Fast Statistical Analysis of Rare Circuit Failure Events via Scaled-Sigma Sampling for High-Dimensional Variation Space

Shupeng Sun[1], Xin Li[1], Hongzhou Liu[2], Kangsheng Luo[2], and Ben Gu[3]
[1]Carnegie Mellon University, Pittsburgh, PA 15213 USA, {shupengs, xinli}@ece.cmu.edu
[2]Cadence Design Systems, Pittsburgh, PA 15238 USA, {hliu, ksluo}@cadence.com
[3]Cadence Design Systems, Austin, TX 78759 USA, gxin@cadence.com

ABSTRACT
Accurately estimating the rare failure rates for nanoscale circuit blocks (e.g., SRAM, DFF, etc.) is a challenging task, especially when the variation space is high-dimensional. In this paper, we propose a novel scaled-sigma sampling (SSS) method to address this technical challenge. The key idea of SSS is to generate random samples from a distorted distribution for which the standard deviation (i.e., sigma) is scaled up. Next, the failure rate is accurately estimated from these scaled random samples by using an analytical model derived from the theorem of "soft maximum". Several circuit examples designed in nanoscale technologies demonstrate that the proposed SSS method achieves superior accuracy over the traditional importance sampling technique when the dimensionality of the variation space is more than a few hundred.

1. INTRODUCTION
With aggressive technology scaling, process variation has become a growing concern for today's integrated circuits (ICs) [1]. As a complex IC may integrate numerous circuit components (e.g., millions of SRAM bit-cells integrated in a high-performance microprocessor), each component must be designed to be extremely robust to large-scale process variations. For instance, the failure rate of an SRAM bit-cell must be less than 10^{-8}~10^{-6} so that the full microprocessor system, containing millions of SRAM bit-cells, can achieve sufficiently high yield [2]-[3]. For this reason, efficiently simulating the rare failure events for circuit components and accurately estimating their failure rates is an important task for the IC design community.

To address this issue, a large number of statistical algorithms and methodologies have been developed in the literature [4]-[14]. Most of these traditional methods focus on failure rate estimation for SRAM bit-cells that consist of few (e.g., 6~10) transistors. In these cases, only a small number of (e.g., 10~50) independent random variables are used to model process variations and, hence, the corresponding variation space is low-dimensional. However, several recent trends suggest us to re-visit the aforementioned assumption of low-dimensional variation space:

- **Dynamic SRAM bit-cell stability related to peripherals**: It has been demonstrated that dynamic SRAM bit-cell stability depends not only on the bit-cell itself but also on its peripherals (e.g., other bit-cells connected to the same bit line) [15]. Hence, a large number of transistors from multiple SRAM bit-cells and their peripherals must be considered to simulate the dynamic stability. As a result, many independent random variables must be used to model the process variations, including device mismatches, for these transistors.
- **Rare failure events for non-SRAM circuits**: In addition to SRAM bit-cells, a complex IC system may contain a large number of other circuit components (e.g., DFFs) that must be designed with extremely low failure rates. Taking DFF as an example, it typically contains about 20 transistors [16] and the random mismatch of a single transistor is often modeled by 10~40 independent random variables at an advanced technology node. Hence, the total number of independent random variables can easily reach a few hundred for DFF analysis.

The combination of these recent trends renders a high-dimensional variation space that cannot be efficiently handled by most traditional techniques. It, in turn, poses an immediate need of developing a new CAD tool to accurately capture rare failure events in *high-dimensional* variation space with low computational cost.

In this paper, we propose a novel *scaled-sigma sampling* (SSS) method to estimate the rare failure rate in a high-dimensional variation space. SSS is particularly developed to address the following two fundamental questions: (i) how to efficiently draw random samples from the rare failure region, and (ii) how to estimate the failure rate based on these random samples. Unlike the brute-force Monte Carlo analysis that directly samples the variation space and, hence, only few samples fall into the failure region, SSS draws random samples from a distorted probability density function (PDF) for which the standard deviation (i.e., sigma) is scaled up. Conceptually, it is equivalent to increasing the magnitude of process variations. As a result, a large number of samples can now fall into the failure region.

Once the distorted random samples are generated, an analytical model is further derived to estimate the failure rate based upon the theorem of "soft maximum" [17]. While most traditional techniques (e.g., importance sampling) become inefficient as the dimensionality increases, our proposed SSS approach does not suffer from such a dimensionality problem, as will be explained in the technical sections of this paper. In addition, a new statistical algorithm is developed to accurately estimate the confidence interval of SSS based on re-sampling. Our numerical experiments in Section 5 demonstrate that SSS achieves superior accuracy over the traditional importance sampling method for several circuit examples where hundreds of independent random variables are used to model process variations.

The remainder of this paper is organized as follows. In Section 2, we briefly review the background on Monte Carlo analysis and importance sampling, and then propose the SSS method in Section 3. In Section 4, a few implementation issues are discussed in detail, and several circuit examples are presented to demonstrate the efficacy of SSS in Section 5. Finally, we conclude in Section 6.

2. BACKGROUND
2.1 Monte Carlo Analysis
Suppose that the vector

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_M]^T \tag{1}$$

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

is an M-dimensional random variable modeling process variations and its joint PDF is $f(\mathbf{x})$. Typically, $f(\mathbf{x})$ is modeled as a multivariate Normal distribution. Without loss of generality, we further assume that the random variables $\{x_m;\ m = 1, 2, \ldots, M\}$ in the vector $\mathbf{x}$ are mutually independent and standard Normal (i.e., with zero mean and unit variance)

$$f(\mathbf{x}) = \prod_{m=1}^{M}\left[\frac{1}{\sqrt{2\pi}}\cdot\exp\left(-\frac{x_m^2}{2}\right)\right] = \frac{\exp\left(-\|\mathbf{x}\|_2^2/2\right)}{\left(\sqrt{2\pi}\right)^M} \tag{2}$$

where $\|\bullet\|_2$ denotes the L_2-norm of a vector. Any correlated random variables that are jointly Normal can be transformed to the independent random variables $\{x_m;\ m = 1, 2, \ldots, M\}$ by principal component analysis [18].

