NOIDA: Noise-resistant Intra-cell Diagnosis

Soumya Mittal and R. D. (Shawn) Blanton
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
https://www.ece.cmu.edu/∼actl/

***Abstract*—The goal of diagnosis is to identify defect locations and subsequently, identify the root cause so as to minimize (and ideally eliminate) the need for physical failure analysis. With advanced technology nodes, there has been an increasing number of front-end (i.e., within a standard cell) defects. Conventional diagnosis approaches typically fail to localize such defects. In addition, circuit-level noise can change the tester response in an unexpected way, and can decrease the quality of diagnosis. This work describes a noise-resistant approach called NOIDA (NOise-resistant Intra-cell DiAgnosis) for effectively diagnosing cell-level defects based on the analysis of the intra-cell physical neighborhoods surrounding likely defect locations. Defect behavior is derived based on the neighborhood, instead of relying on a specific fault model. Experiments demonstrate the effectiveness of NOIDA using a library of standard cells. The results show that for over 16,000 static and sequence-dependent defects, the method achieves an average resolution improvement of 12.1% over prior work with a small accuracy loss (specifically, 1.6%). Additionally, NOIDA is found to be more robust to noise in the tester response. Specifically, in the presence of noisy tester response, NOIDA attains an accuracy of 97.6% with an average resolution improvement of 48.6% over prior work.**

I. Introduction

Diagnosis is a software-based process for determining defect locations within a failing circuit and sometimes, in addition, can characterize the nature of the defects residing in those locations. Suspect defect locations, often called candidates[1], are determined by comparing the observed circuit response with the expected, defect-free response. The outcome of diagnosis is sometimes used to aid physical failure analysis (PFA) in order to identify the root cause of the failure. The quality of diagnosis, which is typically defined by *resolution* (the number of locations reported) and *accuracy* (i.e., are the reported locations correlated to actual defect locations), is a major factor in determining the success rate of PFA.

Besides guiding PFA, the information obtained from volume diagnosis aids the understanding of failure mechanisms, which consequently facilitates yield learning. Moreover, a variety of volume diagnosis approaches aim at using the diagnoses of a statistically-significant number of failing chips to determine if there are any systematic defects [1], [2]. Diagnosis results can also be used to estimate defect distribution for a population of failing chips [3], and grade the effectiveness of fault models [4].

Numerous techniques have been proposed over the years to improve the diagnosis of a failed circuit. Approaches such as [5]–[11] use a logic-level description of the circuit to find a possible cause and location of failure. To improve defect localization, techniques have been developed that incorporate design layout information [12]–[18]. All of these techniques focus on diagnosing defects outside the cell and report either an interconnect, a standard-cell pin or the cell itself as a possible candidate. These techniques therefore collectively perform back-end defect diagnosis, which means these approaches cannot locate defects inside a cell. It has been shown in [19] however that the root cause of a significant number of manufacturing defects lies within a cell for advanced nanometer technologies (90nm and below). Because one goal of diagnosis is to eliminate the need for PFA, knowledge of the exact defect location is of utmost importance. Complex standard cells such as full adders, multipliers and scan flip flops contain a large number of transistors that makes PFA challenging [20]. Pinpointing each defect location inside a cell will decrease the physical area that must be examined for failure analysis, resulting in more efficient and cost-effective PFA [19]–[22]. For example, if all the candidates are found to be located within the polysilicon layer, the failing die can be de-layered up to the polysilicon layer and all the higher layers can be ignored during PFA. Moreover, some failure mechanisms including poly-contact shorts, poly-contact opens, and poly-active shorts can only exist within a cell and would not be found by a back-end diagnosis technique. Additionally, diagnoses of intra-cell defects can be correlated to identify yield-limiting layout features within a cell.

A diagnosis approach typically correlates observed defect behavior with one or more fault models to find candidates. Many fault models have been created over the years to capture the variety of observed defect behaviors. While fault models such as single stuck-line, wired logic bridge and transition have been commonly used for diagnosis, their ability to precisely capture the defect behavior is decreasing [23]. Fault models that consider layout information have been shown to be more effective in modeling defects [19], [24]. However, as technology advances, new materials and fabrication steps are employed, which results in new defect types and failure mechanisms. New failure mechanisms (e.g., fin-related defects in FinFET-based circuits [25]) can create misbehaviors that are not sufficiently captured by existing fault models [26]. Thus, the unpredictability of defects necessitates the need for a more generalized approach to defect diagnosis.

