A variety of yield-learning techniques are essential since no single approach can effectively find every manufacturing perturbation that can lead to yield loss. Test structures, for example, can range from being simple in nature (combs and serpentine structures for measuring defect-density and size distributions) to more complex, active structures that include transistors, ring oscillators, and SRAMs. Test structures are designed to provide seamless access to a given failure type: its size, its location, and possibly other pertinent characteristics. But their downside is that they cannot be sold for profit, hence, they are typically relegated to the wafer scribe lines, or limited to a small number of wafer lots, and are small in size. These characteristics of test structures are also a source of disadvantage. Specifically, because test structures are not extremely large, they cannot mimic the layout diversity actually found in production ICs. Also, because their volume is limited, their “sampling” of the fabrication process is somewhat bounded.

Using production ICs for yield learning, in addition to test structures, is intuitive because they are the target recipients of yield-learning benefits. One of the main challenges in accomplishing this, however, has been the problem of deconvoluting the failing IC’s test response for not only failure localization at the logical (“netlist”) level (which is the conventional objective of diagnosis), but also for defect characterization that involves deriving the type of defect, its size, its three-dimensional physical location in the layout, and possibly the root cause of its manifestation. We (and others) have for a number of years explored the use of test data from the actual production IC for yield learning [1]–[4]. This is not a totally new idea, however, in that past work has proposed the use of production-IC test data to derive valuable information about the IC fabrication process (e.g., [5]–[6]).

For the more recent work, many have used the phrase “volume diagnosis” to emphasize the central role that diagnosis plays. For test structures, the defect type and location are easily obtained since the structure is designed to be very sensitive to a particular defect type and access to the defect-affected location is direct. For example, a comb structure is sensitive to a bridge but is insensitive to an open, and the existence of a bridge can be easily identified by an impedance measurement. Unfortunately, the same characteristics do not generally
hold true for a production IC. In other words, a production IC is designed to implement a customer function, so sensitivity to any particular defect type or location outside the regular structures (memories, scan chains, etc.) is therefore somewhat arbitrary, and thus necessitates the need for diagnosis (i.e., an automated approach to reason about the location and nature of a defect).

Understanding the failure of a single production IC is not sufficient for driving improvements in yield. Multiple diagnoses are needed to understand whether a significant subset of failures has commonalities that imply a systemic problem of some sort, a problem which if identified and/or confirmed through physical failure analysis and then fixed accordingly will increase yield. Thus, the term “volume” in volume diagnosis is meant to imply the derivation of a statistically significant number of diagnoses so that conclusions concerning the existence of a systematic defect can be made.

In the following, we describe work performed in the Advanced Chip Testing Laboratory (ACTL) at Carnegie Mellon University, Pittsburgh, PA, that is focused on volume diagnosis with particular emphasis on its application to yield learning. Specifically, Section II describes the important role of diagnosis and our comprehensive, layout-based approach that we call “physically aware diagnosis” [7]–[9]. Section III describes novel work in IC yield learning through physically aware diagnosis. In particular, Section III-A examines defect density and size distribution (DDSD) extraction [10]–[11], Section III-B overviews systematic defect identification [4], Section III-C presents work in defect type distribution extraction [9], and Section III-D describes design-for-manufacturability (DFM) rule evaluation [12]. Section IV summarizes our contributions and describes on-going work in volume diagnosis. Finally, as already mentioned, other researchers have also conducted extensive work in this area, but because of space limitations, we refer the interested reader to the sample of works cited.

Diagnosis

In a conventional diagnosis, the objective is to identify each (logical) site $s_i$ within a failing IC that is adversely affected by one or more defects. This is typically accomplished using a three-step effect-cause diagnosis that employs some form of path tracing, fault simulation, and ranking. In path-tracing, outputs (primary or scan) that have failed during test serve as starting points for working backwards into the driving logic to identify potentially faulty signal lines. The resulting path-traced sites are then fault simulated to determine which faulty sites produce a simulation response that best mimics what has been produced by the failing IC. Because there can be, and often are, mismatches between a fault simulation response and the actual response of the failing IC, various ranking heuristics have been developed. Alternatively, the subresponse of several faults can be combined together to reduce the amount of mismatch, a technique known as a per-test diagnosis [13].