The failure rate of a circuit can be mathematically represented as

$$P_f = \int_{\Omega} f(\mathbf{x})\cdot d\mathbf{x} \tag{3}$$

where Ω denotes the failure region, i.e., the subset of the variation space where the performance of interest does not meet the specification. Alternatively, the failure rate in (3) can be defined as

$$P_f = \int_{-\infty}^{+\infty} I(\mathbf{x})\cdot f(\mathbf{x})\cdot d\mathbf{x} \tag{4}$$

where $I(\mathbf{x})$ represents the indicator function

$$I(\mathbf{x}) = \begin{cases} 1 & \mathbf{x}\in\Omega \\ 0 & \mathbf{x}\notin\Omega \end{cases}. \tag{5}$$

The failure rate P_f can be estimated by a brute-force Monte Carlo analysis. The key idea is to draw N random samples from $f(\mathbf{x})$, and then compute the mean of the indicator function $I(\mathbf{x})$ based on these samples

$$P_f^{MC} = \frac{1}{N}\cdot\sum_{n=1}^{N} I\left[\mathbf{x}^{(n)}\right] \tag{6}$$

where $\mathbf{x}^{(n)}$ is the n-th random sample.

When a brute-force Monte Carlo analysis is applied to estimate the failure rate P_f that is extremely small (e.g., 10^{-8}~10^{-6}), most random samples drawn from the PDF $f(\mathbf{x})$ do not fall into the failure region Ω. Hence, a large number of (e.g., 10^7~10^9) samples are needed to accurately estimate P_f. Note that each Monte Carlo sample is created by running an expensive transistor-level simulation. In other words, 10^7~10^9 simulations must be performed in order to collect 10^7~10^9 samples. It, in turn, implies that a brute-force Monte Carlo analysis can be extremely expensive for our application of rare failure rate estimation.

2.2 Importance Sampling

To reduce the computational cost, several statistical algorithms based on importance sampling have been proposed [4], [7], [10], [12], [14]. The key idea is to sample a distorted PDF $g(\mathbf{x})$, instead of the original PDF $f(\mathbf{x})$, so that most random samples fall into the failure region Ω. In this case, the failure probability can be expressed as

$$P_f = \int_{-\infty}^{+\infty} \frac{I(\mathbf{x})\cdot f(\mathbf{x})}{g(\mathbf{x})}\cdot g(\mathbf{x})\cdot d\mathbf{x}\,. \tag{7}$$

If N random samples $\{\mathbf{x}^{(n)};\ n = 1, 2, \ldots, N\}$ are drawn from $g(\mathbf{x})$, the failure rate in (7) can be approximated by

$$P_f^{IS} = \frac{1}{N}\cdot\sum_{n=1}^{N}\frac{f\left[\mathbf{x}^{(n)}\right]}{g\left[\mathbf{x}^{(n)}\right]}\cdot I\left[\mathbf{x}^{(n)}\right]. \tag{8}$$

The estimated failure rates in (6) and (8) are identical, if and only if the number of samples (i.e., N) is infinite. In practice, when a finite number of samples are available, the results from (6) and (8) can be substantially different. If the distorted PDF $g(\mathbf{x})$ is properly chosen for importance sampling, P_f^{IS} in (8) can be much more accurate than P_f^{MC} in (6). Theoretically, the optimal $g(\mathbf{x})$ leading to maximum estimation accuracy is defined as

$$g^{OPT}(\mathbf{x}) = \frac{f(\mathbf{x})}{P_f}\cdot I(\mathbf{x})\,. \tag{9}$$

Intuitively, if $g^{OPT}(\mathbf{x})$ is applied, P_f^{IS} in (8) becomes a constant with zero variance. Therefore, the failure rate can be accurately estimated by P_f^{IS} with very few samples.

Eq. (9) implies that the optimal PDF $g^{OPT}(\mathbf{x})$ is non-zero if and only if the variable $\mathbf{x}$ sits in the failure region. Namely, we should directly sample the failure region to achieve maximum accuracy. Furthermore, $g^{OPT}(\mathbf{x})$ is proportional to the original PDF $f(\mathbf{x})$ of process variations. In other words, the entire failure region should not be sampled uniformly. Instead, we should sample the high-probability failure region that is most likely to occur.

Applying importance sampling, however, is not trivial in practice. The optimal PDF $g^{OPT}(\mathbf{x})$ in (9) cannot be easily found, since the indicator function $I(\mathbf{x})$ is unknown. Most existing importance sampling methods attempt to approximate $g^{OPT}(\mathbf{x})$ by applying various heuristics. The key idea is to first search the high-probability failure region and then a distorted PDF $g(\mathbf{x})$ is constructed to directly draw random samples from such a high-probability failure region.

While the traditional importance sampling methods have been successfully applied to low-dimensional problems (e.g., 10~50 random variables), they remain ill-equipped to efficiently explore the high-dimensional variation space (e.g., 10^2~10^3 random variables) that is of great importance today. One major bottleneck lies in the high computational cost of the search algorithm, as it cannot easily find the high-probability failure region in a high-dimensional variation space. Such a computational cost issue is most pronounced, when the failure region of interest has a complicated (e.g., non-convex or even discontinuous) shape. To the best of our knowledge, there is no existing algorithm that can efficiently search a high-dimensional variation space. It, in turn, motivates us to develop a new approach in this paper to solve such high-dimensional problems.

3. SCALED-SIGMA SAMPLING

Unlike the traditional importance sampling methods that must explicitly identify the high-probability failure region, our proposed SSS approach takes a completely different strategy to address the following two fundamental questions: (i) how to efficiently draw random samples from the high-probability failure region, and (ii) how to estimate the failure rate based on these random samples. In what follows, we will derive the mathematical formulation of SSS and highlight its novelties.

3.1 Statistical Sampling

For the application of rare failure rate estimation, a failure event often occurs at the tail of the PDF $f(\mathbf{x})$. Given the jointly Normal distribution $f(\mathbf{x})$ in (2), it implies that the failure region is far away from the origin $\mathbf{x} = \mathbf{0}$, as shown in Figure 1(a). Since the failure rate is extremely small, a brute-force Monte Carlo analysis

cannot efficiently draw random samples from the failure region. Namely, many samples cannot reach the tail of the probability distribution.