[1] A candidate can be (a) an interconnect or a cell in the logic-level representation of the circuit, or (b) an intra-cell net in the transistor-level representation of the circuit. Additional information such as its likely behavior and physical location is also sometimes reported depending on the particular diagnosis approach employed.

Even if fault models accurately captured the typical behaviors of the targeted defects, circuit-level noise can cause deviations between the predicted behavior and the behavior observed on the tester. Various sources of noise exist in digital circuits that include process variation, cross-talk signal noise, power supply noise, and substrate-coupling noise [27]. Deviations between the observed and the predicted behavior can also result from inaccurate SPICE models used for defect extraction, and the transistor-level simulation employed for formulating fault models. Depending on the amount of deviation, a weak logic value at a cell output, due to an intra-cell defect, can be interpreted as a logic-1 or a logic-0 by a receiver cell (depending on its switching threshold, which too is affected by noise), and can result in a tester response that is not predicted by the corresponding fault. Thus, circuit-level noise and the resulting ambiguity in signal logic values warrant the need for a noise-resistant diagnosis approach.

This work describes a novel methodology to diagnose front-end defects that we term NOIDA (NOise-resistant Intra-cell DiAgnosis). Here, the defects are assumed to be localized, i.e., the behavior of a defect is influenced by the circuitry within some radius r surrounding it. Hence, NOIDA derives the defect behavior by analyzing the nets surrounding its location, instead of using a particular fault model. The output of NOIDA is a set of candidates, where each candidate is a tuple consisting of its physical location (x-y coordinate and the physical layer) and its likely behavior (and consequently its defect type). Several defect injection experiments are performed using a 45nm standard-cell library [28] to evaluate NOIDA when noise is introduced to the tester response. Results demonstrate better accuracy and significant resolution improvement for intra-cell defects when compared to [19].

The rest of the paper is organized as follows. Section II provides a brief background on front-end diagnosis and motivates NOIDA. This section also describes NOIDA in detail. Experiment results are presented in Section III. The final section concludes the paper.

II. Diagnosis Methodology

Many approaches have been put forward over the years to diagnose intra-cell defects. Work in [29] assumes a defect model for defects within the transistor-level description of a cell and maps these defects to logic-level defects using complex transformation rules. Transformation is applied to cells that adhere to the following criteria: the stuck-at fault simulation response at the output of a cell should match the observed circuit response. Logic-level diagnosis tools can then be used on the modified netlist to find the defective cells, and in turn the intra-cell defects. One main drawback of using this approach is that diagnostic accuracy largely depends on the defect models and the transformation rules. Another disadvantage is that a different model is required for each defect type, which means unknown defect types may go undiagnosed.

In [22], possible defective cell locations (i.e., cell candidates) are identified based on the realistic assumption that the excitation of a cell-internal defect is highly correlated to the input logic values applied to a cell. Logic values at the inputs of each cell candidate are collected for Tester-Fail-Simulation-Fail (TFSF) patterns, that is, patterns that fail on the tester and propagate the error effects from the defect location to the circuit outputs; and Tester-Pass-Simulation-Fail (TPSF) patterns, that is, patterns that pass on the tester but propagate the error effects from the defect location to the circuit outputs. Such input-value combinations will henceforth be referred to as cell-level failing and passing patterns. Cells with inconsistent input conditions, i.e., input conditions that appear in both cell-level passing and failing patterns, are discarded from further analysis. This form of consistency checking is a special case of the approach described in [18]. For the remaining consistent cells, the input conditions are matched with a fault dictionary. The fault dictionary is created by performing a switch-level simulation of various intra-cell defects.

In [19]–[21], [30], [31], a fault dictionary is constructed from extracting realistic defects from physical layout and transistor-level simulation. This has the advantage of generating more accurate responses, though the simulation time may be a limiting factor. Moreover, if the defect extraction or simulation steps are not accurate, then diagnosis will produce inaccurate results.

Methods described in [19]–[22], [30], [31] generate a fault dictionary by extracting intra-cell defects using various techniques. On the other hand, an effect-cause approach is described in [32], where critical path tracing [33] is utilized to trace back from the output of a failing cell to cell inputs. The main drawback of this work is that physical layout information is not considered for diagnosis. For example, bridge defects implicated by this methodology may not be very likely due to lack of proximity.

Prior work on front-end diagnosis discussed up to this point suffers from one major disadvantage – potentially inaccurate defect modeling. NOIDA circumvents this problem by avoiding the use of a specific fault model. Instead, it derives the defect behavior by analyzing the logic activity of the nets surrounding the likely defect location. Here, a defect is assumed to be localized and controlled by the circuitry in close proximity. This assumption holds true for a variety of defects such as bridge, open and transistor defects. The nets near the candidate are collectively referred to as its neighborhood; the logic values applied to the nets in the neighborhood form a neighborhood state. Changes in neighborhood state over time can also be important for sequence- and timing-dependent defects.