In the 1990s, layout was used to diagnose specific types of defects. A decade later, we developed a novel approach that used layout in a more comprehensive manner, where a defect type is derived instead, as opposed to being assumed [7]–[9]. Specifically, in a physically aware diagnosis, before ranking or combining sites, each site is further analyzed at the layout level by taking into account the behavior of the transistors and wires that physically surround the site. In particular, the following neighborhood observation about a defect that affects a site $s_i$ is exploited [14]: A defective site $s_i$ is controlled by its surrounding lines. This observation concerning the nature of defects is quite conservative and certainly holds for the variety of defects that are known to affect transistors and the wiring that physically interconnects them. Signal lines that surround a site $s_i$ are called neighbors of $s_i$ and includes all wires that are physically close to $s_i$, that is, the “wires” (metal, vias, polysilicon, etc.) that serve as inputs to the cells that drive the physical neighbors, the physical neighbors themselves, the inputs to the cell that drives $s_i$, and the side-inputs of all the cells that are driven by $s_i$. The layout and logic-netlist descriptions of the circuit are used to automatically identify neighbors. Specifically, for every metal layer where $s_i$ resides, every neighboring line within some distance $d$ is deemed a neighbor. The distance $d$ is based on the fabrication technology; for example, we have used 250 nm for 130 nm designs.

We assume the behavior of a defect is, for the most part, repeatable and consistent with the
neighborhood observation. For static defects (i.e., those that do not depend on the timing or the sequence of test patterns applied), consistency means that the tests for a site $s_i$ that establish the same logical state $Q_{s_i}$ for the neighbors must have the same circuit response in terms of the site’s pass–fail behavior. More formally, consider two test patterns $t_j$ and $t_k$ that sensitize a site $s_i$ and establish the same neighborhood state $Q$. Site $s_i$ is consistent if the status (pass or fail) of the circuit tester response is the same for all pairs of such tests, otherwise, it is inconsistent.

Table 1 illustrates the consistency check for two sites $s_x$ and $s_y$, each of which has neighboring lines $n_1$, $n_2$, and $n_3$. The table shows the eight possible states for the neighborhood. The pass–fail status for a particular state $Q$ is “P” (“F”) if there are one or more test patterns that sensitize a site $s_i$ and establish the neighborhood state $Q$. Site $s_i$ is consistent if the status (pass or fail) of the circuit tester response is the same for all pairs of such tests, otherwise, it is inconsistent.

Table 1 Sites (a) $s_x$ and (b) $s_y$ behaving inconsistently and consistently as a conditional stuck-at fault, respectively.

<table>
<thead>
<tr>
<th>Neighborhood state ($Q = n_1n_2n_3$)</th>
<th>Pass-fail status of $s_x$</th>
<th>Pass-fail status of $s_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>P</td>
<td>F</td>
</tr>
<tr>
<td>001</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>010</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>011</td>
<td>-</td>
<td>P</td>
</tr>
<tr>
<td>100</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>101</td>
<td>P, F</td>
<td>-</td>
</tr>
<tr>
<td>110</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>111</td>
<td>-</td>
<td>F</td>
</tr>
</tbody>
</table>

(a) (b)

Checking the consistency of potential defect sites is a novel and very effective method for improving diagnostic resolution. In both simulation and silicon experiments, we have shown that a significant number of sites can be ruled out as possible defect locations [7]–[9] based on inconsistency, thus significantly improving diagnostic resolution (i.e., reducing the number of sites reported). For example, Figure 1 shows diagnosis results for 2,533 chip failures from the LSI Corporation. The LSI design is a fabrication-characterization vehicle consisting of an array of identical 64-bit ALUs fabricated in 130-nm technology. In this experiment, conventional diagnosis using a commercially-available tool was applied to fail data produced by the 2,533 chips. The result of that analysis produces a list of ranked candidate faults. The dark gray histogram in Figure 1 is the original resolution resulting from the use of the commercial tool, that is, the x-axis is the number of candidate faults reported, and the y-axis reports the number of test-chip failures that exhibited the corresponding resolution. A neighborhood consistency check is then applied to all the candidates using a neighborhood size of 250 nm. Of the 2,533 chips, 9% of the chips (239) did not have any consistent candidate faults. Whether this is a negative or positive result cannot be easily assessed since physical failure analysis (PFA) is too costly to apply to all failing chips. It should be noted however that over 50% of those 239 chips exhibited evidence that they are affected by multiple defects which means it is possible that a significant number, if not all, of the candidates reported are indeed incorrect. PFA however has been applied to five of the failing chips. The results in Table 2 demonstrate that resolution can be improved by the consistency check without degrading diagnostic accuracy. The resolution of the remaining 2,294 chips is shown as the light-gray histogram in Figure 1. (The overlap of the two histograms is shown as black bars.) The original number of candidates (39,842) is significantly reduced (13,219), resulting in a percentage improvement of 67%.

