DREAMS: DFM Rule EvAluation using Manufactured Silicon

R. D. Blanton, F. Wang, C. Xue, PK Nag, Y. Xue and X. Li
Advanced Test Chip Laboratory (www.ece.cmu.edu/~actl)
ECE Department
Carnegie Mellon University
Pittsburgh, PA 15213-3890

ABSTRACT

DREAMS (DFM Rule EvAluation using Manufactured Silicon) is a comprehensive methodology for evaluating the yield-preserving capabilities of a set of DFM (design for manufacturability) rules using the results of logic diagnosis performed on failed ICs. DREAMS is an improvement over prior art in that the distribution of rule violations over the diagnosis candidates and the entire design are taken into account along with the nature of the failure (e.g., bridge versus open) to appropriately weight the rules. Silicon and simulation results demonstrate the efficacy of the DREAMS methodology. Specifically, virtual data is used to demonstrate that the DFM rule most responsible for failure can be reliably identified even in light of the ambiguity inherent to a non-ideal diagnostic resolution, and a corresponding rule-violation distribution that is counter-intuitive. We also show that the combination of physically-aware diagnosis and the nature of the violated DFM rule can be used together to improve rule evaluation even further. Application of DREAMS to the diagnostic results from an in-production chip provides valuable insight in how specific DFM rules improve yield (or not) for a given design manufactured in particular facility. Finally, we also demonstrate that a significant artifact of DREAMS is a dramatic improvement in diagnostic resolution. This means that in addition to identifying the most ineffective DFM rule(s), validation of that outcome via physical failure analysis of failed chips can be eased due to the corresponding improvement in diagnostic resolution.

1. INTRODUCTION

Simply put, DFM (design for manufacturability) rules are constraints placed on the physical characteristics of a design (i.e., the layout) that are intended to improve yield or some other desired design property. The cost of DFM for a given design can be measured in terms of the additional die area, power consumption, and design time resulting from imposing the rules on the design layout. The payoff in terms of yield is extremely difficult to measure however, especially when rules are applied in varying degrees and in an ad hoc fashion. For example, it is not uncommon to hear from designers that top-priority rules are applied or imposed at a 0.90 adherence rate, second-priority rules are imposed at a 0.60 adherence rate, and low-priority rules are imposed at a 0.40 rate, where adherence rate is the fraction of layout locations where the rule is applicable and actually imposed. More often than not, the applicable layout locations a rule is imposed or not imposed is somewhat arbitrary in nature, not taking into account, for example, if rule adherence at a site A is more advantageous than at site B.

Most approaches for evaluating DFM rules are employed prior to high-volume manufacturing [1-3]. Our methodology, termed here DREAMS (DFM Rule EvAluation using Manufactured Silicon), has the goal of measuring DFM rule effectiveness using information extracted from actual failed ICs. Specifically, DREAMS correlates failed-IC diagnosis results with the design's DFM rule-violation database to identify those rules (if any) that are effective in preventing failure. Unlike previous work [4], we take into account the type of failure that a given rule is meant to guard against, along with precise information concerning the frequency of violations within the design and among the locations reported by diagnosis. These additional insights, along with a custom formulation of the expectation-maximization algorithm [5], improve accuracy dramatically and also, as a by-product, improve diagnostic resolution [6].

This raises an obvious question concerning DREAMS. "How can the results of DFM rule evaluation for an in-production IC be applied to that same IC?" The practical answer is that it cannot be unless one is willing to absorb the additional cost associated with re-designing the layout and taping out the altered design, which of course is exorbitant. But there are several trends in the chip industry that make DREAMS and other similar methodologies worthwhile.

1. Many design houses and foundries commonly fabricate product-like test chips at volumes that sufficiently meet the sample-size requirements of DREAMS. This means that a DFM rule deck can be evaluated using actual product-like layout features, with results on rule importance being fed back to designers for deployment on actual customer designs.
2. For a given technology, it is typically the case that many subsequent designs are launched after the lead product. This is certainly true in the automotive industry where up to a dozen follow-on designs are launched after the lead product. Under this scenario, DFM rule evaluation can be continuously applied to chips 1 through i with the resulting learning applied to chip design $i + 1$.
3. DREAMS can also be used as on-going monitor of the fabrication process. Specifically, monitoring rule-failure rate gives insight into what part of the process has to be tuned since a given rule is concerned with particular features fabricated by specific steps of the manufacturing process (e.g., a via type between two adjacent layers i and i+1).

The rest of this paper is organized as follows. Section 2 gives an overview of the DREAMS methodology, while Section 3 validates the methodology using both simulation- and silicon-based experiments. Section 3 also demonstrates how diagnostic resolution improves as a result of DFM rule evaluation. Finally, Section 4 concludes the paper and provides directions for future research in this area.

2. DREAMS

In this section, the DREAMS methodology for evaluating a set of DFM rules $R = \{r_1, r_2, \ldots, r_K\}$ using a set of diagnosis results $D = \{d_1, d_2, \ldots, d_N\}$ is described in detail, where K is the number of rule-violation types found among the N failed chips that have been diagnosed. The implicit assumption is that the N

978-1-4799-1071-7/13/$31.00 ©2013 IEEE

chips being analyzed have failed due to a single violated rule. So unlike the work in [4], we do not take account that an adhered-rule instance can also cause chip failure. In addition, failures due to random-spot contaminations, systematic defects, etc., are assumed to have been filtered out of the N failed chips being analyzed by DREAMS. For each failed IC, the specific goal of DREAMS is to identify the most likely violated rule responsible for causing failure. Accumulating responsibility data for all N failed chips enables the derivation of the yield-loss contribution for each rule with respect to the population. The overall relative importance of each rule for a given design can then be obtained by combing the yield-loss contribution with the number of violated instances of each rule in the design.