Fig. 1 illustrates a typical software diagnosis framework. Given a test response, the first step is to find candidates at the logic level. The result of this logic-level analysis is an initial set of interconnect and cell candidates. Next, the interconnect candidates are examined using back-end diagnosis techniques like [18], [34]. In parallel, each cell candidate identified is further investigated via front-end diagnosis approaches [19]–[22], [29]–[32] to pinpoint defect locations inside a cell.

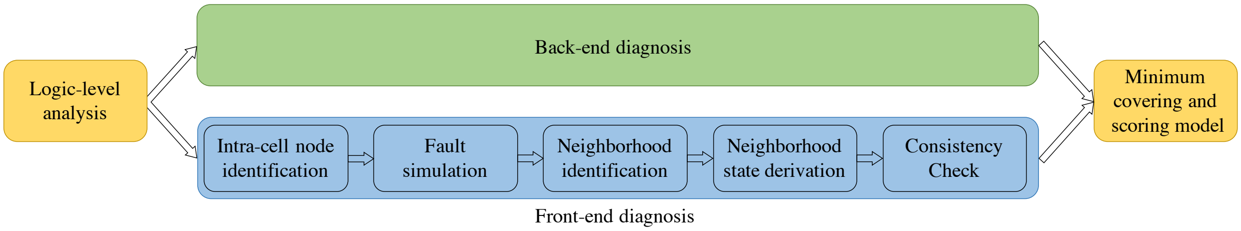
Back-end diagnosis
Logic-level analysis
Intra-cell node identification
Fault simulation
Neighborhood identification
Neighborhood state derivation
Consistency Check
Minimum covering and scoring model
Front-end diagnosis

Fig. 1. Overview of a generic diagnosis framework.

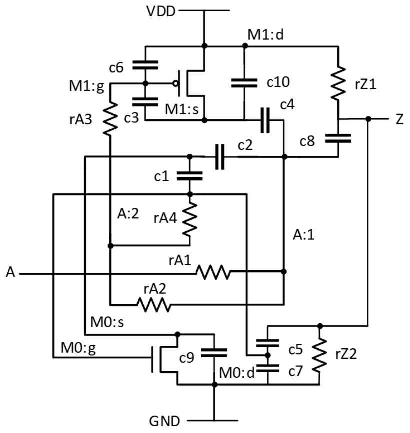

Fig. 2. A schematic view of an inverter cell with parasitics extracted. Parasitics affecting power rails are not shown for clarity.

The resulting set of interconnect and intra-cell candidates are then merged together to constitute a final set of candidates. Minimum set covers are selected from this set and ranked using a scoring model. This paper, in particular, is focused on describing front-end diagnosis. Each step involved in the flow is described next:

1) *Intra-cell node identification*: A transistor-level description of a cell (such as its physical layout, SPICE netlist, etc.) is used to identify intra-cell nodes. A SPICE netlist with parasitics extracted is used here. The inverter schematic is used to illustrate node identification in Fig. 2. The parasitic resistances are denoted using 'r' and the coupling capacitances using 'c'. There are ten internal nodes for this cell, namely, $\{A, A{:}1, A{:}2, Z, M0{:}g, M0{:}d, M0{:}s, M1{:}g, M1{:}d, M1{:}s\}$.

2) *Fault simulation*: Each intra-cell node is faulted at the opposite value of the expected value for the cell-level failing and passing patterns[2]. This is achieved by adding a near-zero resistor between each node and *VDD* (or *GND*, depending on the fault value) in the SPICE netlist. The altered SPICE netlist is then simulated with an analog simulator for the cell-level failing and passing patterns. Each simulation response is digitized using the following criteria: the logic value at the cell output is deemed a logic-1 (logic-0) if the output voltage is more (less) than half of the supply voltage. This logic threshold value depends on the process technology and can be specified by the user. The cell-level passing (failing) patterns for which a faulted node produces and propagates an error to the cell output are the cell-level TPSF (TFSF) patterns for the faulted node. Each faulted node with at least one cell-level TFSF pattern is deemed an initial diagnosis candidate.