In this experiment, checking neighborhood consistency significantly improves resolution, but at what cost? The average time to perform diagnosis for a modern design can be several minutes, which given typical yield levels, means that all of the failures produced daily can be diagnosed overnight using a moderate number of computer servers.

2The logical state established by a test $t_j$ for a set of signal lines is the logical values driven onto those lines when $t_j$ is applied to the circuit.
Using layout to improve diagnosis is obviously more computationally expensive. However the additional cost is easily amortized by extracting all neighborhood states before diagnosis begins. In this way, the neighborhood states for performing the consistency check and the fault-model extraction (explained next) can be accomplished using straightforward table look-ups.

In addition to the consistency check, another unique characteristic of physically-aware diagnosis is the formal extraction of a customized, failure-specific fault model that captures the behavior of each site \( s_i \) from the states established by test patterns that sensitize \( s_i \) [8]. In model extraction, we do not simply check the behavior of a site against a prede-termined list of fault models but instead we use the test data to drive the derivation of a model that precisely describes the logical misbehavior that has been observed from the failing IC. For example, Table 1(b) shows the behavior of a consistent site \( s_y \).\(^3\) If we treat the “P” entries as maxterms, “F” entries as minterms, and the “-” entries as don’t cares, the table then describes a Boolean fault-activation function for site \( s_y \). Using well-known logic minimization techniques, the truth table can be simplified to a minimal sum-of-products expression. For example, the activation function described in Table 1(b)

\[ f_{s_y}(x) = \sum m(1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30) \]

<table>
<thead>
<tr>
<th>SEM of failure</th>
<th>4-line bridge</th>
<th>Gate-to-gnd short</th>
<th>Poly-to-active short</th>
<th>Open via</th>
<th>Open contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip 1</td>
<td>32</td>
<td>1</td>
<td>4</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>Chip 2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Chip 3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Chip 4</td>
<td>Open via</td>
<td>Open contact</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^3\)Site \( s_y \) is completely consistent since there are no two sensitizing test patterns, where one passes and one fails, that establish the same neighborhood state.

Figure 1. Resolution improvement for 2,294 failing test chips using the consistency check.

Table 2 For these five physically analyzed IC failures, diagnostic resolution is improved or maintained without degrading accuracy.
simplifies to the expression \( n_1 n_2 + n_1 n_2 \), that is, whenever \( n_1 \) and \( n_2 \) are equal, site \( s_y \) fails.

Besides the improved resolution, physically aware diagnosis offers a number of additional advantages. One obvious advantage is that the diagnosis-extracted fault model provides information about the defect’s behavior in addition to its location. Physical localization is also improved however, in several ways. As already mentioned, the consistency check significantly reduces the number of potential defect sites. The minimization process used to derive the defect model (something we represent as a “fault-tuple macro-fault” [15]) also leads to the identification of irrelevant neighbors. For the neighborhood of Table I(b), neighbor \( n_3 \) is proven to have no impact on the faulty behavior exhibited by site \( s_y \) for the test patterns applied. Removal of irrelevant lines from the neighborhood can effectively shrink the size of the neighborhood, resulting in a more precise, physical three-dimensional localization of the defect. The actual neighbors that are found relevant (\( n_1 \) and \( n_2 \) for \( s_y \)) also reveal insight about the nature and location of the defect. For example, a site \( s_i \) that has all its neighbors eliminated except for the inputs of its driving cell can be affected by a front-end defect that affects the driving cell itself; a branch net that exhibits both stuck-at-0 and stuck-at-1 behavior with or without relevant physical neighbors is likely affected by an open; and two faulty nets that are mutual neighbors with opposing values are probably affected by a bridge defect. By using these characteristics of the extracted macro-fault model, we can further refine the physical localization of the diagnosis outcome. Table 3 shows the localization improvement achieved using this approach for 73 LSI chips that had ideal resolution [16]. Thus, physically-aware diagnosis improves localization on two fronts.