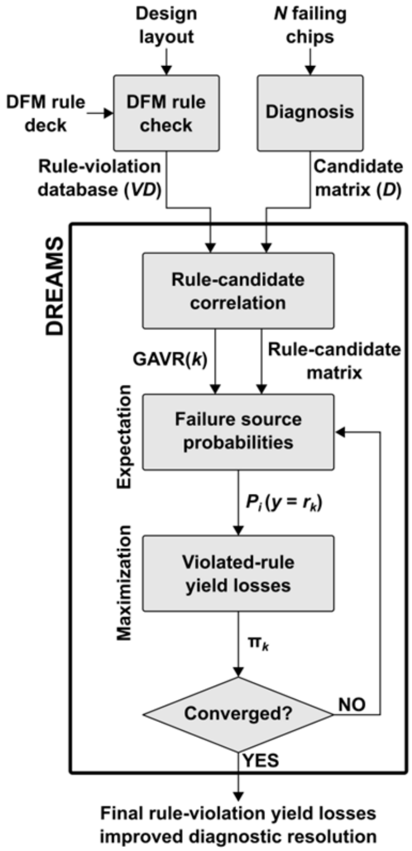
Design layout
N failing chips
DFM rule deck
DFM rule check
Diagnosis
Rule-violation database (VD)
Candidate matrix (D)
DREAMS
Rule-candidate correlation
GAVR(k)
Rule-candidate matrix
Expectation
Failure source probabilities
P_i(y = r_k)
Maximization
Violated-rule yield losses
π_k
Converged?
NO
YES
Final rule-violation yield losses
improved diagnostic resolution

Figure 1: DREAMS flow diagram.

The DREAMS methodology (Figure 1) consists of three steps. In the first step, the rule-violation database VD is correlated with the candidate matrix D. The rule-violation database $VD = \{v_1, v_2, \ldots, v_H\}$ consists of H total violations that exist for all the rules R for the IC design under investigation, where each $v_i \in VD$ is simply a two tuple consisting of a violated rule r_j and its location(s). In this correlation step, the objective is to identify the set of rules violated for each diagnosis candidate associated with each failed-chip diagnosis result $d_i \in D$. Details for rule-candidate matrix construction are given in Section 2.1 Also, in step one, for each rule type r_k, we also calculate the average number of rule violations per net for the entire design. This quantity, denoted GAVR(k), characterizes the DFM properties of the design and is discussed further in Section 2.2. In the second step, a probability is calculated for every rule-violation type that captures the likelihood that it is the source of failure for each failed-chip i. This is the expectation step of our customized EM (expectation-maximization) formulation. Details of this step are provided in Section 2.2. In the third and final step, maximization is performed to calculate the yield loss of each rule-violation type. The two-step EM process is repeated until the yield-loss contributions converge and sum to unity. The EM formulation used in DREAMS is an augmented/customized version of the implementation found within Matlab [7]. The output of the EM analysis is the yield-loss contribution for each rule-violation type. Details of our maximization step are also presented in Section 2.2. In Section 2.3, we describe our initial thoughts on the amount of data needed (i.e., the number of failed chips N) to produce a high-confident evaluation. Finally, as already noted, another significant output of DREAMS is an inherent improvement in the diagnostic resolution. Specifically, for a given diagnosis $d_i \in D$, once the violated rule type r_j responsible for failure has been identified, diagnostic resolution can be improved by eliminating any candidates that do not have rule r_j violated. Details of resolution improvement are presented in Section 2.4.

2.1 Rule-Candidate Correlation

In this section, we describe our approach for identifying all the possible violated rules that could be the source of failure for a given failed chip using diagnosis data [8-10]. Specifically, this first step of DREAMS relies on geometrically matching the possible layout locations identified/implied by diagnosis with known locations of rule violations. Specifically, the objective of this step is to produce a *rule-candidate* matrix, where the rows are failed-chip diagnosis candidates and the columns are DFM rule types. A matrix entry is simply the number of violations for the given rule-candidate combination.

Assumptions: We assume a scan-test chip failure is due to a single DFM rule violation, a "systematic defect", or a random-spot contamination. Occurrence of a random spot defect is a function of critical area and the defect density and size distributions associated with the underlying fabrication process [11]. A systematic defect is less random due to its dependency on both design and fabrication parameter values that were initially unknown at the time of the defect's first manifestation. In other words, it is simply a manufacturability issue that was missed or deemed not worthy to guard against using appropriate DFM rules. A DFM rule violation, in general, is defined as a set of regions (i.e., each region is a bounding box characterized by a pair of coordinates in a particular IC layer) in the layout where the corresponding geometry violates the specified rule. There can be multiple rule types that are applicable to a given IC layer, each meant to prevent one or more defect types. For each type of DFM rule, it is assumed that the violating regions (i.e., the rule-violation instances for the entire layout) have already been computed as part of the normal design flow, each of which resides in the rule-violation database VD.

It is also assumed that each failed chip has been *logically* diagnosed to identify (at the very least) the set of nets (i.e., the candidates) that are the most likely failure sites. In the best case scenario, *physically-aware* diagnosis [8-10] has been utilized which not only provides locations but also a listing of all the most likely defect types that can produce the errors observed on the tester. For either form of diagnosis, layout regions for each candidate can easily be computed from the layout.

Neither logical nor physically-aware diagnosis is perfect however, meaning that it is frequently the case that not all of the tester response of a failed chip can be predicted/explained completely by the candidates identified by diagnosis. Even if the tester response can be completely accounted for, it is often the case that multiple candidates are reported due to the fact that various defects at various locations can cause the same unexpected behavior observed by the tester. Typically, for each diagnosis candidate a score is reported that represents how closely it can predict/explain the entire tester response produced by the corresponding failed IC. In physically-aware diagnosis, the results include the defect type (open, bridge, cell-level defect, etc.) for each candidate, and the layout location(s). The score value and defect type are used to filter rule-violation instances associated with candidates. In summary, diagnosis results for a failed IC is represented as a matrix or table, where each row describes a candidate in terms of net and/or its possible defect types, their locations (layers and bounding boxes), and scores. This table is referred to as the *candidate matrix D*.