3) *Neighborhood identification*: Neighborhood of each node is found from the extracted SPICE netlist of the cell. Neighbors of a node constitute all nodes that are coupled to it by capacitors. This is a reasonable way to identify neighbors and adheres to the localization assumption. For instance, in Fig. 2, three coupling capacitors, $c2$, $c4$ and $c8$, are associated with node $A{:}1$ and hence the neighbors of $A{:}1$ include $\{Z, M1{:}s, M0{:}s\}$. If parasitic extraction is not possible, then the neighbors can simply be found by identifying the internal nodes that are in close physical proximity to the candidate.

4) *Neighborhood state derivation*: Analog voltage values at each of the internal nodes are stored during fault simulation in step 2. Each value is then converted into its logic equivalent by using the same logic threshold value mentioned in step 2. For static defects, the neighborhood state of a candidate is the set of logical values established in its neighborhood for the pattern applied. For sequence-dependent defects, the neighborhood state tracks the logical values for two or more patterns.

5) *Consistency check*: The neighborhood state of each intra-cell candidate is analyzed for cell-level TPSF and TFSF patterns. If the neighborhood state of a candidate is the same for any pair of cell-level TPSF and TFSF patterns, then the candidate is inconsistent and is removed from the candidate set. It should be noted that the amount of inconsistency can be modulated for candidate elimination. Finally, minimum set covers of the consistent intra-cell candidates are selected to jointly explain the failures observed at the cell output.

III. Experiments

This section presents the experiment details of evaluating NOIDA on 79 standard cells within the 45nm standard-cell library of [28].

A population of defective cells is created by injecting cell defects (one at a time) into the layout of each cell. The defects injected include opens, bridges (feedback and non-feedback), stuck-open and stuck-closed transistors with resistance values that range from 1Ω to $20k\Omega$ for bridges, and from $1G\Omega$ to $1k\Omega$ for opens. For each defective cell layout, a corresponding transistor-level netlist is extracted. Because the behavior of the defect is unknown, i.e., whether it is static or sequence-dependent, analog simulation is performed on each altered netlist using an exhaustive two-pattern test set. A defect is

[2] The cell-level failing and passing patterns can be obtained via any logic-level diagnosis technique. However, an exhaustive two-pattern test set is used in Section III because the experiments performed in this work involve diagnosis of individual cells and not a circuit of interconnected cells.

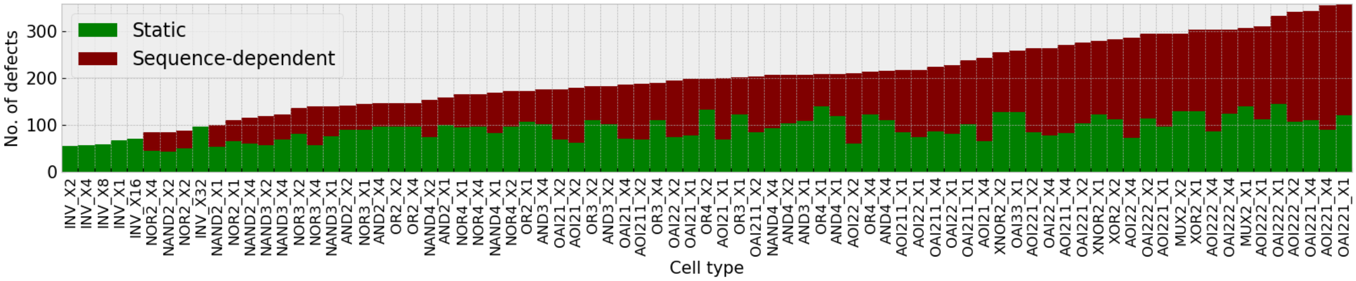

Fig. 3. Distribution of static and sequence-dependent defects for a 45nm standard-cell library [28].

considered detected if the voltage at the cell output deviates from its expected, defect-free value of *VDD* or *GND* by more than 50%. Also, and importantly, patterns that detect the defect are deemed cell-level failing patterns, while the remaining patterns are treated as cell-level passing patterns.

Each defect simulation response is analyzed to determine if it exhibits static or sequence-dependent behavior. A sequence-dependent defect requires a sequence of patterns (two patterns in this case) for detection. Detection of a static defect is independent of the first pattern applied for all two-pattern combinations. In other words, a static defect is detected by a single pattern.

Fig. 3 shows the number of static and sequence-dependent defects for each standard cell in the library. A total of 16,037 defects are injected and simulated, out of which 7,110 (44.3%) are static while 8,927 (55.7%) are sequence-dependent. Out of 7,110 static defects, 449 (6.3%) are open defects, 3,290 (46.3%) are bridge defects, and 3,371 (47.4%) are transistor defects. Similarly, out of 8,927 sequence-dependent defects, 3,711 (41.6%) are open defects, 1,424 (15.9%) are bridge defects, and 3,792 (42.5%) are transistor defects. It is also observed that most of the open defects (89.2%) require a sequence of patterns for detection, and a majority of bridge defects (69.8%) are static.