First, the number of possible defect locations is significantly reduced based on the consistency check. Second, the physical localization within the layout based on the characteristics of the extracted defect model further reduces the ambiguity of diagnosis. Consequently, the accuracy of volume-diagnosis applications significantly improves because of the resulting refinement in defect localization.

Yield learning

The results of volume diagnosis by themselves are not very useful. It is only when the diagnostic data are statistically analyzed do they reveal actionable information for improving design, manufacturing, and even test itself [17]. In the following, we describe work in the Advanced Chip Testing Laboratory (ACTL) that uses the results from physically-aware volume diagnosis to extract defect density and size distributions (DDSDs), identify systematic defects, derive defect-type distributions, and to perform DFM rule/guideline evaluation.

Defect density and size distribution extraction

While many types of defects lead to yield loss, one important characteristic is systematic versus random defects. While the line dividing the two can be blurry, a “systematic” is a specific defect that occurs repeatedly under a set of conditions, presumably due to some process-design interaction that increases its likelihood significantly. On the other hand, random defects are just that—random. Because a systematic defect occurs repeatedly, due to the same cause, it is important to identify the cause if its impact on yield is significant. Random defects can have many different sources, each with its own sets of remedies. For example, random defects resulting from contaminants are kept under control by ensuring regular cleaning and maintenance of the processing tools/chambers and wafer containers. To have a complete solution for understanding yield loss, both random and systematic defects must be continually dealt with.

We have developed an approach that combines layout information and diagnosis results to extract DDSDs for each metal layer. In addition to scribe-line test structures, this approach offers the additional advantage of using a diverse set of layout geometries from the production IC to measure defect characteristics. Critical area [18] is extracted for a range of defect sizes for multiline bridges identified from the
design layout. Using diagnosis results for bridge-defect failures that are filtered from all chip failures [19], nonlinear optimization is used to find the DDSD values that minimize the difference between the observed diagnosis results and those predicted from a critical-area yield model [18]. Such an approach is appealing for a number of reasons. For example, it allows fabless companies to track/compare DDSDs from their suppliers, and it provides manufacturers an alternative approach to derive these important fabrication parameters.

Various simulation and silicon experiments have been performed to validate the DDSD extraction procedure [10], [11]. Figure 2 compares derived DDSDs with ones that have been injected into the ISCAS85 benchmark circuit c3540 using simulation. Specifically, injected DDSDs and DDSDs derived using linear regression for all metal layers are compared, where the solid line represents the injected DDSD, circles plot the derived DDSD based only on test responses, while the diamonds and squares assume perfect (all cases have resolution of 100%) and non-perfect diagnosis (cases where resolution is less than 5%), respectively. For all cases, the derived DDSD closely matches what has been injected. In addition, DDSDs have been also been derived from actual IC failures. Figure 3 shows the DDSDs extracted for all metal layers from the LSI test chips [10] described in the last section, where the actual values (missing from the y-axis for proprietary reasons) mimic those closely measured by more conventional techniques. Each plotted point is the predicted defect density (y-axis) for a specific range of defect radii (x-axis) for all six metal layers (metal1–metal6) used in the LSI design. The 95% confidence intervals for all the points are quite small except for the smallest defect sizes. The outcome for small defects is very much expected given that they have very little critical area even though such defects are much more likely.

Systematic defect identification

Unlike random defects, the failures due to design-process interactions are systematic in nature. In other words, they can cause failures repeatedly in different ICs and show up in different areas of a design with similar features. If these yield-limiting features can be identified then it is possible their failures can be eliminated through improved process control and/or design alteration. The latter action being most likely for subsequent designs taped out for the same fabrication process.

