Using Presilicon Knowledge to Excite Nonlinear Failure Modes in Large Mixed-Signal Circuits

Parijat Mukherjee
Intel Corporation

Peng Li
Texas A&M University

Editor's notes:
With the continuous growing of analog circuit complexity, identifying critical failure modes is of great importance today. This article develops an information-theoretic framework to rank important parameters based on presilicon data. The proposed framework is further used to design the optimal test set that maximizes the probability of observing out-of-specification failures. A phase-locked loop example is adopted to validate the proposed framework.
—Xin Li, Carnegie Mellon University

With the continued scaling of CMOS technology, the complexity of analog/mixed-signal design has grown over time. Along with bigger and more complex circuits, we are forced to deal with increasing design uncertainties such as startup conditions; signal and power noise; and process variation. Slight parametric shifts in analog components can cause the output to change significantly in such a scenario. More importantly, multiple parameters may interact nonlinearly to cause out-of-specification failures. Detecting such out-of-specification failure mechanisms in the presilicon domain [1] helps, but presilicon assumptions about design uncertainty may not be entirely accurate. Moreover, downstream effects of out-of-specification failures on system-level performance are not always known, making postsilicon testing of such failure mechanisms critical.

It is impossible to test for functional failures under every possible input and parametric variation, making design for test (DfT) indispensable. Identifying what parameters to control, and what values to excite, is especially difficult in high-dimensional, highly nonlinear systems since linearity and worst case corner assumptions no longer apply. Traditional sensitivity-based approaches find it difficult to scale as a result. We instead start with prior work on identifying presilicon failure mechanisms [1] and demonstrate a novel way to excite these failure mechanisms. More specifically, the contributions made in this work are twofold.

1) We develop an information-theoretic parameter ranking scheme to assist with design-for-test decisions. This parameter ranking is only meant to serve as a guide, as designer intervention is still required for the actual design of test structures.
2) We then demonstrate how a test set can be selected to maximize the probability of observing out-of-specification failures. This holds true even when dealing with limited design-for-test budgets as demonstrated on a high-dimensional phase-locked loop test case.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/MDAT.2016.2593399
Date of publication: 19 July 2016; date of current version: 19 August 2016.

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

Identifying failure mechanisms

We start by treating the design under test as a statistical model where any design variable that can affect the outcome is treated as an input distribution. Individual distributions may be part of the specification (e.g., input jitter and voltage swing), obtained from measurement data (e.g., process parameters) or may simply reflect a designer's belief about a system (e.g., input signal expectations). Treating circuit parameters as statistical quantities is actually common practice in areas such as yield estimation [2]–[4].

Assuming there are m such parameters, every point in this joint parameter space $x \in X$ is a multidimensional vector of the form $\{x_1, \ldots, x_m\}$. Of these, $k < m$ parameters will be controllable in the post-silicon domain and may be represented as $c = \{x_1, \ldots, x_k\}$. The remaining parameters $\bar{c}|c = \{x_{k+1}, \ldots, x_m\}$ are uncontrollable. The set of all legal values of $c \in C$ is the controllable space.

Being able to predict the probability of functional failure $P(F|x)$ at a given point in the parameter space $x \in X$ is critical to both contributions detailed in this work. The simplest such knowledge representation scheme is a set of hypercubes lying in a high-dimensional parameter space such as the 2-D example in Figure 1. Such "failure regions" of the form $R_i = \cap_j a_{ij} < x_j < b_{ij}$ can be constructed from simulation data using standard machine learning algorithms [1]. While any model that returns $P(F|x)$ may be used for the purpose of failure excitation, an advantage to the "failure-region"-based approach is that it is easy to map chosen tests to preidentified failure modes.