Each intra-cell defect is diagnosed using NOIDA and [19]. NOIDA returns a set of minimum set covers (step 5 of NOIDA), where each set cover jointly produces a response that exactly matches the observed response at the cell output. Using [19] for diagnosis means that all modeled faults that have a simulation response that matches the observed response are reported as diagnosis candidates. Both diagnosis methodologies are evaluated on two criteria, namely, resolution and accuracy. For NOIDA, resolution is defined as the number of unique intra-cell nodes present in the minimum set covers. For [19], resolution is calculated by counting the number of unique intra-cell nodes corresponding to the faults returned as candidates. For both methodologies, a defect is considered to be accurately diagnosed if the candidates returned include at least one of the intra-cell nodes used for defect injection.

Table 1 shows the number of static and sequence-dependent defects accurately diagnosed by NOIDA and [19]. It indicates that while [19] attains perfect accuracy, NOIDA achieves near perfect accuracy for static defects (99.7%) and an accuracy of 97.4% for sequence-dependent defects.

TABLE 1
DIAGNOSTIC ACCURACY ACHIEVED BY NOIDA AND [19].

Diagnosis approach	Static	Sequence-dependent
NOIDA	7090 (99.7%)	8692 (97.4%)
[19]	7110 (100.0%)	8927 (100.0%)

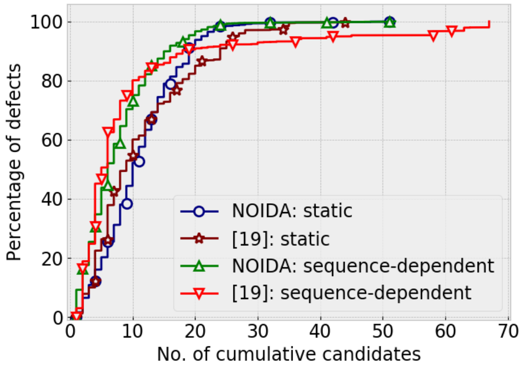

Fig. 4. Cumulative distribution of the number of candidates reported by NOIDA and [19]. Plots marked with "○" and "△" correspond to the resolution obtained by NOIDA for static and sequence-dependent defects, respectively. Plots marked with "*" and "▽" correspond to the resolution obtained by [19] for static and sequence-dependent defects, respectively.

Table 2 categorizes the accuracy attained by NOIDA by defect type for static and sequence-dependent defects. It is observed that 54 transistor defects and 201 open defects are inaccurately diagnosed by NOIDA. These defects are investigated further to determine the reason for inaccuracy. It is discovered that the analog voltage values at the cell output and some of the internal cell nodes lie close to the logic threshold value (i.e., ± 10%) for some patterns. One of the reasons for intermediate voltage at a node is the time at which the voltage of each node is sampled. This observation has two consequences. First, a failing pattern can be interpreted as a passing pattern which can eliminate the correct candidate. Second, voltage (that is close to logic threshold) at an internal node can be inaccurately converted to its logic equivalent, which can change a neighborhood state and in turn, make the correct candidate inconsistent.

Fig. 4 shows the diagnostic resolution achieved by NOIDA

TABLE 2
DIAGNOSTIC ACCURACY ACHIEVED BY NOIDA FOR STATIC AND SEQUENCE-DEPENDENT DEFECTS.

	Static		Sequence-dependent	
Defect type	**Accurate**	**Inaccurate**	**Accurate**	**Inaccurate**
Bridge	3290 (100.0%)	0	1424 (100.0%)	0
Open	449 (100.0%)	0	3510 (94.6%)	201
Transistor	3351 (99.4%)	20	3758 (99.1%)	34