Figure 4 shows an overview of our approach [4] for identifying layout-based systematic defects. It begins by applying physically-aware diagnosis to a sufficiently large number of IC failures (i.e., volume diagnosis). One outcome of each diagnosis run, as described earlier, is the physical location of every possible failure location which could be further confirmed using inspection data. Layout “snapshots” of
each suspect location are then automatically extracted and converted to pixel-based images. Clustering is applied to all the images resulting from the volume diagnoses to group similar layout snapshots together to identify any commonalities. Clusters that are relatively large are suspected to contain defects that are systematic in nature. To confirm this suspicion, PFA can be performed on several of the ICs that correspond to the large clusters and/or a detailed fabrication simulation (lithography, chemical mechanical polishing, etc.) can be used to determine if the layout features identified by the clusters reveal a deficiency in the fabrication process.

The flow of Figure 4 has been applied to virtual (simulation-based) failures [20] and actual IC failures from both Nvidia and LSI. Figure 5 shows the result from one of these experiments. Specifically, an experiment was carried out using the LSI test-chip failure data. The test chip, fabricated using 130 nm technology, consists of 384 exact copies of a 64-bit ALU design. Each ALU has approximately 3000 gates and is tested within a full-scan environment for 100% stuck-at fault coverage with over 200 tests. Diagnosis of the failing test chips revealed that 106 resulted in singles while the remaining had two or more candidates. In order to remove any ambiguity associated with an imperfect diagnosis, the experiment only focused on the 106 ICs that resulted in a single location.

Table 4 provides details of the clustering experiment that includes the number of clusters and their minimum and maximum sizes for each layer. As expected, the via and contact layers have very few clusters given their limited geometrical diversity. The metal and polysilicon layers, on the other hand, have much more diversity despite the small number of failing ICs (106) examined.

Although the 106 IC failures examined are not statistically significant, we found that two of five SEM images of the physically analyzed failures available to us from LSI (see Table 2) correspond exactly to two of the larger clusters listed in Table 4. Figure 5 compares the SEM (scanning electron microscope) images with representative layout snapshots from the two corresponding clusters. We believe this is strong evidence that our approach can uncover
systematic defects. Similar correlations have been found in our examination of Nvidia GPU chips.

Defect-type distribution derivation

Using the results of physically-aware diagnosis, we have developed a methodology that can effectively estimate the defect-type distribution of a failing-IC population. Such a distribution has a variety of applications. For example, with knowledge of the defect-type distribution affecting a given IC design that is being manufactured in a particular process, fine-grain adaptive testing can be employed to reduce test escapes [17]. Moreover, the defect-type distribution can be used as a quantitative measure in identifying problems in the manufacturing process, thus enabling corrections that improve yield which is our focus in this section.

Our overall approach can be described as follows. First, for each defective site deduced from physically aware diagnosis of a failing IC, the signal lines that are identified to be most relevant to defect activation are used to ascribe a defect behavior to the failing chip. This first step is applied in volume. Next, PFA results that indicate the defect types for a small number of failing ICs (less than 100 provides near 90% accuracy [9]) provide valuable knowledge of the behavior distributions (derived from diagnosis) among the different defect types revealed by PFA. In other words, this information provides the likelihood that a defect of a specific type will exhibit a particular misbehavior. Finally, the defect-type distribution of a failing IC population is estimated by analyzing the actual occurrence rate of each behavior from volume diagnosis along with the knowledge of the behavior distribution of each defect type from PFA. Figure 6 illustrates the flow.

As already mentioned, defect behavior is categorized based on the neighbors deemed to be relevant to defect activation. As described in Section II, the complete set of neighbors for a potential defective site includes: 1) intra- and interlayer lines that are physically close to the defective site (i.e., physical neighbors), 2) standard-cell inputs that drive physical neighbors and the defective site, and 3) the side inputs of standard cells driven by the line corresponding to the defective site. Recall that in the model extraction step within physically-aware diagnosis, the set of neighbors is potentially reduced to a smaller set called the relevant neighborhood, that is, the set of neighbors deemed to control defect activation. Based on the relevant neighborhood, rules have been developed to categorize the behavior [9].