Correlation: Correlating rule-violation instances to failed-chip diagnosis candidates consists of two steps: First, finding intersections and overlaps of the layer geometry of the candidate locations and the rule-violation locations, followed secondly by filtering the data for noise reduction.

Intersection/overlap: A detailed geometric analysis is performed using the candidate matrix D and rule-violation database VD. Specifically, for each failed IC, the information associated with each of its candidates is augmented to include the count and type of its associated rule violations. A rule-violation instance is associated with a candidate if any of its bounding boxes intersects/overlaps with the layout locations of the net and/or defects associated with the candidate. Specifically, the analysis determines if bounding boxes of an implicated section of geometry associated with a candidate overlaps or intersects any region corresponding to a rule violation. Because the number of rule-violation instances can be significant, a scan-line technique [12-15] is used to identify all intersections and overlaps.

Noise reduction: Two techniques are used to reduce the "noise" inherent in the rule-candidate matrix. The first method reduces the size of the candidate matrix D by only considering the highest-ranked candidates that are associated with at least one rule-violation instance. The second method compares the objective of the rule type with the possible defect types identified by physically-aware diagnosis. If it so happens that a rule-violation type cannot cause the defect type reported by diagnosis (e.g., a redundant-via rule violation cannot cause an observed bridge defect), then the corresponding violation is not associated with the candidate. This analysis is easily automatable and is quite invaluable for reducing the noise typically associated with both diagnosis and the rule-violation database. For example, a candidate corresponding to a long net may traverse many layers with a limited number of vias and long runs of metal in each layer, leading to an overwhelmingly large number of rule-violation types and instances. Such a scenario, if not carefully handled, can cause the EM algorithm to produce spurious results.

Table 1 illustrates the type of information contained in the rule-candidate matrix for a single failed chip. Specifically, it shows that diagnosis of this failed chip i produced three, top-scoring candidates ($c_{i,1}$, $c_{i,2}$ and $c_{i,3}$), each of which has at least one rule-violation instance. In general, the matrix would contain candidates from N failed chips. Because the example IC design has five rules, each candidate, in theory, is susceptible to five different rule-violation types. Some rule violations, for a number of possible reasons, cannot affect a given candidate however. For example, candidate $c_{i,1}$ has zero violations as shown in Table 1 for rule r_3 because (i) it has no r_3 violations only adherences, (ii) c_1 does not have any net segments residing in layout regions where r_3 is applicable, or (iii) the type of defect that can be caused by an r_3 violation is incompatible with the defect types associated with candidate $c_{i,1}$. Candidate $c_{i,3}$, on the other hand, has violations for all five rules, and for rule r_5 it actually has three distinct instances of violations. Finally, candidate c_2 is present in Table 1 to explicitly illustrate how incompatibility between the defect type(s) that can result from an r_2 violation and the defect type(s) derived from diagnosis can eliminate a violation from consideration.

Diagnosis candidates	Rule-violation types				
	r_1	r_2	r_3	r_4	r_5
$c_{i,1}$	1	2	0	2	1
$c_{i,2}$	0	~~2~~1	0	3	0
$c_{i,3}$	1	1	1	1	3

Table 1: Example of a rule-candidate matrix for the diagnosis results obtain from a single failed chip i: Each row is a rule-candidate vector $l_{i,j}$ for chip i with diagnosis result d_i and candidate $c_{i,j}$. Row-entries are integers that indicate the number of times a rule type was violated for the candidate. Also, a shaded entry indicates violation(s) that are incompatible with the defect type(s) identified by physically-aware diagnosis.

2.2 Expectation-Maximization

Similar to our previous work in [4], we use the expectation-maximization (EM) algorithm to identify the most likely source of failure for each diagnosed chip. In [4], the unprocessed rule-candidate matrix data for each chip (see Table 1 for an example) is simplified by ignoring multiple instances of a given violation. In other words, the $j_i \times K$ matrix T_i for a given failed-chip diagnosis $d_i \in D$ (where j_i is the number of candidates for d_i) is reduced to a $1 \times K$ vector V_i. More specifically, vector V_i is obtained by converting each entry to binary and then disjunctively combining the resulting contents of each column of T_i. For example, applying this approach to Table 1 produces the vector shown in Table 2. Essentially, the rule-chip correlation is assigned a '1' when at least one diagnosis candidate violates the rule. On the other hand, the index is assigned a '0' when none of the candidates violates the corresponding rule. However, this simplified matrix cannot sufficiently represent the significance or amount of violation since only two values are possible (i.e., 0 or 1). In other words, a rule that has a greater number of rule-violation instances should be distinguishable from a rule that has a much lower number of instances.

Rule-violation types				
r_1	r_2	r_3	r_4	r_5
1	1	1	1	1

Table 2: Simplified rule-candidate matrix derived from Table 1 based on the approach in [4].

In light of this, we propose to formulate an EM algorithm that directly uses knowledge of all the rule-violation instances. Specifically, we preserve the rule-candidate matrix that has a row for each failed-chip candidate as illustrated in Table 1.