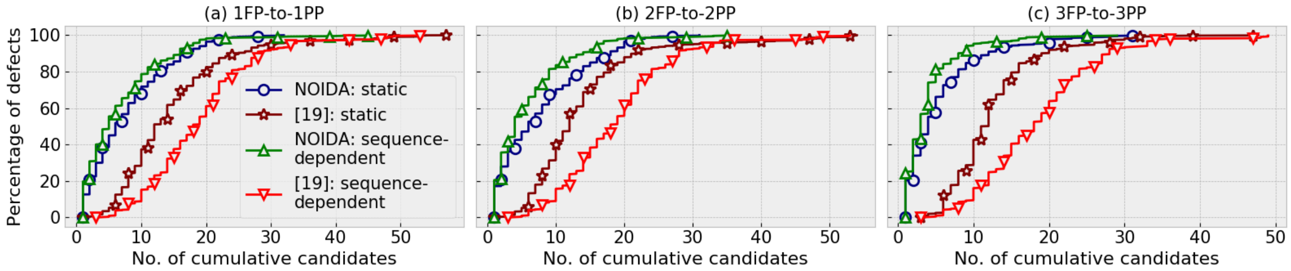

Fig. 5. Cumulative resolution distribution when (a) one (1FP-to-1PP), (b) two (2FP-to-2PP) and (c) three (3FP-to-3PP) cell-level failing patterns are randomly changed to passing patterns. Plot lines marked with "○" and "△" correspond to the resolution obtained by NOIDA for static and sequence-dependent defects, respectively. Plot lines marked with "*" and "▽" correspond to the resolution obtained by [19] for static and sequence-dependent defects, respectively.

and compares it with the resolution achieved by [19]. Fig. 4 contains four plots. Each plot is sorted by the number of candidates. For a plot point x-y, the x-value denotes the number of candidates, and the y-value denotes the percentage of defects y with resolution less than or equal to x. Plots marked with "○" and "△" represent the resolution obtained by NOIDA for static and sequence-dependent defects, respectively. Plots marked with "*" and "▽" is the resolution obtained from the faults derived in [19], for static and sequence-dependent defects, respectively. It is observed that NOIDA returns 12.1% fewer candidates compared to [19]. Moreover, the average resolution obtained from NOIDA is 9.2 candidates per defect, showing an improvement of 1.3 candidates per defect over [19].

However, the resolution improvement achieved by NOIDA is associated with some accuracy loss. When [19] is used for diagnosis, it achieves perfect accuracy. But its important to note here that each defect injected is identical to an instance of the faults extracted in [19] and therefore, it is not at all surprising that the accuracy of [19] is 100.0%.

Another point is that diagnosis here is performed on individual cells, that is, not on a circuit that contains interconnected cells. For circuit-level diagnosis, it is possible that a cell-level failing (passing) pattern can become a circuit-level passing (failing) pattern due to noise. This is because voltage deviation due to noise can change how a driven cell interprets the value. In NOIDA (and other front-end diagnosis approaches [19]–[21], [30], [31]), voltage deviation can also result from inaccurate SPICE models used during defect extraction and analog fault simulation. Thus, a weak logic-1 at a cell output can actually be or interpreted as a weak logic-0 and vice versa, which in turn can transform a cell-level failing pattern to a circuit-level passing pattern. However, a large deviation is required for a strong logic-0 (logic-1) to become a weak logic-1 (logic-0). Thus, for fault models that equate passing with strong logic values, it is less likely for a cell-level passing pattern to become a circuit-level failing pattern. Therefore, to evaluate both NOIDA and [19], "realistic" defect responses are created by randomly changing cell-level failing patterns to passing patterns. Specifically, three new responses are created for each original defect response by changing one, two and three cell-level failing patterns to passing patterns. (Corresponding defect injection experiments will henceforth be referred to as 1FP-to-1PP, 2FP-to-2PP and 3FP-to-3PP.) So as not to modify the behavior of a defect entirely, a defect response is altered only when the decrease in the number of failing patterns is less than or equal to 50%. For instance, defects with less than six failing patterns are not considered for 3FP-to-3PP. Compared to 16,037 defects considered in the first set of experiments (i.e., when original defect responses are used), the number of defects reduces to 10,891 for 1FP-to-1PP, 8,712 for 2FP-to-2PP, and 5,229 for 3FP-to-3PP.

Table 3 shows the percentage of defects accurately diagnosed by NOIDA and [19] when one, two and three failing patterns are changed to passing patterns. The results indicate that the accuracy of [19] remains at 100.0% while the accuracy of NOIDA slightly increases for 1FP-to-1PP. This is because 102 out of 255 defects inaccurately diagnosed earlier had only one failing pattern and are thus not considered for 1FP-to-1PP. For 2FP-to-2PP and 3FP-to-3PP, it is noticed that the accuracy of [19] drops to 96.5% and 94.2%, respectively while NOIDA performs comparatively better, attaining an accuracy of 97.9% and 96.3%.