Parameter ranking

In this section, we describe how the failure regions we discuss previously may be used to rank parameters to guide design-for-test decisions. Each failure region R_i in [1] already contains some amount of information regarding a given parameter x_j. If the parameter x_j is critical for region R_i, $[a_{ij}, b_{ij}]$ does not default to $[-\infty, \infty]$, where $a_{ij} \leq x_j \leq b_{ij}$ defines bounds for x_j in R_i. Removing these bounds from region R_i might cause the resulting region (denoted by R_{i-x_j}) to additionally encompass many nonfailure points.

For example, let us assume that a circuit has two parameters x_1 and x_2 and the circuit only fails if one of the parameters (let us say x_1) exceeds a certain bound a_1. This means that failure region $R_1 = \{x_1, x_2 | x_1 > a_1\}$ is made up entirely of failure points. Removing this bound from R_1 would cause the resulting region R_{1-x_1} to encompass many nonfailure points (meaning that its population is not as pure). On the other hand, the presence or absence of x_2 makes no difference as the failure

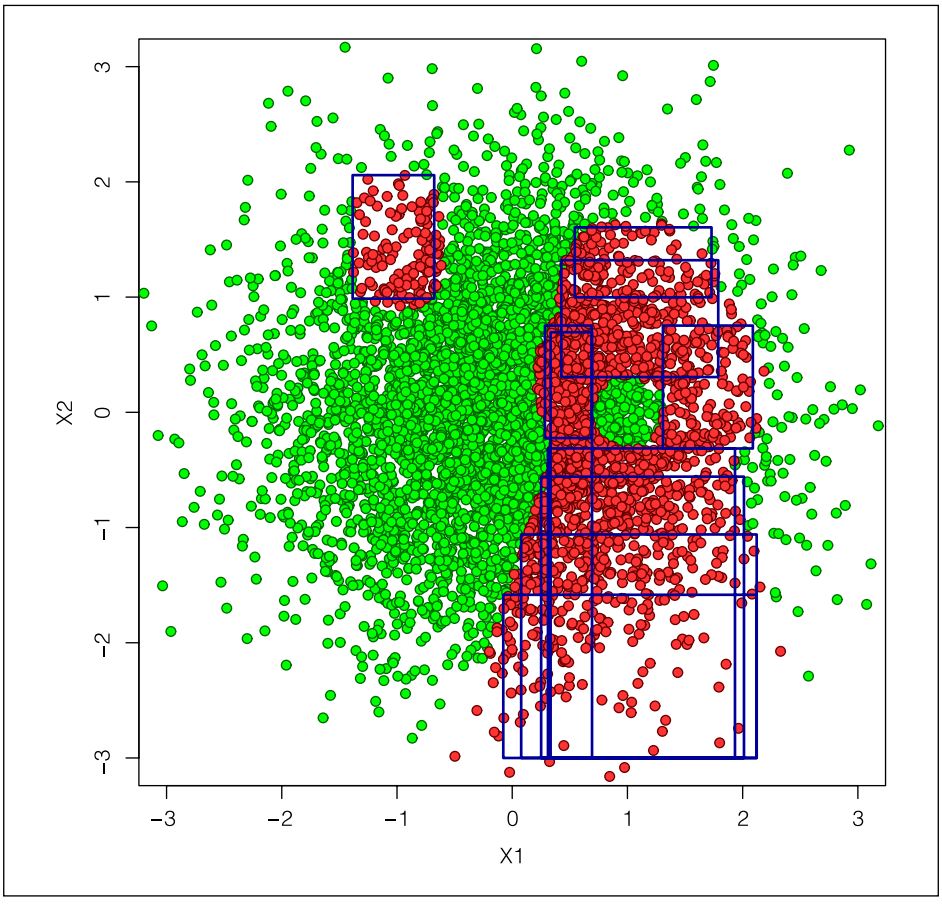

Figure 1. Region primitives: Two-dimensional example. *X1* and *X2* are two independent Gaussian parameters. Regions that do not satisfy specifications are marked in red. Each rectangle corresponds to a failure region primitive we wish to generate.

mechanism is determined by the value of x_1 alone. Clearly, x_1 is a better choice for a design-for-test structure than x_2 in this case.