One important form of data embedded within the rule-candidate matrix is the Local Average Rule-Violation Rate (LAVR($i,\ k$)). The LAVR($i,\ k$) is the average number of rule-violation instances for the kth rule among all candidates for a single failed chip i which is calculated as:

$$\mathrm{LAVR}(i,k) = \frac{\sum_{j=1}^{j_i} l_{i,j,k}}{j_i}, \qquad (1)$$

where j_i represents total number of diagnostic candidates from the ith failed chip and $l_{i,j,k}$ is the number of kth rule-violation instances corresponding to the jth candidate of the ith failed chip. For example, in the case of Table 1, the LAVR($i,$ 4) value for r_4 is (2+3+1)/3=2. A related parameter derived from the design is the Global Average Rule-Violation Rate (GAVR(k)) for the kth rule which is calculated as:

$$\mathrm{GAVR}(k) = \frac{\sum_{p=1}^{P} l_{p,k}}{P}, \qquad (2)$$

where P is the total number of nets in the design and $l_{p,k}$ is the number of kth rule-violation instances corresponding to the pth net. Intuitively, the GAVR(k) is the average occurrence of rule-violation instances per net for the entire design. The GAVR(k) can be determined by interrogating the rule-violation database, or estimated from the violations for all candidates. GAVR(k), as opposed to LAVR($i,\ k$), describes the general rule-violation statistics for the design. One important insight concerning these parameters is that LAVR(i, k) will likely deviate from GAVR(k) if the kth rule is responsible for more than its expected number of failures. The intuition that substantiates this insight can be described as follows. Diagnosis candidates from all failed chips fall into two categories. The first category consists of ‘correct candidates’, i.e., those that correspond to the actual failure. The second (opposite) category is associated with diagnosis noise, i.e., the incorrect candidates. For the first category, the rule causing failure must be violated. For the second category, rule violations will follow the trend captured by the corresponding GAVR(k). Category-1 candidates will have their corresponding LAVR($i,\ k$) deviate from GAVR(k). Sensitivity to an LAVR($i,\ k$)–GAVR(k) deviation is quantitatively encoded in our EM framework.

The rule-candidate matrix can be improved further by taking into account the severity of the rule violation. Rule-violation databases typically report the level of severity associated with a rule violation. For example, for the industrial data analyzed in this work, rules meant to prevent bridge defects have a severity range, where a larger value indicates a more severe violation. Although not yet exploited in DREAMS, this information can be easily normalized and used to additionally weight rule-violation instances associated with failed-chip diagnosis candidates.

As already mentioned, each failed chip is assumed to be the result of a single rule-violation instance. The associated rule type is referred to as the source of failure for this given failed chip. Failures due to other sources (e.g., random-spot contaminations, systematic defects, etc.) are assumed to have been already removed from the population of N failed chips using existing approaches [17-19]. This means that the failed-chip population can be distributed among the K rules. Once the yield-loss contributions of the K rule types are determined, rankings and relative importance values can then be computed, enabling corrective actions by the manufacturer and/or the designer.

Let $\pi_k = P(y = r_k)$ denote the overall yield-loss contribution for rule r_k with respect to the N failed-chip population, where y represents the failure source, $0 \le r_k \le 1$ for $1 \le k \le K$, and $\pi_1 + \pi_2 + \ldots + \pi_k + \pi_{k+1} + \ldots + \pi_K = 1$. To determine the yield-loss contributions for all K rules $\{r_1, r_2, \ldots, r_k, r_{k+1}, \ldots, r_K\}$, a custom formulation of the EM algorithm is used. EM is a two-step iterative method that derives the model parameters that maximizes the likelihood of the data observed. Before describing the EM customization used in DREAMS, we list our assumptions:

1. A failed chip i only has one failure source, a single violated rule r_k.
2. Among all the diagnostic candidates derived for a failed chip i, only one is the correct candidate CC_i.
3. GAVR(k) can be calculated from the design or estimated using all of the diagnosis candidates.
4. LAVR($i,\ k$) will deviate from GAVR(k) if r_k is the failure source.
5. Candidates that have no associated rule violations are assumed to be incorrect.

Given these assumptions, the EM algorithm is formulated using the following two steps:

Expectation: For each rule, this step estimates the probability that r_k is the source of failure for the ith failed chip, i.e., $P_i(y = r_k)$. It begins by calculating the probability for a candidate to be correct given that rule r_k is the failure source:

$$P(CC_i = c_{i,j} \mid y = r_k) = \begin{cases} 0, & if \quad \sum_j l_{i,j,k} = 0 \\ \dfrac{l_{i,j,k}}{\sum_j l_{i,j,k}}, & otherwise \end{cases}, \qquad (3)$$

where $c_{i,j}$ represents the jth candidate of failed chip i. The intuition behind Eq. 3 is quite straightforward in that the probability for each candidate is calculated based on the extent of rule violation, given the source of failure is r_k. If rule r_k is not violated for any of the candidates, the conditional probability is simply zero. Next, the joint probability of the correct candidate and the failure source is calculated:

$$P(CC_i = c_{i,j}, y = r_k) = \pi_k \times P(CC_i = c_{i,j} \mid y = r_k)\,. \qquad (4)$$

Eq. 4 includes the overall yield-loss contribution π_k for rule r_k, which is iteratively updated by the maximization step. The marginal probability for each candidate is calculated by summing the individual probabilities for all rules:

$$P(CC_i = c_{i,j}) = \sum_{k=1}^{K} P(CC_i = c_{i,j}, y = r_k)\,. \qquad (5)$$

The marginal probability for candidate $c_{i,j}$ from Eq. 5 essentially captures the importance of a particular candidate, or in other words, how likely $c_{i,j}$ is the correct candidate.