The improvement in resolution however is more significant. Specifically, Fig. 5 highlights the improvement in diagnostic resolution from NOIDA. Fig. 5 shows three parts, one each corresponding to 1FP-to-1PP, 2FP-to-2PP and 3FP-to-3PP. Each part contains four plots. Plot lines marked with "○" and "△" show the cumulative diagnostic resolution distribution of static and sequence-dependent defects, respectively, reported by NOIDA. Plot lines marked with "*" and "▽" represent the cumulative resolution distribution of static and sequence-dependent defects, respectively, achieved by [19]. It is observed that NOIDA returns 46.3%, 43.6% and 55.8% fewer candidates compared to [19] for 1FP-to-1PP, 2FP-to-2PP and 3FP-to-3PP, respectively. Moreover, NOIDA shows an improvement of 7.5, 7.3 and 9.5 candidates per defect over [19] for 1FP-to-1PP, 2FP-to-2PP and 3FP-to-3PP, respectively. In addition, the plots reveal that NOIDA achieves a perfect resolution for 13.7%, 17.9% and 20.1% of defects for 1FP-to-1PP, 2FP-to-2PP and 3FP-to-3PP, respectively. On the other

TABLE 3
DIAGNOSTIC ACCURACY FOR NOIDA AND [19] WHEN ONE (1FP-TO-1PP), TWO (2FP-TO-2PP) AND THREE (3FP-TO-3PP) CELL-LEVEL FAILING PATTERNS ARE CHANGED TO PASSING PATTERNS.

Diagnosis approach	**1FP-to-1PP**	**2FP-to-2PP**	**3FP-to-3PP**
NOIDA	98.6%	97.9%	96.3%
[19]	100.0%	96.5%	94.2%

hand, less than 1.0% of defects have a resolution of one when [19] is used. Thus, NOIDA performs significantly better than [19] in terms of accuracy and resolution in the presence of noise.

IV. Conclusions

This work presents a novel generalized methodology for front-end defect diagnosis we call NOIDA (NOise-resistant Intra-cell DiAgnosis). NOIDA consists of finding defect locations within a cell and deriving defect behavior based on the nets that surround the suspected defect location, instead of correlating the observed defect response with a particular fault model.

Simulation experiments for over 16,000 intra-cell defects are used to evaluate NOIDA on various standard cells. Results indicate that the approach achieves a resolution improvement of 12.1% when compared to [19] with a slight loss in accuracy. Furthermore, when additional defect injection experiments are performed by adding noise to the tester response, it is seen that NOIDA is able to diagnose 97.6% defects accurately compared to 96.9% by [19]. More importantly, NOIDA performs significantly better than [19] in terms of resolution. Specifically, NOIDA returns 48.6% fewer candidates (8.1 fewer candidates per defect, on average) and achieves a perfect resolution for 17.2% of defects. This reduction in the number of potential defective locations within a cell while achieving near perfect accuracy makes PFA efficient and cost-effective, and likely enhances yield learning significantly.

The experiments presented here include running NOIDA on combinational standard cells with a single defect injected at a time. Current work includes running NOIDA on sequential cells. Future work will be focused on extending this approach to include diagnosis of multiple defects, and deriving a scoring model to rank the candidates for further resolution improvement.

References

[1] H. Tang *et al.*, "Analyzing Volume Diagnosis Results with Statistical Learning for Yield Improvement," in *IEEE European Test Symposium*, May 2007, pp. 145–150.

[2] R. D. Blanton *et al.*, "Yield Learning Through Physically Aware Diagnosis of IC-failure Populations," *IEEE Design Test of Computers*, vol. 29, no. 1, pp. 36–47, Feb 2012.

[3] X. Yu and R. D. Blanton, "Estimating Defect-Type Distributions through Volume Diagnosis and Defect Behavior Attribution," in *IEEE International Test Conference*, Nov 2010, pp. 1–10.

[4] Y. T. Lin and R. D. Blanton, "METER: Measuring Test Effectiveness Regionally," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 7, pp. 1058–1071, July 2011.

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

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

[7] V. Boppana and W. K. Fuchs, "Fault Dictionary Compaction by Output Sequence Removal," in *International Conference on Computer-Aided Design*, 1994, pp. 576–579.

[8] S. Holst and H. J. Wunderlich, "Adaptive Debug and Diagnosis without Fault Dictionaries," in *IEEE European Test Symposium*, May 2007, pp. 7–12.

[9] T. Bartenstein *et al.*, "Diagnosing Combinational Logic Designs using the Single Location at-a-time (SLAT) Paradigm," in *IEEE International Test Conference*, 2001, pp. 287–296.

[10] S. Venkataraman and S. B. Drummonds, "POIROT: A Logic Fault Diagnosis Tool and its Applications," in *IEEE International Test Conference*. IEEE, 2000, pp. 253–262.