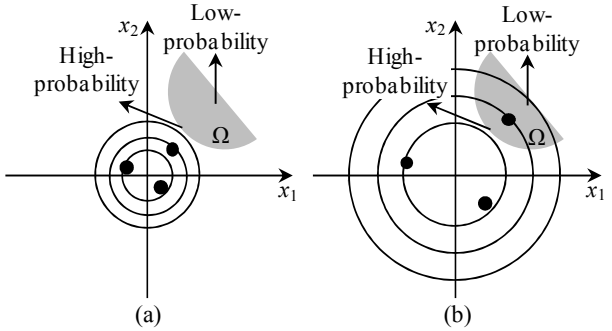


Figure 1. The proposed SSS is illustrated by a 2-D example where the grey area Ω denotes the failure region and the circles represent the contour lines of the PDF. (a) Rare failure events occur at the tail of the original PDF $f(\mathbf{x})$ and the failure region is far away from the origin $\mathbf{x} = \mathbf{0}$. (b) The scaled PDF $g(\mathbf{x})$ widely spreads over a large region and the scaled samples are likely to reach the far-away failure region.

In this paper, we propose a novel-yet-simple idea to address the aforementioned sampling issue. Given the PDF $f(\mathbf{x})$ in (2) for the M-dimensional random variable $\mathbf{x}$, we scale up the standard deviation of $\mathbf{x}$ by a *scaling factor* s ($s > 1$), yielding the following distribution

$$g(\mathbf{x}) = \prod_{m=1}^{M} \left[\frac{1}{\sqrt{2\pi}s} \cdot \exp\left(-\frac{x_m^2}{2s^2} \right) \right] = \frac{\exp\left(-\|\mathbf{x}\|_2^2 / 2s^2 \right)}{\left(\sqrt{2\pi} \right)^M s^M}. \quad (10)$$

Once the standard deviation of $\mathbf{x}$ is increased by a factor of s, we conceptually increase the magnitude of process variations. Hence, the PDF $g(\mathbf{x})$ widely spreads over a large region and the probability for a random sample to reach the far-away failure region increases, as shown in Figure 1(b).

From an alternative viewpoint, the original random variables $\{x_m;\ m = 1, 2, \ldots, M\}$ follow the independent standard Normal distributions defined in (2). If we scale each of them (say, x_m) by a factor of s, the scaled random variables $\{s \cdot x_m;\ m = 1, 2, \ldots, M\}$ follow the PDF $g(\mathbf{x})$ in (10). Hence, when sampling the scaled PDF $g(\mathbf{x})$, we can first draw random samples from the original PDF $f(\mathbf{x})$ and then scale each random sample by a factor of s. As a result, the scaled samples will move far away from the origin $\mathbf{x} = \mathbf{0}$ and are likely to reach the failure region, as shown in Figure 1(b).

On the other hand, it is important to note that the mean of the scaled PDF $g(\mathbf{x})$ remains $\mathbf{0}$, which is identical to the mean of the original PDF $f(\mathbf{x})$. Hence, for a given sampling location $\mathbf{x}$, the likelihood defined by the scaled PDF $g(\mathbf{x})$ remains inversely proportional to the length of the vector $\mathbf{x}$ (i.e., $\|\mathbf{x}\|_2$). Namely, it is more (or less) likely to reach the sampling location $\mathbf{x}$, if the distance between the location $\mathbf{x}$ and the origin $\mathbf{0}$ is smaller (or larger). It, in turn, implies that the high-probability failure region associated with the original PDF $f(\mathbf{x})$ remains the high-probability failure region after the PDF is scaled to $g(\mathbf{x})$, as shown in Figure 1(a) and (b). Scaling the PDF from $f(\mathbf{x})$ to $g(\mathbf{x})$ does not change the location of the high-probability failure region; instead, it only makes the failure region easy to sample.

Once the scaled random samples are drawn from $g(\mathbf{x})$ in (10), we need to further estimate the failure rate P_f defined in (4). Since the scaled PDF $g(\mathbf{x})$ and the original PDF $f(\mathbf{x})$ are different, we cannot simply average the random samples generated by $g(\mathbf{x})$ to calculate the failure rate P_f defined by $f(\mathbf{x})$. A major contribution of this paper is to derive an analytical model to accurately estimate the failure rate P_f from the scaled random samples, as will be discussed in detail in the next sub-section.

3.2 Failure Rate Estimation

We assume that N random samples $\{\mathbf{x}^{(n)};\ n = 1, 2, \ldots, N\}$ are drawn from the scaled PDF $g(\mathbf{x})$ in (10). One straightforward way to estimate the failure rate P_f is based upon the theory of importance sampling. Namely, since the random samples are generated by the scaled PDF $g(\mathbf{x})$ that is different from the original PDF $f(\mathbf{x})$, we can estimate the failure rate P_f by calculating the average of $f(\mathbf{x}) \cdot I(\mathbf{x}) / g(\mathbf{x})$, as shown by the estimator P_f^{IS} in (8).

Such a simple approach, however, does not result in an accurate failure rate, if the dimensionality of the variation space (i.e., M) is large. To understand the reason, we note that each random sample $\mathbf{x}^{(n)}$ contributes to the estimator P_f^{IS} in (8) by the following amount

$$\frac{f\left[\mathbf{x}^{(n)}\right]}{g\left[\mathbf{x}^{(n)}\right]} \cdot I\left[\mathbf{x}^{(n)}\right]. \quad (11)$$

Substitute (2) and (10) into (11), yielding

$$\frac{f\left[\mathbf{x}^{(n)}\right]}{g\left[\mathbf{x}^{(n)}\right]} \cdot I\left[\mathbf{x}^{(n)}\right] = s^M \cdot \exp\left[\frac{1}{2} \cdot \left(\frac{1}{s^2} - 1 \right) \cdot \left\| \mathbf{x}^{(n)} \right\|_2^2 \right] \cdot I\left[\mathbf{x}^{(n)}\right]. \quad (12)$$

Given a fixed scaling factor s ($s > 1$), the random variables $\{x_m;\ m = 1, 2, \ldots, M\}$ in the vector $\mathbf{x}$ follow the independent Normal distributions defined by (10). The summation of the squares of these random variables, i.e., $\|\mathbf{x}\|_2^2$, is a random variable that follows a scaled chi-square distribution [19]. It can be shown that the variance of $\|\mathbf{x}\|_2^2$ is equal to

$$\mathrm{VAR}\left(\|\mathbf{x}\|_2^2 \right) = 2 \cdot s^4 \cdot M \quad (13)$$

where $\mathrm{VAR}(\bullet)$ denotes the variance of a random variable. Eq. (13) implies that the variance of $\|\mathbf{x}\|_2^2$ increases with the dimensionality M. After taking the exponential function in (12), the variance of $f(\mathbf{x}) \cdot I(\mathbf{x}) / g(\mathbf{x})$ can be prohibitively large in a high-dimensional variation space. It, in turn, implies that the estimator P_f^{IS} based on importance sampling has a large variance and is not sufficiently accurate. It does not fit our need of high-dimensional failure rate estimation in this paper.