At this point, the conditional probability $P(y=r_k \mid CC_i = c_{i,j})$ is calculated based on the LAVR(i, k) and GAVR(k) deviation. If LAVR(i, k) is higher than GAVR(k) for a rule r_k, then r_k is deemed more likely to be the failure source. On the other hand, if LAVR(i, k) is smaller than GAVR(k), the rule is deemed less likely to be the source of failure. The following weighting function is used to capture the LAVR(i, k)–GAVR(k) deviation:

$$W(y=r_k \mid CC_i = c_{i,j}) = \frac{1}{1+\dfrac{\mathrm{GAVR}(k)}{\mathrm{LAVR}(i,k)}}. \quad (6)$$

The conditional probability is obtained by normalizing the weighing function across all the rules:

$$P(y=r_k \mid CC_i = c_{i,j}) = \begin{cases} 0, & \text{if } \sum\limits_k W(y=r_k \mid CC_i = c_{i,j}) \times b_{i,j,k} = 0 \\ \dfrac{W(y=r_k \mid CC_i = c_{i,j}) \times b_{i,j,k}}{\sum\limits_k W(y=r_k \mid CC_i = c_{i,j}) \times b_{i,j,k}}, & \text{otherwise} \end{cases}, \quad (7)$$

where

$$b_{i,j,k} = \begin{cases} 0, & \text{if} \quad l_{i,j,k} = 0 \\ 1, & \text{otherwise} \end{cases}. \quad (8)$$

Now the marginal probability for each rule of ith chip can be calculated as:

$$P_i(y=r_k) = \sum_{j=1}^{j_i} P(CC_i = c_{i,j}) \times P(y=r_k \mid CC_i = c_{i,j}). \quad (9)$$

Maximization: The overall yield loss contribution π_k for rule r_k is calculated as follows:

$$\pi_k = \frac{\sum\limits_i P_i(y=r_k)}{\sum\limits_k \sum\limits_i P_i(y=r_k)}. \quad (10)$$

The expectation and maximization steps are repeated iteratively until convergence. As already mentioned, for a given failed chip i, $P_i(y=r_k)$ is the chip-level probability that rule r_k is the source of failure. Therefore, the "label" for failed chip i is simply $argmax\{P_i(y=r_k)\}$. π_k is a global probability that describes the general features of each rule r_k. The rule with the maximum π_k is essentially the rule that causes the most yield loss for the failed-chip population. Finally, the relative importance of a rule r_k, with respect to a given design, is computed as π_k/GAVR(k) and is proportional to the rule-violation failure rate of rule r_k.

2.3 Confidence Interval Estimation

For a given rule-candidate matrix, our EM formulation will report a set of yield-loss contributions (i.e., { π_1, π_2, …, π_K}). A question that naturally arises centers on the fidelity of the values reported. In other words, one typically wants to know the confidence intervals for the derived model parameters. This is an important problem that we plan to address in the near future. Here we only introduce the general flow of the confidence interval estimation algorithm, which comprises of two steps:

1. Abstract the provided data-set distribution, and generate perturbed rule-candidate matrices.
2. Perform EM using the perturbed matrices in order to estimate the confidence interval according to the algorithm output statistics.

2.4 Diagnostic Resolution Improvement

The results from EM can be easily used to improve diagnostic resolution. For a particular failed chip i, $P_i(y=r_k)$ represents the contribution of each rule to the failure of chip i. Intuitively, if π_k is significant, it is then likely that r_k is the source of failure. On the other hand, if the contribution is small, the corresponding rule will have little likelihood to cause failure.

One can use $P_i(y=r_k)$ to improve diagnostic resolution. Specifically, for failed chip i, the most significant violated rule type r_m is determined using $argmax\{P_i(y=r_k)\}$ and deemed the source of failure of the ith chip. Any candidates without r_m violated are deemed to be incorrect and eliminated. For example, in Table 1, if r_5 is deemed the failure source, candidate $c_{i,2}$ can be removed because r_5 is not violated anywhere along $c_{i,2}$. In this case, a 33% improvement in diagnostic resolution is achieved.

3. EXPERIMENTS

We describe two experiments for demonstrating the viability of DREAMS: one is simulation-based and another that uses actual silicon design and failure data.

3.1 Virtual Data

A substantial number of virtual data sets are generated to examine how the accuracy of DREAMS changes with N (i.e., the number of failed chips), diagnosis resolution, and the global average rule-violation rate (GAVR). A viable DFM rule evaluation method should identify the most important rule given a sufficient number of failed chips and despite any significant variances in resolution and GAVR. In the simulation-based experiment, virtual data sets are generated using a range for N that varies from 20 to 1000 failed chips, a range for failed-chip diagnostic resolution that varies from 1 to 16 candidates, and finally a range for rule-violation yield loss that varies from 0.10 to 1.00. Recall that a rule-violation type r_k is said to have a yield-loss contribution π_k (where $0 \le \pi_k \le 1$) if a violated instance of r_k is the source of failure for $N\times \pi_k$ failed chips. For each data set, a single rule is randomly selected to have the highest yield loss, that is, $\pi_k > \pi_j$, $\forall k \neq j$. The remaining failed chips ($N - N\times\pi_k$) have a failure source that is randomly-selected from the remaining K–1 rules such that $N\times\pi_1 + N\times\pi_2 + \ldots + N\times\pi_K = N$. For nearly equal values of GAVR, a rule with the highest yield loss should be relatively easy to identify, assuming that diagnosis is accurate. On the other hand, a rule with a relatively low yield loss and a high GAVR is quite likely to confound the DFM rule evaluation. In order to mimic the industrial design and diagnosis data we have in hand (discussed in detail in Section 3.2), the number of rules examined in the virtual experiments is chosen to be 20, and the GAVR is varied between 0.05 and 0.5. As described in Section 2.2, we compute the yield-loss contributions for each rule r_k as π_k and compare it to the known values established in each virtual data set. The results produced by DREAMS for a given virtual data set is considered accurate when the rule with the highest computed π_k is the same as the rule selected to have the highest yield loss.