Since all our circuit parameters have been modeled as random variables with known distributions, this concept can be more formally captured by the expected information gain [5]. Information-gain-based feature selection has been extensively studied in decision tree algorithms [6] and can be easily extended to ranking circuit parameters. To do so, we must first describe entropy which serves as a measure of impurity within a given region. In our case, this measure reduces to

$$H(R_i) = -\sum_{f_k \in (f,\bar{f})} p(f_k|R_i)\log_2 p(f_k|R_i) \quad (1)$$

where f (fail) and $\bar{f}$ (pass) are the two classes of observations in the training data set indicating whether specifications are violated. $p(f_k|R_i)$ is the proportion of observations belonging to class f_k in R_i. The value of H is highest when $p(f_k|R_i) = 0.5$ and lowest when $p(f_k|R_i) = 1$ or 0. For example, $P(f|R_1)$ in the 2-D example described earlier is 1 meaning that $H(R_i) = 0$.

Intuitively, if the addition of the bound $a_{ij} \leq x_j \leq b_{ij}$ helps reduce the entropy of R_{i-x_j}, parameter x_j is critical in defining R_i. The expected information gain from the introduction of parameter x_j to region R_{i-x_j} is given by

$$\mathrm{IG}\left(R_{i-x_j}, x_j\right) = H\left(R_{i-x_j}\right) - H\left(R_{i-x_j}|x_j\right) \quad (2)$$

with the entropy of R_{i-x_j} when split using $a_{ij} \leq x_j \leq b_{ij}$, and $H(R_{i-x_j}|x_j)$ can be evaluated as

$$H\left(R_{i-x_j}|x_j\right) = \sum_{v \in \mathrm{vals}(x_j)} p\left(v|R_{i-x_j}\right) H\left(R_{i-x_j}|v\right) \quad (3)$$

where $\mathrm{vals}(x_j)$ can take one of two values $[a_{ij}, b_{ij}]$ (bounds for x_j in R_i) or $[-\infty, \infty] - [a_{ij}, b_{ij}]$ (entire legal range of values excluding the obtained bounds for x_j).

$\mathrm{IG}(R_{i-x_j}|x_j)$ also refers to the information lost if parameter x_j is removed from R_i which gives us the relative importance of parameters in a given region R_i. If we were to obtain $\mathrm{IG}(R_{i-x_j}|x_j)\ \forall i$, we can then evaluate $\mathrm{IG}(R|x_j)$ over the entire set of regions as

$$\mathrm{IG}(R|x_j) = \sum_{R_i \in \{R_1,\ldots,R_m\}} p\left(R_{i-x_j}\right)\mathrm{IG}\left(R_{i-x_j}|x_j\right). \quad (4)$$

Ranking parameters in decreasing order of $\mathrm{IG}(R|x_j)$ thus allows us to identify which parameters were most important in describing the set of failure regions.

It is to be noted that this manner of ranking is an artifact of the failure rules and not the circuit itself and is therefore not perfect. Notably, the information gain may be