Instead of relying on the theory of importance sampling, our proposed SSS method attempts to estimate the failure rate P_f from a completely different avenue. We first take a look at the "scaled" failure rate corresponding to the scaled PDF $g(\mathbf{x})$

$$P_g = \int_{-\infty}^{+\infty} I(\mathbf{x}) \cdot g(\mathbf{x}) \cdot d\mathbf{x}. \quad (14)$$

Our objective is to study the relation between the scaled failure rate P_g in (14) and the original failure rate P_f in (4). Towards this goal, we partition the M-dimensional variation space into a large number of identical hyper-rectangles with the same volume and the scaled failure rate P_g in (14) can be approximated as

$$P_g \approx \sum_k I\left[\mathbf{x}^{(k)}\right] \cdot g\left[\mathbf{x}^{(k)}\right] \cdot \Delta\mathbf{x} \quad (15)$$

where $\Delta\mathbf{x}$ denotes the volume of a hyper-rectangle. The approximation in (15) is accurate, if each hyper-rectangle is sufficiently small. Given the definition of the indicator function $I(\mathbf{x})$ in (5), Eq. (15) can be re-written as

$$P_g \approx \sum_{k\in\Omega} g\left[\mathbf{x}^{(k)}\right]\cdot \Delta\mathbf{x} \tag{16}$$

where $\{k;\ k \in \Omega\}$ represents the set of all hyper-rectangles that fall into the failure region.

Substituting (10) into (16), we have

$$P_g \approx \frac{\Delta\mathbf{x}}{\left(\sqrt{2\pi}\right)^M s^M}\cdot \sum_{k\in\Omega} \exp\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right]. \tag{17}$$

Taking the logarithm on both sides of (17) yields

$$\log P_g \approx \log\frac{\Delta\mathbf{x}}{\left(\sqrt{2\pi}\right)^M} - M\cdot\log s + \underset{k\in\Omega}{\text{lse}}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right] \tag{18}$$

where

$$\underset{k\in\Omega}{\text{lse}}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right] = \log\left\{\sum_{k\in\Omega}\exp\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right]\right\} \tag{19}$$

stands for the log-sum-exp function. The function lse(•) in (19) is also known as the "soft maximum" from the mathematics [17]. Namely, it is considered as a good approximation of the maximum operator

$$\underset{k\in\Omega}{\text{lse}}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right] \approx \max_{k\in\Omega}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right]. \tag{20}$$

Intuitively, since the maximum of $\{-\|\mathbf{x}^{(k)}\|_2^2/2s^2;\ k \in \Omega\}$ is the largest element

$$\max_{k\in\Omega}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right] \ge -\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2 \quad (k\in\Omega), \tag{21}$$

its exponential should be the dominant term of the summation

$$\sum_{k\in\Omega}\exp\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right] \approx \exp\left\{\max_{k\in\Omega}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2s^2\right]\right\}. \tag{22}$$

As a result, the approximation of soft maximum in (20) holds.

Substituting (20) into (18) yields

$$\log P_g \approx \alpha + \beta\cdot\log s + \frac{\gamma}{s^2} \tag{23}$$

where

$$\alpha = \log\frac{\Delta\mathbf{x}}{\left(\sqrt{2\pi}\right)^M} \tag{24}$$

$$\beta = -M \tag{25}$$

$$\gamma = \max_{k\in\Omega}\left[-\left\|\mathbf{x}^{(k)}\right\|_2^2 \Big/ 2\right]. \tag{26}$$

Eq. (23) reveals the important relation between the scaled failure rate P_g and the scaling factor s. The approximation in (23) does not rely on any specific assumptions of the failure region. It is valid, even if the failure region is non-convex or discontinuous.

While (24)-(26) show the theoretical definition of the model coefficients α, β and γ, finding their exact values is not trivial. For instance, the coefficient γ is determined by the hyper-rectangle that falls into the failure region Ω and is closest to the origin $\mathbf{x} = \mathbf{0}$. In practice, without knowing the failure region Ω, we cannot directly find out the value of γ. For this reason, we propose to determine the analytical model in (23) by linear regression. Namely, we first estimate the scaled failure rates $\{P_{g,q};\ q = 1, 2, \ldots, Q\}$ by setting the scaling factor s to a number of different values $\{s_q;\ q = 1, 2, \ldots, Q\}$. As long as the scaling factors $\{s_q;\ q = 1, 2, \ldots, Q\}$ are sufficiently large, the scaled failure rates $\{P_{g,q};\ q = 1, 2, \ldots, Q\}$ are large and can be accurately estimated with a small number of random samples. Next, the model coefficients α, β and γ are fitted by linear regression for the model template in (23) based on the values of $\{(s_q, P_{g,q});\ q = 1, 2, \ldots, Q\}$. Once α, β and γ are known, the original failure rate P_f in (4) can be predicted by *extrapolation*. Namely, we substitute $s = 1$ into the analytical model in (23)

$$\log P_f^{SSS} = \alpha + \gamma \tag{27}$$

where P_f^{SSS} denotes the value of the failure rate P_f estimated by the proposed SSS method. Apply the exponential function to both sides of (27) and we have

$$P_f^{SSS} = \exp(\alpha + \gamma). \tag{28}$$

More details of the proposed SSS algorithm will be further discussed in Section 4.