We compare DREAMS with the approach of [4] using three different GAVR distributions. For each distribution, the GAVR of certain rules are either "high" (0.5) or "low" (0.05). Table 3 summarizes the results: Table 3(a) has a single rule that has high GAVR and a low yield loss, 18 rules have low GAVR and a low yield loss, and the one remaining rule has a low GAVR with a high yield loss. This virtual data set captures the situation where some rule has little impact on failure but is significantly violated throughout the design. For Table 3(b), all rules have a low GAVR and a single rule has a high yield loss. Finally, for Table 3(c), the rule with the highest yield loss has a high GAVR while the remaining 19 rules have low yield loss and low GAVR.

	Diagnostic resolution			
Failed chip count *N*	**1**	**4**	**8**	**16**
20	0.20(+0.00)	0.40(+0.50)	0.50(+0.50)	0.50(+0.50)
50	0.10(+0.00)	0.40(+0.60)	0.30(+0.70)	0.30(+0.70)
100	0.10(+0.00)	0.30(+0.70)	0.20(+0.80)	0.30(+0.70)
200	0.10(+0.00)	0.30(+0.70)	0.30(+0.70)	0.20(+0.80)
500	0.10(+0.00)	0.30(+0.70)	0.30(+0.70)	0.20(+0.80)
1000	0.10(+0.00)	0.30(+0.70)	0.30(+0.70)	0.30(+0.70)

(a)

	Diagnostic resolution			
Failed chip count *N*	**1**	**4**	**8**	**16**
20	0.10(+0.20)	0.30(+0.10)	0.40(+0.10)	0.70(+0.30)
50	0.20(+0.00)	0.30(+0.00)	0.30(+0.10)	0.50(+0.40)
100	0.10(+0.00)	0.20(+0.00)	0.30(+0.00)	0.30(+0.30)
200	0.10(+0.00)	0.20(+0.00)	0.20(+0.00)	0.30(+0.00)
500	0.10(+0.00)	0.20(+0.00)	0.20(+0.10)	0.20(+0.10)
1000	0.10(+0.00)	0.10(+0.00)	0.10(+0.00)	0.20(+0.00)

(b)

	Diagnostic resolution			
Failed chip count *N*	**1**	**4**	**8**	**16**
20	1.00(–0.20)	0.90(–0.10)	1.00(–0.30)	1.00(–0.20)
50	0.80(–0.10)	0.80(+0.10)	1.00(–0.30)	1.00(–0.30)
100	0.60(+0.00)	0.60(+0.00)	0.60(+0.10)	0.60(+0.00)
200	0.60(–0.10)	0.50(+0.10)	0.50(+0.10)	0.70(+0.10)
500	0.60(–0.10)	0.50(+0.00)	0.50(+0.00)	0.50(+0.00)
1000	0.50(+0.00)	0.50(+0.00)	0.50(+0.00)	0.50(+0.00)

(c)

Table 3: Each table entry is the minimum π_k required by DREAMS to identify the corresponding rule most responsible for yield loss for three different distributions. The numbers in parentheses are the differences resulting from a comparison with the approach described in [4].

The column labels of Table 3 correspond to different levels of diagnostic resolution, i.e., number of candidates per failed chip. The row labels correspond to the number of chips (N) in the failed-chip population. Each table entry is the minimum yield loss (π_k) required by DREAMS to identify the rule most responsible for yield loss within the population. Ranging from 1.00 (i.e., all failures are caused by some rule r_k) down to 0.10 (i.e., only 1/10 of all failures are caused by some rule r_k), the experiment checks whether the correct rule r_k is identified by DREAMS for reductions of 0.10 in the value of π_k. For example, a table entry of 0.80 indicates that DREAMS identifies the correct rule when its corresponding yield-loss value π_k is at least 0.80. The numbers within the parentheses are the differences compared with the method used in [4]. For example, a table entry of 0.20(+0.10) means that the method of [4] requires at least a yield loss of 0.30 to identify the correct rule, while DREAMS only needs 0.20. Each table entry is based on five different virtual populations, each of which has the corresponding failed-chip count and resolution, but with slightly different values for yield loss and GAVR that stems from our data-generation process. The actual π_k value reported in a table entry is the minimum yield loss agreed upon by three of the five evaluations performed.

As can be observed from Table 3(a), when resolution is poor ($\geq$ 4), a rule with high GAVR and low yield loss confounds the EM formulation of [4]. In other words, the formulation of [4] only identifies the rule most responsible for yield loss when its corresponding π_k is extremely high (i.e., out of 18 table entries, 17 are π_k=1.00 and the reaming case is π_k=0.90). DREAMS, on the other hand, correctly identifies the culprit rule for much lower π_k values since its EM formulation accounts for the discrepancy that may exist between GAVR and LAVR for each rule.

Tables 3(b) and 3(c) summarize the result of the second and third distributions, respectively. DREAMS performs equally well or better as the method in [4] for the second distribution (Table 3b), and when population size is > 100 for the third distribution (Table 3c). For the third distribution, since the rule most responsible for failure has a high GAVR, the ratio LAVR/GAVR is very close to 1. So DREAMS requires more chips (>100) to identify the correct rule. Overall however DREAMS provides a more reliable evaluation of DFM rule effectiveness than the EM formulation described in [4] since one cannot know a priori what type of distribution one has of GAVR(k) and π_k values within a failed-chip population.

3.2 Silicon Data

DREAMS is applied to the diagnosis and design data of an in-production 55-nm automotive-control IC. The IC contains about 7.5 million transistors that correspond to about 20 different analog modules in addition to digital circuits. It utilizes five metal layers for the main circuits, and one additional metal layer for power distribution and is approximately 64mm^2. Out of the 1,202 failed ICs, 1,119 have candidates and 1,038 have actionable diagnosis results. DFM rule-violation data consisted of the *x*-*y*-*z* locations (represented as polygons) for 20 rules that included: two contact rules, one polysilicon rule, five metal rules, and 12 via rules (three each for the four via layers). All the metal rules are designed to prevent bridge defects, while the remaining rules are meant to prevent opens. The two contact rules address the number of contacts and the overlap of polysilicon, whereas the three via rules deal with the number of vias and their corresponding metal overlaps. Commercial diagnosis reports, on average, 30 diagnosis candidates for each failed chip. Using only the top-10 ranked candidates, the rule-candidate matrix is derived using the procedure described in Section 2.1. The GAVR(k) for each rule r_k is estimated using all diagnosis candidates and is reported in the first two columns of Table 4.