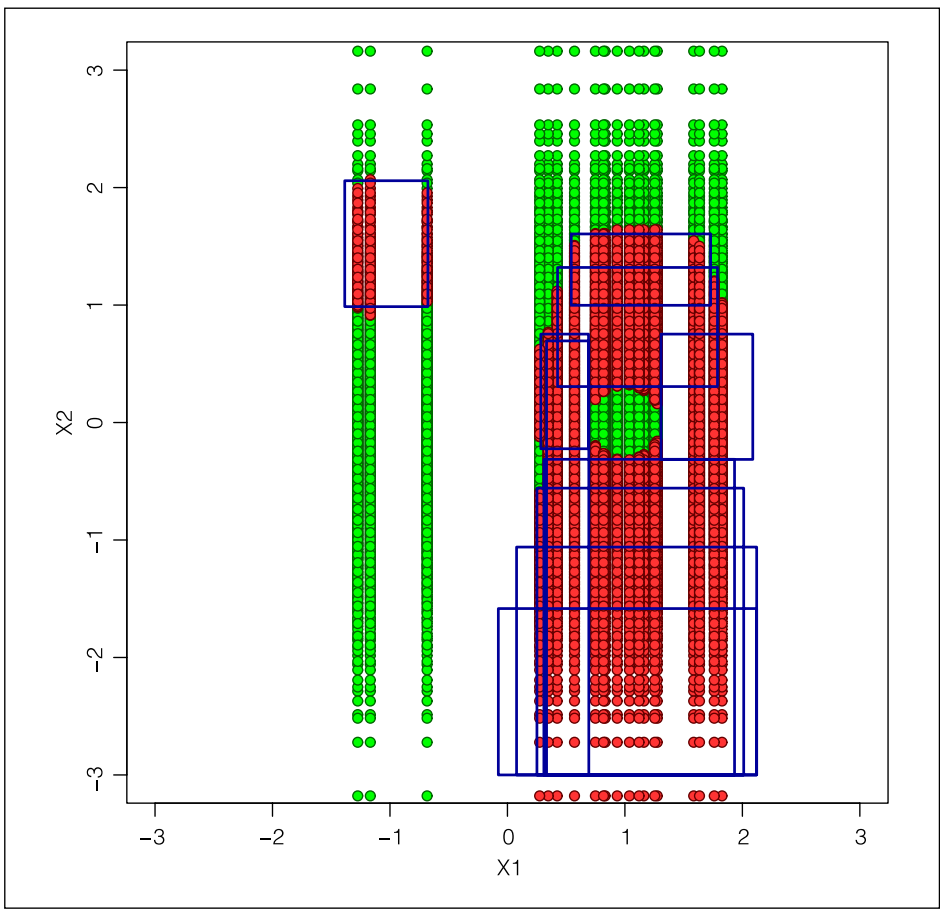

Figure 2. Selecting values for controllable test parameters. *X*1 is the controllable parameter. For every sampled point on *X*1, the expected uncertainty along *X*2 is demonstrated. The expected uncertainty is based on presilicon assumptions.

biased when applied to parameters that can take on a large number of distinct values. This occurs when the piecewise constant bounds $R_i = \cap_j a_{ij} < x_j < b_{ij}$ need too many regions to capture complex interactions. Since we are dealing with low failure probabilities, this does not prove to be a problem in our case.

Test set selection to maximize observed failures

Implementing DfT structures where we can vary the value of a parameter over its entire legal range is usually very expensive. Practically, DfT structures are designed to operate in one of a few different modes which can be controlled with a digital knob. Effectively, the continuous space C gets discretized into a finite set of controllable points $S \subset C$. The goal of test-set selection is to pick a set of t points $\hat{C} \subset S$ such that the probability of exciting a failure $P(F|\hat{C})$ is maximized.

Intuitively, we simply wish to boost the number of points falling within the set of failure regions R_i instead of dealing with $P(F|c,\bar{c})$. This is empirically demonstrated in Figure 2. An added benefit of a failure-region-based approach is that it now becomes possible to ensure that both R_i and R_j (where $i \neq j$) have a nonzero probability of being covered by $\hat{C}$ [as compared to just maximizing $P(F|\hat{C})$]. This is possible even if the effect of uncontrollable dimensions is higher for R_i.

More formally, we wish to sample a set of points $\hat{C} \subseteq S$ such that the probability of exciting regions $R = \{R_1, \ldots, R_m\}$, $P(R|\hat{C})$ is maximized. To do so, we observe that

$$P(R|c) = \sum_{\bar{c} \in \bar{C}} P(R|c,\bar{c}) p(\bar{c}|c) \quad (5)$$

$$P(R|\hat{C}) = \sum_{c \in \hat{C}} P(R|c) \quad (6)$$

where $P(R|c,\bar{c})$ is either 0 or 1 depending on whether $x = \{c,\bar{c}\} \in R$, and $p(\bar{c}|c)$ represents the expected distribution along the uncontrollable dimensions for a given controllable point c.