Simulation experiments have shown that the categorization rules are very accurate. Specifically, for hundreds of simulation-based defects created at the layout level [20] that did not exhibit single stuck line (SSL) behavior, 85% of the cell defects, 99% of the bridges, and 99% of the opens were categorized correctly. The error associated with the incorrectly categorized defects all stems from the ambiguity resulting from a nonideal diagnostic resolution.

If each observed defect behavior can be attributed to a specific defect type, then a defect-type distribution can be derived directly from the behaviors resulting from volume diagnosis. However, there is a level of uncertainty when deducing defect types. For example, the misbehavior caused by a defective cell can also be caused by an open or bridge defect. The likelihood that the observed misbehavior could be caused by one of the possible defect types could be partitioned equally among all the defect categories. This, of course, may not match reality however. To address this problem, expectation-maximization [9] is used to estimate defect-type distribution to better cope with the uncertainty inherent to defect classification. Specifically, for

\[ \text{Figure 6. Defect-type distribution derivation based on physically-aware volume diagnosis.} \]
each behavior that cannot be attributed to a specific defect type, the expected defect type for an observed behavior is derived according to 1) the current defect-type distribution, and 2) prior knowledge of defect behaviors which is obtained from the results of failing ICs that have been physically examined. Then, a new defect-type distribution is derived using the expected defect types for each behavior. This process iterates until it converges. More details of this two-step expectation and maximization process employed are given in [9].

Using several benchmark circuits and circuit-level simulation of layout-level defects, we created a virtual population that consisted of more than 17,000 “failing ICs.” Table 5 shows the number and types of defects created for each benchmark.

For each benchmark and each defect type shown in Table 5, 10% of the population is randomly selected to deduce a prior knowledge of the behaviors of different defect types. This step corresponds to the left side of Figure 6. The remaining 90% of the population is then used to derive a behavior distribution (right side of Figure 6) which is then combined with the results of the first step to form the defect-type distribution. Comparisons between estimated defect-type distributions versus the actual distributions are shown in Figure 7. For each circuit, “actual” refers to the actual defect-type distribution created using layout and circuit-level simulation, while “estimate” refers to the defect-type distribution derived using the flow in Figure 6. Figure 7 shows that the estimate closely matches (overall average accuracy is 85%) the actual defect-type distribution for a population consisting of a variety of defects.

DFM rule evaluation

Constraints placed on the design layout to improve yield, collectively known as design for manufacturability (DFM), can be viewed as another form of yield learning. DFM rules and guidelines stem from the analysis of the data generated by various types of test chips that are fabricated in the early development stages of a new manufacturing process. It is a challenge, however, to predict the actual effectiveness of a given set of DFM rules/guidelines for a given design in a given fabrication process. Our work in this area attempts to address this challenge.

Figure 8 illustrates the flow developed to measure the effectiveness of a given DFM rule/guideline. Failure sites resulting from physically-aware volume diagnosis and other randomly-selected layout sites are placed into one of four categories if a given DFM rule/guideline can be or has been applied to the site:

- $X_{a0}$: Site $s_i$ has failed and the DFM rule/guideline was applied to the site;
- $X_{a1}$: Site $s_i$ has failed and the DFM rule/guideline was not applied to the site;
- $X_{b0}$: Site $s_i$ has not failed and the DFM rule/guideline was applied to the site;
- $X_{b1}$: Site $s_i$ has not failed and the DFM rule/guideline was not applied to the site.

The counts associated with each category ($X_{a0}, X_{a1}, X_{b0}, X_{b1}$) are then subjected to various types of statistical analyses to identify which rules/guidelines, if not adhered to, will likely lead to failure. In addition, for those rules/guidelines associated with failure, statistical analysis is used to rank the relative importance of various rules/guidelines. To ensure that there are no confounding factors, such
as critical area, causation confirmation is also employed [12].