4. IMPLEMENTATION DETAILS

To make the proposed SSS method of practical utility, a number of efficient algorithms are further studied in this section, including: (i) model fitting via maximum likelihood estimation, and (ii) confidence interval estimation via re-sampling. In what follows, we will discuss these implementation issues in detail.

4.1 Model Fitting via Maximum Likelihood Estimation

While the basic idea of SSS has been illustrated in Section 3, we will develop a statistically optimal algorithm to implement it in this sub-section. Our goal is to determine the maximum likelihood estimation (MLE) for the model coefficients α, β and γ in (23). The MLE solution can be solved from an optimization problem and it is considered to be statistically optimal for a given set of random samples.

Without loss of generality, we assume that N_q scaled random samples are collected for the scaling factor s_q, and the scaled failure rate $P_{g,q}$ is estimated by

$$P_{g,q}^{MC} = \frac{1}{N_q}\cdot\sum_{n=1}^{N_q} I\left[\mathbf{x}^{(n)}\right]. \tag{29}$$

The variance of the estimator $P_{g,q}^{MC}$ in (29) can be approximated as [19]

$$\sigma_{g,q}^2 = \frac{1}{N_q}\cdot P_{g,q}^{MC}\cdot\left(1 - P_{g,q}^{MC}\right). \tag{30}$$

If the number of samples N_q is sufficiently large, the estimator $P_{g,q}^{MC}$ in (29) follows a Normal distribution according to the central limit theorem [19]

$$P_{g,q}^{MC} \sim Gauss\left(P_{g,q}, \sigma_{g,q}^2\right) \tag{31}$$

where $P_{g,q}$ denotes the actual failure rate corresponding to the scaling factor s_q.

To further derive the probability distribution for $\log P_{g,q}^{MC}$, we apply the first-order delta method [19]. Namely, we approximate the nonlinear function log(•) by the first-order Taylor expansion around the mean value $P_{g,q}$ of the random variable $P_{g,q}^{MC}$

$$\log P_{g,q}^{MC} \approx \log P_{g,q} + \frac{P_{g,q}^{MC} - P_{g,q}}{P_{g,q}} \approx \log P_{g,q} + \frac{P_{g,q}^{MC} - P_{g,q}}{P_{g,q}^{MC}}. \tag{32}$$

Based on the linear approximation in (32), $\log P_{g,q}^{MC}$ follows the following Normal distribution

$$\log P_{g,q}^{MC} \sim Gauss\left[\log P_{g,q}, \sigma_{g,q}^2 \Big/ \left(P_{g,q}^{MC}\right)^2\right]. \tag{33}$$

Eq. (33) is valid for all scaling factors $\{s_q;\ q = 1, 2, \ldots, Q\}$. In addition, since the scaled failure rates corresponding to different scaling factors are estimated by independent Monte Carlo simulations, the estimated failure rates $\{P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ are mutually independent. Therefore, the Q-dimensional random

variable

$$\log \mathbf{P}_g^{MC} = \left[\log P_{g,1}^{MC} \quad \log P_{g,2}^{MC} \quad \cdots \quad \log P_{g,Q}^{MC}\right]^T \tag{34}$$

satisfies the following jointly Normal distribution

$$\log \mathbf{P}_g^{MC} \sim Gauss\left(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right) \tag{35}$$

where the mean vector $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$ are equal to

$$\boldsymbol{\mu}_g = \left[\log P_{g,1} \quad \log P_{g,2} \quad \cdots \quad \log P_{g,Q}\right]^T \tag{36}$$

$$\boldsymbol{\Sigma}_g = \mathrm{diag}\left[\sigma_{g,1}^2 \Big/ \left(P_{g,1}^{MC}\right)^2, \sigma_{g,2}^2 \Big/ \left(P_{g,2}^{MC}\right)^2, \cdots, \sigma_{g,Q}^2 \Big/ \left(P_{g,Q}^{MC}\right)^2\right] \tag{37}$$

where diag(•) denotes a diagonal matrix.

Note that the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}_g$ in (37) can be substantially different. In other words, the accuracy of $\{\log P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ associated with different scaling factors $\{s_q;\ q = 1, 2, \ldots, Q\}$ can be different, because the scaled failure rates $\{P_{g,q};\ q = 1, 2, \ldots, Q\}$ strongly depend on the scaling factors. In general, we can expect that if the scaling factor s_q is small, the scaled failure rate $P_{g,q}$ is small and, hence, it would be difficult to accurately estimate $\log P_{g,q}$ from a small number of random samples. For this reason, instead of equally "trusting" the estimators $\{\log P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$, we must carefully model and consider the "confidence" for each $\log P_{g,q}^{MC}$, as encoded by the covariance matrix $\boldsymbol{\Sigma}_g$ in (37). Such "confidence" information will be fully exploited by the MLE framework to fit a statistically optimal model.

Since the scaled failure rates $\{P_{g,q};\ q = 1, 2, \ldots, Q\}$ follow the analytical model in (23), the mean vector $\boldsymbol{\mu}_g$ in (36) can be re-written as

$$\boldsymbol{\mu}_g = \alpha + \beta \cdot \begin{bmatrix} \log s_1 \\ \log s_2 \\ \vdots \\ \log s_Q \end{bmatrix} + \gamma \cdot \begin{bmatrix} s_1^{-2} \\ s_2^{-2} \\ \vdots \\ s_Q^{-2} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{\Theta} \tag{38}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & \log s_1 & s_1^{-2} \\ 1 & \log s_2 & s_2^{-2} \\ \vdots & \vdots & \vdots \\ 1 & \log s_Q & s_Q^{-2} \end{bmatrix} \tag{39}$$

$$\boldsymbol{\Theta} = \left[\alpha \quad \beta \quad \gamma\right]^T. \tag{40}$$

Eq. (38) implies that the mean value of the Q-dimensional random variable $\log \mathbf{P}_g^{MC}$ depends on the model coefficients α, β and γ. Given the failure rates $\{P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ estimated from the scaled random samples, the key idea of MLE is to find the optimal values of α, β and γ so that the likelihood of observing $\{P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ is maximized.