In the presilicon domain, both $p(c)$ and $p(\bar{c}|c)$ are approximately known (based on certain assumptions). This leaves $P(F|c,\bar{c}) = P(F|x)$ that can be directly ascertained from any failure model as described previously. Thus, test-set selection using failure regions can be performed in two steps: 1) Monte Carlo sampling over $\bar{c} \in \bar{C}$ to estimate $P(R|c)$ over all candidate test points $c \in S$; and 2) maximize $P(R|\hat{C})$ by sampling $c \in C$ with probability $P(R|c)$. The pseudocode in Algorithm 1 performs both these functions.

Algorithm 1: GenTestParams $(p(C), p(\bar{C}|C), R = \{R_1, \ldots, R_m\})$

```
1:  S ← sample c ∈ C with probability p(c)
2:  for c ∈ S do
3:    S̄ ← sample c̄ ∈ C̄ using p(c̄|c)
4:    for c̄ ∈ S̄ do
5:      if {c, c̄} ∈ R then
6:        P(R|c, c̄) ← 1
7:      Else
8:        P(R|c, c̄) ← 0
9:      end if
10:   end for
11: P(R|c) ← Σ_{c̄∈S̄} P(R|c,c̄)p(c̄|c)
12:  end for
13:  Ĉ ← sample c ∈ S with probability P(R|c)/Σ_{c∈S} P(R|c)
```

Computation of $P(R|c)$ for every $c \in S$ forms the majority of pseudocode given in Algorithm 1 (lines 2–12). The inner loop (lines 4–10) computes $P(R|c,\bar{c})\ \forall \bar{c}|c$. The outer loop computes $P(R|c)\ \forall c \in S$. Line 1 simply generates the candidate set S. Line 13 picks $\hat{C}$ from S using $P(R|c)\ \forall c \in S$.

It is to be noted that the simplest way to maximize (6) is by simply ranking all $c \in S$ by $P(R|c)$ and picking the first t controllable points. However, this may cause $\hat{C}$ to be concentrated in debugging a single failure mechanism. Instead, we use a weighted sampling scheme [7] where the probability of selecting $c \in S$ is proportional to $P(R|c)$. Points with the highest $P(R|c)$, of course, have the highest probability of being selected where $P(R|c) = 0$ will ensure that a point is never selected. The goal here is to ensure that $\hat{C}$ is not completely clustered around the same failure mechanism as that is not very interesting from a debug perspective.

It is to be noted that the "failure regions" (or any other alternate knowledge representation) need not be perfect. This is because the observed $P(F|c)$ during actual measurements will show a high degree of variance due to small sample size, given the effect of the uncontrollable parameters. This variance is far larger than the error arising