The flow of Figure 8 was applied to an Nvidia GPU fabricated in a 90-nm process. We analyzed the effectiveness of 19 rules for this design involving constraints on metal enclosure of vias, metal density, and metal width for various metal layers. The number of failing ICs analyzed is exactly 9,400. For one of the 19 rules examined, the resulting counts for the four categories are: $X_{01} = 163$, $X_{11} = 5,252$, $X_{00} = 75$, and $X_{10} = 1,329$. Figure 9 plots the relative rankings (and associated confidence intervals) of the 19 rules examined. The plot is partitioned into several areas based on their computed association score [12]. The first two rules M5.ENC.2R and M4. ENC.2R exhibit strong association with yield loss and therefore should be enforced based on this analysis given their significant impact on IC yield. The next seven rules (i.e., M5.ENC.1R-M2.ENC.2R) exhibit non-negligible association with yield loss and should be further investigated using more fail data. The remaining rules have negligible association with yield loss and therefore can be safely ignored. The rule rankings reveal that enclosure rules used in this GPU tend to be more effective in the middle metal layers than the lower and upper layers. In addition, the enclosure rules are, in general, more effective than the density rules which suggests that via issues lead to more yield loss than planarity/density problems for this particular process-design combination.

Ongoing analyses of DFM effectiveness based on a flow (like the one shown in Figure 8) will enable designers to better focus their resources for DFM-based yield improvement. In addition, we also believe it is possible to use the results of these analyses to predict DFM needs for the next technology node, which is a topic of current research within ACTL.

**Figure 8. Flow for evaluating the effectiveness of a DFM rule/guideline.**

**Figure 9. DFM rule ranking results for 19 rules applied to an Nvidia GPU. Region A includes rules (x-axis) that are critical to yield, region B includes rules that have substantial impact on yield, and region C has negligible impact on yield.**
removed, (i.e., diagnosis is assumed to be perfect with 100% resolution), the average accuracy of the defect-type distribution derived for the data of Figure 7 improves to 92% [9]. Thus, as we (and other researchers) continue to develop these and other volume-diagnosis applications, the test community should also simultaneously be focused on continually improving the foundation which is diagnosis.

Acknowledgment

The authors would like to acknowledge the design and test data received from various industrial partners that include Phil Nigh from IBM, Bruce Cory from Nvidia, and Bob Madge, Brady Benware, and Chris Schuermyer when they were with the LSI Corporation. They would also like to thank their research sponsors that include the Semiconductor Research Corporation and the National Science Foundation. Finally, they thank Dr. Rao Desineni for the seminal work he performed at Carnegie Mellon University on physically-aware diagnosis and his review of the early versions of this manuscript, and Matthew Beckler for recreating all the graphics appearing in this manuscript.

References


Ronald DeShawn Blanton is currently a Professor with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, where he also serves as the Director of the Center for Silicon System Implementation, an organization consisting of 20 faculty members and over 80 graduate students whose focus is on the design and manufacture of silicon-based systems. His current research interests include the test and diagnosis of integrated, heterogeneous systems and design, manufacture, and test information extraction from tester measurement data. Blanton has a BS in engineering from Calvin College, Grand Rapids, MI, an MS in electrical engineering from the University of Arizona, Tucson, and a PhD in computer science and engineering from the University of Michigan, Ann Arbor. He is a fellow of the IEEE.

Wing Chiu (Jason) Tam research interests are in the areas of CAD for design for manufacturing and IC testing. He is a PhD candidate in the Electrical and Computer Engineering department of Carnegie Mellon University. He is a student member of IEEE and ACM.

Xiaochun Yu has been with Intel Corporation, Hillsboro, OR, working in Diagnosis and Debug Solutions since June 2011. Her research has included multiple defect diagnosis, layout-aware diagnosis, defect-type distribution estimation, and adaptive testing. She has a PhD in electrical and computer engineering from Carnegie Mellon University.

Jeffrey E. Nelson currently works at IBM Corporation in East Fishkill, NY. His research interests include yield learning, DFT, DFM, logic diagnosis, and test generation. Nelson has a BS in electrical and computer engineering from Rutgers University, and an MS and PhD in the same field from Carnegie Mellon University, respectively.

Osei Poku is now working in the area of iOS game development. Poku has a BS in computer engineering from Hampton University, Hampton, VA, and an MS in electrical and computer engineering from Carnegie Mellon University (CMU), Pittsburgh, PA. He is a student member of the IEEE.

Direct questions and comments about this article to Ronald DeShawn Blanton, Advanced Chip Testing Laboratory, Carnegie Mellon University, PA.