Recall from Eq. 1 that GAVR(k) for rule r_k is equated to the total number of r_k violations divided by the total number of signal lines in the design. As already mentioned, we estimate GAVR(k) by using the average rule-violation count for all diagnosis candidates (regardless of ranking) from all failed chips (second column of Table 4). Table 4 reveals that some rules have significantly higher GAVR than other rules, reflecting the fact some optional DFM rules are frequently not adhered to. This characteristic adversely affects the capability of a conventional EM formulation as demonstrated in Section 3.1. After GAVR(k) is estimated for each rule, candidates that have a low diagnostic score or no rule violations are discarded. The resulting candidate matrix D and the rule-violation database (VD) are given as input to DREAMS, and the computed yield-loss contribution π_k for each rule is shown in the third column of Table 4. The last column of Table 4 is the yield-loss contribution divided by the global average violation rate (π_k/GAVR (k)). This value is proportional to the rule-violation failure rate and represents the relative importance of a rule for a given design. From Table 4, it can be observed that rules "M1X.S.1_dfm.a" and "M4X.S.1_dfm.a" have importance values that are 2X higher than the other 18 rules.

Rule name	GAVR(k)	π_k	π_k/GAVR (k)
VIA4X_dfm.a	0.36	0.0095	0.0268
CO_dfm.a	3.53	0.0859	0.0243
PO.W.1_dfm	442.81	0.0282	0.0001
M3X.S.1_dfm.a	3.58	0.1048	0.0293
VIA1X_dfm.a	3.68	0.0891	0.0242
M2X.S.1_dfm.a	2.17	0.0681	0.0314
M1X.S.1_dfm.a	1.25	0.0750	0.0599
M4X.S.1_dfm.a	1.39	0.0747	0.0538
VIA3X_dfm.a	0.61	0.0145	0.0239
CO_dfm.b	53.64	0.0614	0.0012
VIA3X.EN.1_dfm.a	1.21	0.0284	0.0236
VIA2X_dfm.a	2.02	0.0334	0.0165
VIA3X.EX.1_dfm.a	0.38	0.0079	0.0209
VIA4X.EN.1_dfm.a	0.71	0.0188	0.0267
VIA2X.EN.1_dfm.a	3.99	0.0652	0.0164
M5X.S.1_dfm.a	0.85	0.0267	0.0315
VIA2X.EX.1_dfm.a	1.43	0.0244	0.0170
VIA1X.EX.1_dfm.a	2.32	0.0548	0.0237
VIA4X.EX.1_dfm.a	0.23	0.0060	0.0257
VIA1X.EN.1_dfm.a	5.71	0.1231	0.0216

Table 4: Yield-loss and relative rule-importance values calculated by DREAMS for 20 DFM rules for an in-production chip. Rule "M1X.S.1_dfm.a" and rule "M4X.S.1_dfm.a" have importance values that are 2X higher than the others.

3.3 Diagnostic Resolution Improvement

The result of DFM rule evaluation can be further utilized to improve diagnostic resolution. An output of DREAMS is the calculation of $P_i(y{=}r_k)$ which is the probability that an r_k rule-violation instance is the source of failure for failed-chip i. The rule type with the highest $P_i(y{=}r_k)$ value is the predicted source of failure for chip i. Diagnostic candidates that violate this rule are deemed the more likely sites of failure. By discarding candidates that do not violate the rule, diagnostic resolution is improved. Experiments to demonstrate this improvement in resolution are performed using the virtual data described in Section 3.1. Tables 5(a) and 5(b) show the resolution improvement and accuracy, respectively, using a virtual data set based on each rule having GAVR=0.05 and one rule having yield loss of π_k=0.70 with the remaining 19 rules each having an approximate yield loss of π_j=0.30/19. Resolution improvement is defined to be the ratio of the reduced set of candidates to the original set of candidates. Diagnostic accuracy is the percentage of the failed-chip population that still includes a diagnostic candidate that corresponds to the known failure site. From Table 5 it can be observed that the diagnosis resolution is significantly improved while still maintaining accuracy. For example, for N=500 and original resolution of 12, resolution is improved by (1–0.18)% = 82% and the corresponding accuracy is high at 73%. It should be noted that resolution and accuracy can be easily traded off by employing a user-specified threshold for the probability $P_i(y{=}r_k)$. Resolution improvement is also calculated for the industrial data discussed in Section 3.2. The resolution improvement achieved for the 1,038 chips is significant at 69%.

	Original diagnostic resolution					
Failed chip count N	**2**	**4**	**6**	**8**	**12**	**16**
20	0.55	0.33	0.23	0.26	0.16	0.18
50	0.57	0.32	0.25	0.21	0.18	0.16
100	0.56	0.32	0.26	0.21	0.17	0.15
200	0.56	0.35	0.26	0.22	0.18	0.16
500	0.56	0.35	0.26	0.23	0.18	0.16
1000	0.56	0.34	0.27	0.22	0.17	0.16

(a)

	Original diagnostic resolution					
Failed chip count N	**2**	**4**	**6**	**8**	**12**	**16**
20	95%	75%	75%	80%	75%	65%
50	86%	76%	76%	78%	78%	70%
100	85%	83%	74%	76%	76%	71%
200	85%	80%	77%	79%	77%	72%
500	84%	80%	76%	77%	73%	71%
1000	86%	80%	78%	76%	74%	69%

(b)

Table 5: (a) Diagnostic resolution improvement and (b) accuracy using DREAMS, where yield loss for one rule is 0.70, yield loss for the remaining 19 rules is 0.30/19, and GVAC=0.05 for all 20 rules.