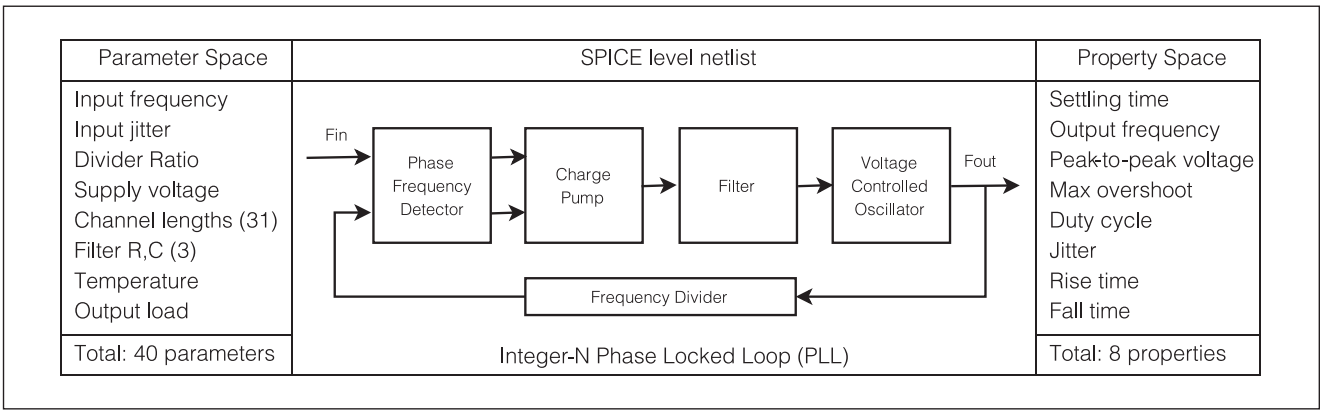

Figure 3. Test case setup.

from an imperfect presilicon model. As long as the most important trends are captured, the computational cost of building an exact presilicon model far outweighs the benefits.

Results

The basic structure of our phase-locked loop test case is described in Figure 3. The parameters are chosen so as to demonstrate the versatility of the proposed methodology. Specifically, the ability to deal with multiple parameter distributions (circuit parameters such as channel lengths and filter Rs and Cs are Gaussian; everything else is uniform) and a mix of discrete and uniform data types (divider ratio is discrete; everything else is continuous) is demonstrated. Last, correlations between parameters have been modeled either explicitly (by defining the joint distribution in such a manner) or implicitly (as a part of the SPICE models used in simulation). For example, transistor channel lengths display both systematic and random variations.

An out-of-specification response along any of our output properties is considered to be a parametric fault. Some of these properties depend on the input parameters as well. For example, the output frequency specification depends on both the input frequency and the divider ratio. The objective of this work is simply to maximize the probability of observing such failures, especially in the presence of nonlinear interactions. Since the probability is highly dependent upon parameter distributions, we assume that chosen controllable parameters are only varied in their legal range of operation. We do not excite parameters outside their input specifications, since that means exciting failures we are not really interested in.

Parameter ranking

The parameter rankings for PLL1 and PLL2 are listed in Tables 1 and 2, respectively. Optimally, our design-for-test structures would allow for controllability of all these parameters. However, process related parameters such as R, C values and channel

Table 1 Top five parameters in PLL1 ranked based on the importance metric described in text.

Parameter	$IG(R\|parameter)$
C1	0.0307
DividerRatio	0.0157
R1	0.0129
InputJitter	0.0102
InputFrequency	0.0046

Table 2 Top five parameters in PLL2 ranked based on the importance metric described in text.

Parameter	$IG(R\|parameter)$
DividerRatio	0.0765
InputFrequency	0.0359
VDD	0.0098
LCP5	0.0069
C2	0.0065

lengths are hard to control in practice though experiments show up to a 5× increase in $P(R|\hat{C})$ should we be able to control these parameters. For the purpose of this article, we assume that the same four parameters (input frequency, divider ratio, supply voltage, and temperature) are controllable for both variants of the test case PLL1 and PLL2.

As can be seen, the choice of DividerRatio and (to a lesser extent) InputFrequency as controllable parameters was justified for PLL1. The remaining three parameters identified as important are uncontrollable by their very nature. Once again, this analysis is based upon the legal range of operation for both DividerRatio and InputFrequency. If the input specifications change, this ranking would change as well.

PLL2 on the other hand has DividerRatio, InputFrequency, and the Supply Voltage VDD as the top three parameters, i.e., the test-set selection can be expected to perform a lot better for PLL2. Our results justify this expectation. The rankings for PLL2 are particularly significant since the generated failure regions are numerous and hard to analyze manually. The parameter ranking is not just desirable but indispensable in this case.