Because the random variable $\log \mathbf{P}_g^{MC}$ follows the jointly Normal distribution in (35), the likelihood associated with the estimated failure rates $\{P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ is proportional to

$$L \sim \exp\left[-\left(\log \mathbf{P}_g^{MC} - \boldsymbol{\mu}_g\right)^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \left(\log \mathbf{P}_g^{MC} - \boldsymbol{\mu}_g\right)\right]. \tag{41}$$

Taking the logarithm for (41), the log-likelihood is proportional to

$$\log L \sim -\left(\log \mathbf{P}_g^{MC} - \boldsymbol{\mu}_g\right)^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \left(\log \mathbf{P}_g^{MC} - \boldsymbol{\mu}_g\right). \tag{42}$$

Substitute (38) into (42), and we have

$$\log L \sim -\left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right)^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right). \tag{43}$$

Note that the log-likelihood $\log L$ in (43) depends on the model coefficients α, β and γ, because the vector $\boldsymbol{\Theta}$ is composed of these coefficients as shown in (40). Therefore, the MLE solution of α, β and γ can be determined by maximizing the log-likelihood function

$$\underset{\boldsymbol{\Theta}}{\text{maximize}} \quad -\left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right)^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right). \tag{44}$$

Since the covariance matrix $\boldsymbol{\Sigma}_g$ is positive definite, the optimization in (44) is convex. In addition, since the log-likelihood $\log L$ is simply a quadratic function of the unknown vector $\boldsymbol{\Theta}$, the unconstrained optimization in (44) can be directly solved by inspecting the first-order optimality condition [17]

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{\Theta}}\left[-\left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right)^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right)\right] \\ &= 2 \cdot \mathbf{A}^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \left(\log \mathbf{P}_g^{MC} - \mathbf{A} \cdot \boldsymbol{\Theta}\right) = 0 \end{aligned} \tag{45}$$

Based on the linear equation in (45), the optimal value of $\boldsymbol{\Theta}$ can be determined by

$$\boldsymbol{\Theta} = \left(\mathbf{A}^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \boldsymbol{\Sigma}_g^{-1} \cdot \log \mathbf{P}_g^{MC}. \tag{46}$$

Eq. (46), where $\boldsymbol{\Theta} = [\alpha\ \beta\ \gamma]^T$, represents the MLE solution of the model coefficients α, β and γ solved from the estimated failure rates $\{(s_q, P_{g,q}^{MC});\ q = 1, 2, \ldots, Q\}$.

Studying (46) reveals an important fact that the estimators $\{\log P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ are weighted by the inverse of the covariance matrix $\boldsymbol{\Sigma}_g$. Namely, if the variance of the estimator $\log P_{g,q}^{MC}$ is large, $\log P_{g,q}^{MC}$ becomes non-critical when determining the optimal values of α, β and γ. In other words, the MLE framework has already optimally weighted the importance of $\{\log P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ based on the "confidence" level of these estimators. Once the model coefficients α, β and γ are solved by MLE, the original failure rate P_f can be directly estimated by (28).

4.2 Confidence Interval Estimation via Re-sampling

While the MLE algorithm has been discussed in the previous sub-section to optimally estimate the model coefficients α, β and γ, and then approximate the original failure rate P_f in (4) by the proposed estimator P_f^{SSS} in (28), it remains an open question how to quantitatively assess the accuracy of our SSS method. Since SSS is based upon Monte Carlo simulation, a natural way for accuracy assessment is to calculate the confidence interval of the estimator P_f^{SSS}. However, unlike the traditional estimator where a statistical metric is estimated by the average of multiple random samples and, hence, the confidence interval can be derived as a closed-form expression, our proposed estimator P_f^{SSS} is calculated by linear regression with nonlinear exponential/logarithmic transformation, as shown in Section 4.1. Accurately estimating the confidence interval of P_f^{SSS} is not a trivial task.

In this paper, we propose a novel re-sampling method to address the aforementioned challenge. Our key idea is to re-generate a large number of random samples based on a statistical model without running transistor-level simulations. These random samples are then used to repeatedly calculate the value of P_f^{SSS} in (28) for multiple times. Based on these repeated runs, the statistics (hence, the confidence interval) of the estimator P_f^{SSS} can be accurately estimated.

In particular, we start from the estimated failure rates $\{P_{g,q}^{MC};\ q = 1, 2, \ldots, Q\}$ in (29). Each estimator $P_{g,q}^{MC}$ follows the Normal distribution in (31) where the variance $\sigma_{g,q}^2$ is estimated by (30). The exact mean $P_{g,q}$ in (31) is unknown; however, we can approximate its value by the estimated failure rate $P_{g,q}^{MC}$. Since we know the statistical distribution of $P_{g,q}^{MC}$, we can re-sample its distribution and generate N_{RS} sampled values $\{P_{g,q}^{MC(n)};\ n = 1,$

2, …, N_{RS}}. This re-sampling idea is applied to all scaling factors {s_q; q = 1, 2, …, Q}, thereby resulting in a large data set {$P_{g,q}^{MC(n)}$; q = 1, 2, …, Q, n = 1, 2, …, N_{RS}}. Next, we repeatedly run SSS for N_{RS} times and get N_{RS} different failure rates {$P_f^{SSS(n)}$; n = 1, 2, …, N_{RS}}. The confidence interval of P_f^{SSS} can then be easily estimated from the statistics of these failure rate values.

There are two important properties that we should emphasize for our proposed confidence interval estimation based on re-sampling. First, the aforementioned re-sampling process does not require any transistor-level simulations. Hence, it is computationally efficient. Second, the only assumption required by our re-sampling method lies in the Normal distribution in (31). The validness of this assumption is guaranteed by the central limit theorem, as long as the number of samples N_q is sufficiently large in (30). For this reason, the proposed re-sampling method can be generally applied to a broad range of application cases with high accuracy, as will be demonstrated by the numerical examples in Section 5.

