walker and S. Director, "VLASIC: A Catastrophic Fault Yield Simulator for Integrated Circuits," *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 5, no. 4, pp. 541-556, Oct. 1986.

[45] J. Khare and W. Maly, "From Contamination to Defect Fault and Yield Loss," *Kluwer Academic Publisher*, 1996.

[46] J. Khare, C. Kellen, and W. Maly, "CODEF: A Contamination-Defect-Fault Mapper," *SRC-CMU Center for Computer-Aided Design Technical Report*, Carnegie Mellon University, May 1995.

[47] *TTL Databook*, Texas Instruments, Dallas, TX, http://www.ti.com.

[48] *The Design Compiler Reference Manual*, Synopsys, Mountain View, CA, http://www.synopsys.com.

[49] *The Silicon Ensemble Reference Manual*, Cadence Design Systems, Inc., San Jose, CA, http://www.cadence.com

[50] *Magic VLSI Layout Tool 7.1*, http://vlsi.cornell.edu/magic.

[51] The MOSIS Service, Marina del Rey, CA, http://www.mosis.org

[52] W. Maly and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation," *Electronics Letters*, Vol. 19, No. 6, pp. 226-227, 17th March 1983.

[53] C. H. Stapper, "Modeling of defects in integrated circuit photolithography patterns," *IBM J. Res. Develop.*, vol. 28, no. 4, pp. 461-474, July 1984.

[54] J. Khare, D. Feltham and W. Maly, "Accurate Estimation of Defect-Related Yield Loss in Reconfigurable VLSI Circuits," *IEEE Journal of Solid State Circuits*, No. 2, pp. 146-156, Feb. 1993.

[55] A. V. Ferris Prabhu, "Role of Defect Size Distribution in Yield Modeling," *IEEE Trans. on Electron Devices*, vol. 32., no. 9, pp. 1727-1736, Sept. 1985.

[56] J. Khare *et al.*, "Yield-Oriented Computer-Aided Defect Diagnosis," *IEEE Trans. on Semiconductor Manufacturing*, vol. 8, no. 2, pp. 195-206, Aug. 1995.

[57] Y. Miura and S. Seno, "Behavior Analysis of Internal Feedback Bridging Faults in CMOS Circuits," *Journal of Electronic Testing* (JETTA), vol. 18, no. 2, April 2002.

[58] P. Simon, W. Maly, D. K. de Vris Brules, "Design Dependency of Yield Loss Due to Tungsten Residues in Spin on Glass Based Planarization Process," *Proc. of ISSM*, pp. 87-88, Oct. 1997.

[59] S. C. Ma, P. Franco, and E. J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiments Results," *Proc. of International Test Conference*, pp. 663-672, Oct. 1995.

[60] E. J. McCluskey and C. Teng, "Stuck-at Fault Tests vs. Actual Defects," *Proc. of International Test Conference*, pp. 336-342, Oct. 2000.

[61] H. K. Lee and D. S. Ha, "On the Generation of Test Patterns for Combinational Circuits," *Technical Report no. 12-93*, Virginia Polytechnic Institute and State University.