Test set selection

Assuming that the failure regions [1] form a fault dictionary we wish to exercise, $P(R|\hat{C}) = \sum_{c\in\hat{C}} P(R|c)p(c|\hat{C})$ gives us a good indicator of the number of failures we should expect to see when exercising $\hat{C}$. Comparing $P(R|\hat{C})$ with $P(R|S)$ also gives us an indicator of how good we expect the test set to be, compared to randomly sampling the test plan. Since we are dealing with a very limited test budget (20 points), there may be a high degree of variability for a single experiment. As a result, the comparisons in Figures 4 and 5 are based on the failure rates averaged over 100 independent experiments.

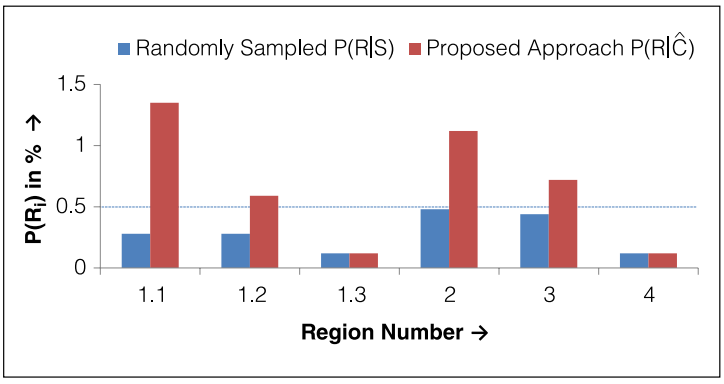

Figure 4. PLL1 test-set selection. Probabilities of regions demarcated by one or more controllable parameters are effectively boosted. If the region is bounded by uncontrollable elements alone, no test set can perform better than randomly sampling the test plan.

The effect of applying our test-set selection methodology on PLL1 is shown in Figure 4. The reason why some regions are boosted more effectively than others becomes immediately obvious if we refer to the parameter ranking in Table 1. The most important controllable parameter happens to be the DividerRatio. As a result, rules demarcated by *DividerRatio* (R_2 and R_3) are boosted more effectively than the others. On the other hand, some rules ($R_{1.3}$ and R_4) are not affected at all since they are not demarcated by any controllable parameter. Their probability remains unaffected for all test sets.

A similar trend is seen if we run Algorithm 1 on the failure regions generated for the alternate

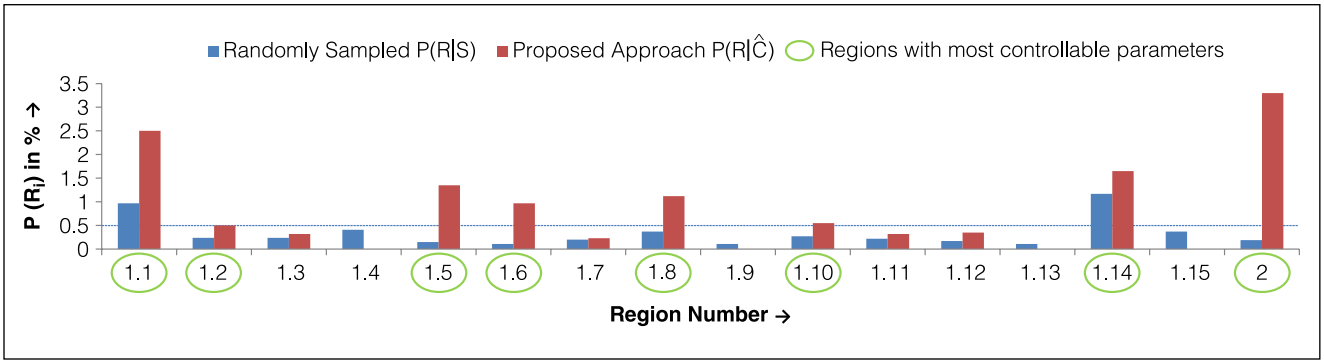

Figure 5. PLL2 test-set selection. Regions with failure probability above 0.5% when excited with chosen test parameters will be considered statistically significant and used for postsilicon debug.

design PLL2 as seen in Figure 5. As expected, rules which feature the most important parameters from our parameter ranking in Table 2 are boosted most effectively as well. Also, given that the DfT for PLL2 is better, we get a much higher chance of observing failure than with PLL1.