3.4 CPU and Memory Requirements

The three-step flow of DREAMS illustrated in Figure 1 has modest compute and memory requirements. For example, only two seconds of desktop CPU time is needed for EM convergence for the 20×7662 rule-candidate matrix corresponding to the N=1,038 failed chips considered in Section 3.2. The more compute-intensive step is the construction of the rule-candidate matrix. As described in Section 2.1, matrix construction requires geometrical analysis of the design layout and thus requires more compute resources. For a laptop with a 2.7GHz quad-core and 16GByte memory, several hours of runtime was consumed to construct 20×7662 rule-candidate matrix. The time for matrix construction will obviously grow with N but we argue that the time is of little consequence since: (i) the matrix-construction task is highly parallel and thus can be easily distributed to multiple cores/machines; (ii) it is likely that an actual rule-violation database is indexed, making rule-candidate correlation much more efficient; and (iii) it is likely the case that rule-candidate matrix will be constructed incrementally as a simple follow-on to conventional diagnosis procedures.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a comprehensive approach for systematically evaluating a set of DFM rules using already-available design data (i.e., the rule-violation database) and diagnosis results from a population of failed ICs. Experiments involving both virtual and silicon/design data has demonstrated the efficacy of the approach we call DREAMS (DFM Rule EvAluation using Manufactured Silicon). In addition, we have shown that DREAMS is superior to prior work and applicable to various scenarios characterized by differing degrees of rule violation. Application of DREAMS to an industrial design and diagnosis data has revealed that two of the twenty rules examined have an overall importance level that doubles the values of the remaining rules, a finding that we are further investigating with our industrial collaborators.

A by-product of DREAMS is a significant improvement in diagnostic resolution. Specifically, by understanding which rule is most likely responsive for failure of a given chip allows one to further localize the failure to a more precise location. For the design and test data examined in Section 3, resolution is improved by 69%. Current work is focused on conducting an even larger experiment, further refining the rule-candidate matrix using the defect types targeted by each rule, and developing an algorithm for statistically bounding the yield-loss contributions derived by DREAMS.

5. ACKNOWLEDGMENTS

The authors would like to thank their industrial collaborators for providing the diagnosis and DFM data that enabled the silicon experiments presented in Section 3.2.

6. REFERENCES

[1] V. Pitchumani, “A Hitchhiker's Guide to the DFM Universe,” *IEEE Asia Pacific Conference on Circuits and Systems* pp. 1103-1106, 2006.

[2] C. Guardiani*, et al.*, “An Effective DFM Strategy Requires Accurate Process and IP Pre-Characterization,” *Design Automation Conference*, pp. 760-761, 2005.

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

[4] W. C. Tam and R. D. Blanton “To DFM or Not to DFM,” *Design Automation Conference*, pp. 65-70, June 2011.

[5] C. M. Bishop, Pattern Recognition and Machine Learning, 1 ed.: Springer-Verlag New York, Inc., 2006.

[6] M. Abramovici, M. A. Breuer, and A. D. Friedman*, Digital Systems Testing & Testable Design*. New York, USA: Wiley-IEEE Press, 1994.

[7] “The MATLAB Help Manual,” 2009a ed. Natick, MA.: The MathWorks Inc., 1984-2009.

[8] 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, 2012.

[9] C. Hora, R. Segers, S. Eichenberger, and M. Lousberg, “An Effective Diagnosis Method to Support Yield Improvement,” *IEEE International Test Conference*, pp. 260-269, 2002

[10] M. Keim et al., “A Rapid Yield Learning Flow Based on Production Integrated Layout-Aware Diagnosis,” *IEEE International Test Conference*, 2006.

[11] J. E. Nelson, T. Zanon, J. G. Brown, O. Poku, R. D. Blanton, W. Maly, B. Benware and C. Schuermyer, “Extracting Defect Density and Size Distributions from Product ICs,” *IEEE Design & Test of Computers*, vol. 23, no. 5, pp. 390-400, May 2006.

[12] F. P. Preparata and M. I. Shamos, “Section 7.2.3: Intersection of line segments,” *Computational Geometry: An Introduction, Springer-Verlag*, pp. 278–287, 1985.

[13] M. I. Shamos, D. Hoey, “Geometric Intersection Problems,” *17th IEEE Conf. Foundations of Computer Science*, pp. 208–215, Nov. 1976.

[14] J. L. Bentley and T. A. Ottmann, “Algorithms for Reporting and Counting Geometric Intersections,” *IEEE Transactions on Computers*, pp. 643–647, Oct. 1979.

[15] N. A. Sherwani, “Algorithms for VLSI Design Automation,” *Kluwer Academic Publishers*, 1999.

[16] X. Yu and R. D. Blanton, “Controlling DPPM through Volume Diagnosis,” *IEEE VLSI Test Symposium*, May 2006.

[17] B. Benware, C. Schuermyer, M. Sharma and T. Herrmann, “Determining a Failure Root Cause Distribution from a Population of Layout-Aware Scan Diagnosis Results,” *IEEE Design & Test of Computers*, vol. 29, no. 1, pp. 8-18, 2012.

[18] W. C. Tam and R. D. Blanton, “Physically-Aware Analysis of Systematic Defects in Integrated Circuits,” *IEEE Design & Test of Computers*, Vol. 12, No. 5, pp. 81-93, Sep./Oct. 2012.

[19] W. C. Tam, O. Poku, and R. D. Blanton, “Systematic Defect Identification through Layout Snippet Clustering,” *IEEE International Test Conference*, Oct. 2010.