4.3 Summary

Algorithm 1: Scaled-Sigma Sampling (SSS)

1. Start from a set of pre-selected scaling factors {s_q; q = 1, 2, …, Q}.
2. For each scaling factor s_q where $q \in$ {1, 2, …, Q}, sample the scaled PDF $g(\mathbf{x})$ in (10) by setting $s = s_q$, generate N_q scaled random samples by running transistor-level simulations, and calculate the scaled failure rate $P_{g,q}^{MC}$ by (29).
3. For each estimator $P_{g,q}^{MC}$ where $q \in$ {1, 2, …, Q}, calculate its variance $\sigma_{g,q}^2$ by (30).
4. Form the Q-dimensional vector $\log \mathbf{P}_g^{MC}$ by taking the logarithm for the estimated failure rates {$P_{g,q}^{MC}$; q = 1, 2, …, Q}, as shown in (34).
5. Form the diagonal matrix $\mathbf{\Sigma}_g$ in (37) and the matrix $\mathbf{A}$ in (39).
6. Calculate the MLE solution $\mathbf{\Theta}$ based on (46), where the vector $\mathbf{\Theta}$ is composed of the model coefficients α, β and γ as shown in (40).
7. Approximate the failure rate of interest P_f by the estimator P_f^{SSS} in (28).
8. For each estimator $P_{g,q}^{MC}$ where $q \in$ {1, 2, …, Q}, re-sample the Normal distribution in (31) for which the exact mean $P_{g,q}$ is approximated as the estimated value $P_{g,q}^{MC}$, and generate N_{RS} re-sampled values {$P_{g,q}^{MC(n)}$; n = 1, 2, …, N_{RS}}.
9. For each data set generated by re-sampling {$P_{g,q}^{MC(n)}$; q = 1, 2, …, Q} where $n \in$ {1, 2, …, N_{RS}}, repeat Step 3~7 to calculate the failure rate $P_f^{SSS(n)}$.
10. Based on the data set {$P_f^{SSS(n)}$; n = 1, 2, …, N_{RS}}, estimate the confidence interval of the estimator P_f^{SSS}.

Algorithm 1 summarizes the simplified flow of the proposed SSS algorithm. It assumes that a set of pre-selected scaling factors {s_q; q = 1, 2, …, Q} are already given. In practice, appropriately choosing these scaling factors is a critical task. On one hand, if these scaling factors are too large, the estimator P_f^{SSS} based on extrapolation in (28) would not be highly accurate, since the extrapolation point s = 1 is far away from the selected scaling factors. On the other hand, if the scaling factors are too small, the scaled failure rates {$P_{g,q}$; q = 1, 2, …, Q} are extremely small and they cannot be accurately estimated from a small number of scaled random samples (i.e., a small number of transistor-level simulations). In this case, the estimator P_f^{SSS} cannot be efficiently implemented with low computational cost. In this paper, the scaling factors are empirically selected and provided as the input of Algorithm 1. In our future research, we will study efficient methodologies to further optimize these scaling factors and improve the efficacy of SSS.

Finally, it is important to mention that while most traditional statistical methods (e.g., importance sampling) cannot efficiently estimate the rare failure rate in a high-dimensional variation space, the proposed SSS algorithm does not suffer from such a dimensionality problem. None of the steps in Algorithm 1 is sensitive to the dimensionality of the variation space. As will be demonstrated by the numerical examples in Section 5, SSS achieves superior accuracy over the traditional importance sampling method where the dimensionality of the variation space exceeds a few hundred.

5. NUMERICAL EXAMPLES

In this section, two circuit examples are used to demonstrate the efficacy of the proposed SSS method. For testing and comparison purposes, three different approaches are implemented: (i) the brute-force Monte Carlo analysis, (ii) the minimum-norm importance sampling (MNIS) [10], and (iii) the proposed SSS method. The brute-force Monte Carlo analysis is used to generate the "golden" failure rates so that the accuracy of MNIS and SSS can be quantitatively evaluated. The implementation of MNIS consists of two stages, as described in [10]. In the first stage, 2000 transistor-level simulations are used to search the variation space and find the failure event that is most likely to occur. Next, importance sampling is applied with a shifted Normal distribution to estimate the rare failure rate. Finally, when implementing the proposed SSS method, five different scaling factors are empirically chosen to estimate the failure rate and 200 re-sampled data points are generated to estimate the confidence interval (i.e., Q = 5 and N_{RS} = 200) in Algorithm 1. All numerical experiments are run on a 3.16GHz computer with 12GB memory.

5.1 SRAM Read Current

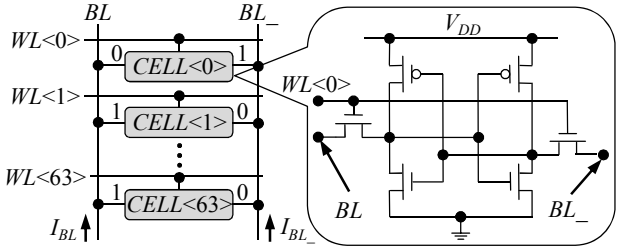


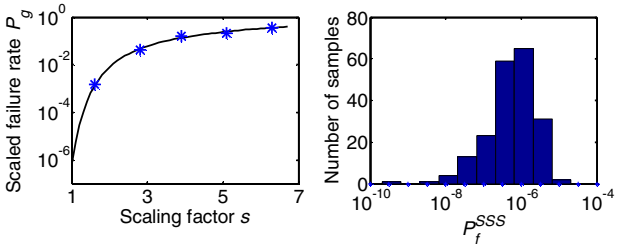
Figure 2. The simplified schematic is shown for an SRAM column designed in a 45nm CMOS process where the read current is defined as the difference between two bit-line currents: $I_{READ} = I_{BL} - I_{BL\_}$.

Shown in Figure 2 is the simplified schematic of an SRAM column designed in a 45nm CMOS process. It consists of 64 bit-cells that are connected to two bit-lines: *BL* and *BL_*. When reading the first bit-cell *CELL*<0>, we first pre-charge both bit-lines to the supply voltage V_{DD}. Next, the word-line *WL*<0> is turned on and *CELL*<0> is activated. All other word-lines are turned off so that the corresponding bit-cells are de-activated. The current from *CELL*<0> then discharges the bit-lines and creates a voltage difference between *BL* and *BL_*.

To mimic the worst-case scenario for read operation, we store "ZERO" in *CELL*<0> and "ONE" in all other bit-cells so that the magnitude of leakage current is maximized. In this example, the

read current $I_{READ} = I_{BL} - I_{BL\_}$ is our performance of interest. It directly impacts the read delay and, therefore, is an important performance metric. If I_{READ} is greater than a pre-defined specification, the SRAM circuit is considered as "PASS". Otherwise, it is considered as "FAIL".