It is to be noted that the proposed methodology works very well even if the postsilicon reality differs significantly from presilicon expectations. While we do not show the detailed analysis here, the difference between $P(R|\hat{C})$ and the observed probability of failure $P(F|\hat{C})$ is negligible unless the shift in the expected value of an uncontrollable parameter x_j is of the same order of magnitude as $|b_{ij} - a_{ij}|\ \forall R_i$.

In this work, we demonstrate a test-set selection methodology that best uses (possibly approximate) presilicon knowledge to choose a test set to excite real silicon with. We also demonstrate an information-theory-based parameter ranking scheme that helps in determining which parameters to control and observe in the first place. These contributions together allow us to maximize the probability of observing postsilicon failures even when dealing with limited measurement budgets and large circuits with highly nonlinear dynamics. More importantly, it does so in a manner that ensures we are not testing around a single failure mechanism. While the contributions detailed in this article are built upon a simple "failure-region"-based knowledge representation scheme, the algorithms and methodologies described herein can be easily extended to use more complex presilicon models. If interpretability of failure modes is not a concern, the only bottleneck in doing so may be the computational cost incurred in querying the (possibly more expensive) model. ■

Acknowledgment

This work was supported by Semiconductor Research Corporation (SRC) through the Texas Analog Center of Excellence at the University of Texas at Dallas (Task ID:1836.128).

■ References

[1] P. Mukherjee and P. Li, "Leveraging pre-silicon data to diagnose out-of-specification failures in mixed-signal circuits," in *Proc. 51st Annu. Design Autom. Conf.*, 2014, pp. 9:1–9:6.

[2] C. Gu and J. Roychowdhury, "An efficient, fully nonlinear, variability-aware non-Monte-Carlo yield estimation procedure with applications to SRAM cells and ring oscillators," in *Proc. Asia South Pacific Design Autom. Conf.*, 2008, pp. 754–761.

[3] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *Proc. 43rd ACM/IEEE Design Autom. Conf.*, 2006, pp. 69–72.

[4] P. Mukherjee, C. S. Amin, and P. Li, "Approximate property checking of mixed-signal circuits," in *Proc. 51st Annu. Design Autom. Conf.*, 2014, pp. 115:1–115:6.

[5] J. T. Kent, "Information gain and a general measure of correlation," *Biometrika*, vol. 70, no. 1, pp. 163–173, 1983.

[6] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann, 1993.

[7] C.-K. Wong and M. C. Easton, "An efficient method for weighted sampling without replacement," *SIAM J. Comput.*, vol. 9, no. 1, pp. 111–113, 1980.

Parijat Mukherjee is currently a Research Scientist at Strategic CAD Labs, Intel Corporation, Hillsboro, OR, USA, working on high-dimensional verification and validation problems. He has multiple publications in the area including a DAC'11 best paper award winning work on stability analysis and more recent work on high-dimensional property checking and failure diagnosis. Mukherjee has a PhD in computer engineering from Texas A&M University, College Station, TX, USA.

Peng Li has been on the faculty of the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA, since August 2004, where he is currently a Professor. He is also a member of the Faculty of Neuroscience and the Graduate Faculty of School of Graduate Studies, Health Science Center at Texas A&M. He has authored and coauthored two books, six book chapters, and over 150 journal and conference publications. Li has a PhD in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA.

■ Direct questions and comments about this article to Parijat Mukherjee, Intel Corporation, Hillsboro, OR 97124 USA; parijat.mukherjee@intel.com.