[11] J. C. M. Li and E. J. McCluskey, "Diagnosis of Sequence-dependent Chips," in *IEEE VLSI Test Symposium*, 2002, pp. 187–192.

[12] M. Keim *et al.*, "A Rapid Yield Learning Flow Based on Production Integrated Layout-Aware Diagnosis," in *IEEE International Test Conference*, Oct 2006, pp. 1–10.

[13] W. Zou, W.-T. Cheng, and S. M. Reddy, "Bridge Defect Diagnosis with Physical Information," in *IEEE Asian Test Symposium*, Dec 2005, pp. 248–253.

[14] S.-Y. Huang, "Diagnosis of Byzantine Open-segment Faults [Scan Testing]," in *IEEE Asian Test Symposium*, Nov 2002, pp. 248–253.

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

[16] Y.-J. Chang *et al.*, "Experiences with Layout-aware Diagnosis - A Case Study," in *Electronic Device Failure Analysis*, vol. 12, no. 12, May 2010, pp. 12–18.

[17] J. Mekkoth *et al.*, "Yield Learning with Layout-aware Advanced Scan Diagnosis," in *International Symposium for Testing and Failure Analysis*, vol. 32, 2006, p. 412.

[18] R. Desineni, O. Poku, and R. D. Blanton, "A Logic Diagnosis Methodology for Improved Localization and Extraction of Accurate Defect Behavior," in *IEEE International Test Conference*, Oct 2006, pp. 1–10.

[19] F. Hapke *et al.*, "Cell-Aware Test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 9, pp. 1396–1409, Sept 2014.

[20] P. Maxwell, F. Hapke, and H. Tang, "Cell-aware Diagnosis: Defective Inmates Exposed in their Cells," in *IEEE European Test Symposium*, May 2016, pp. 1–6.

[21] H. Tang *et al.*, "Diagnosing Cell Internal Defects Using Analog Simulation-based Fault Models," in *IEEE Asian Test Symposium*, Nov 2014, pp. 318–323.

[22] M. E. Amyeen, D. Nayak, and S. Venkataraman, "Improving Precision Using Mixed-level Fault Diagnosis," in *IEEE International Test Conference*, Oct 2006, pp. 1–10.

[23] E. J. McCluskey and C.-W. Tseng, "Stuck-fault Tests vs. Actual Defects," in *IEEE International Test Conference*, 2000, pp. 336–342.

[24] F. J. Ferguson and J. P. Shen, "Extraction and Simulation of Realistic CMOS Faults Using Inductive Fault Analysis," in *International Test Conference, New Frontiers in Testing*, Sep 1988, pp. 475–484.

[25] Y. Liu and Q. Xu, "On Modeling Faults in FinFET Logic Circuits," in *IEEE International Test Conference*, Nov 2012, pp. 1–9.

[26] H. Tang *et al.*, "Diagnosing Timing Related Cell Internal Defects for FinFET Technology," in *VLSI Design, Automation and Test*, Apr 2015, pp. 1–4.

[27] S. Zachariah *et al.*, "On Modeling Cross-Talk Faults," in *Design, Automation and Test in Europe Conference and Exhibition*, Mar 2003, pp. 490–495.

[28] Nangate Inc., "Nangate 45nm Open Cell Library," *http://www.nangate.com*, 2008.

[29] X. Fan *et al.*, "Extending Gate-level Diagnosis Tools to CMOS Intra-gate Faults," *IET Computers Digital Techniques*, vol. 1, no. 6, pp. 685–693, Nov 2007.

[30] A. Ladhar, M. Masmoudi, and L. Bouzaida, "Efficient and Accurate Method for Intra-gate Defect Diagnoses in Nanometer Technology and Volume Data," in *Design, Automation Test in Europe Conference and Exhibition*, Apr 2009, pp. 988–993.

[31] C. W. Tzeng, H. C. Cheng, and S. Y. Huang, "Layout-based Defect-driven Diagnosis for Intracell Bridging Defects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 5, pp. 764–769, May 2009.

[32] Z. Sun *et al.*, "Intra-cell Defects Diagnosis," *Journal of Electronic Testing*, vol. 30, no. 5, pp. 541–555, 2014.

[33] M. Abramovici, P. R. Menon, and D. T. Miller, "Critical Path Tracing - an Alternative to Fault Simulation," in *Design Automation Conference*, 1983, pp. 214–220.

[34] S. Mittal and R. D. Blanton, "PADLOC: Physically-Aware Defect Localization and Characterization," in *IEEE Asian Test Symposium*, Nov 2017, pp. 212–218.