To consider process variations in our experiment, we model the local V_{TH} mismatch of each transistor as an independent Normal random variable. Since one SRAM column consists of 64 bit-cells and each bit-cell is composed of 6 transistors, there are 384 transistors and, hence, 384 Normal random variables in total. It, in turn, renders a high-dimensional variation space for this SRAM example.



(a) (b)

Figure 3. (a) The scaled failure rate P_g is plotted as a function of the scaling factor s where the blue stars represent five empirically selected scaling factors and the estimated failure rates corresponding to these scaling factors. (b) A histogram is generated by 200 re-sampled data points to estimate the confidence interval of the estimator P_f^{SSS}.

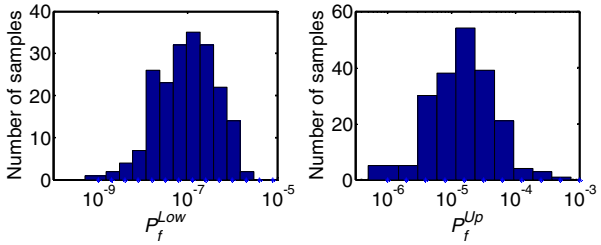
Table 1. Failure rate and 95% confidence interval $[P_f^{Low}, P_f^{Up}]$ estimated by MNIS and SSS ("golden" failure rate = 10^{-6})

# of Sims	MNIS [10]			SSS (Proposed)		
	P_f^{Low}	P_f^{MNIS}	P_f^{Up}	P_f^{Low}	P_f^{SSS}	P_f^{Up}
4×10^3	0	7.9×10^{-11}	2.1×10^{-10}	1.0×10^{-8}	1.1×10^{-6}	6.1×10^{-5}
5×10^3	0	3.5×10^{-11}	7.6×10^{-11}	8.6×10^{-9}	7.5×10^{-7}	1.6×10^{-5}
6×10^3	0	1.8×10^{-10}	4.0×10^{-10}	6.1×10^{-8}	2.3×10^{-6}	2.2×10^{-5}
7×10^3	0	3.3×10^{-11}	6.6×10^{-11}	4.2×10^{-8}	1.2×10^{-6}	1.5×10^{-5}
8×10^3	0	5.9×10^{-9}	1.5×10^{-8}	2.3×10^{-8}	5.8×10^{-7}	7.2×10^{-6}
9×10^3	0	1.5×10^{-10}	3.6×10^{-10}	1.6×10^{-7}	2.0×10^{-6}	1.9×10^{-5}
1×10^4	2.7×10^{-11}	2.2×10^{-10}	4.2×10^{-10}	1.9×10^{-8}	7.3×10^{-7}	5.6×10^{-6}

We first apply a brute-force Monte Carlo analysis with 10^7 random samples to predict the failure rate of the read current. The failure rate estimated by the brute-force Monte Carlo analysis is 10^{-6}. It is considered as the "golden" result to compare the accuracy of other statistical methods in our experiment. Next, we apply the proposed SSS method (i.e., Algorithm 1) to estimate the failure rate. Figure 3(a) shows five empirically selected scaling factors $\{s_q;\ q = 1, 2, \ldots, 5\}$ and their corresponding scaled failure rates $\{P_{g,q}^{MC};\ q = 1, 2, \ldots, 5\}$ estimated by (29). In total, 10^4 transistor-level simulations are used to generate these five data points $\{(s_q, P_{g,q});\ q = 1, 2, \ldots, 5\}$. The black curve in Figure 3(a) further shows the analytical model in (23) that is optimally fitted by MLE. Next, the SRAM failure rate is predicted by the estimator P_f^{SSS} in (28) based on extrapolation at $s = 1$. Figure 3(b) further shows the histogram generated by re-sampling, as described in Algorithm 1. It is calculated from 200 re-sampled data points, and is used to estimate the confidence interval of the estimator P_f^{SSS}. In our experiment, we notice that the computational cost of SSS is completely dominated by the transistor-level simulations required to generate the random samples. The computational time of processing the sampling data by Algorithm 1 takes less than 0.5 second and, hence, is negligible.

Table 1 compares the failure rate and the 95% confidence interval estimated by MNIS [10] and SSS based on different numbers of transistor-level simulations. Studying Table 1 reveals two important observations. First, the failure rate estimated by MNIS is substantially different from the golden result (i.e., 10^{-6}) estimated by 10^7 brute-force Monte Carlo samples. We believe that MNIS fails to identify the critical failure region that is most likely to occur, since the variation space is high-dimensional (i.e., consisting of 384 independent random variables) in this example. Therefore, the importance sampling implemented at the second stage of MNIS fails to estimate the failure rate accurately. On the other hand, the proposed SSS method successfully estimates the failure rate even if the number of transistor-level simulations is as small as 4×10^3.

Second, but more importantly, the 95% confidence interval estimated by MNIS is not accurate either. As shown in Table 1, MNIS does not result in a confidence interval $[P_f^{Low}, P_f^{Up}]$ that overlaps with the golden failure rate (i.e., 10^{-6}). In other words, the confidence interval estimated by MNIS based on importance sampling is highly biased. It is one of the major limitations of MNIS and, in general, the importance sampling technique. Namely, as the confidence interval is inaccurate, it provides a wrong assessment of the accuracy and may completely misguide the user in practical applications. On the other hand, the proposed SSS method is unbiased in both failure rate and confidence interval and, hence, is of great practical utility.



(a) (b)

Figure 4. Histograms of the lower and upper bounds of the 95% confidence interval $[P_f^{Low}, P_f^{Up}]$ are estimated from 200 repeated runs: (a) the lower bound P_f^{Low}, and (b) the upper bound P_f^{Up}.

Finally, in order to further validate the confidence interval estimated by SSS, we repeatedly run Algorithm 1 for 200 times. At each run, the failure rate and the corresponding 95% confidence interval $[P_f^{Low}, P_f^{Up}]$ are estimated from 10^4 transistor-level simulations, resulting in 200 different values for both P_f^{Low} and P_f^{Up}. Figure 4(a) and (b) show the histograms of these 200 values for P_f^{Low} and P_f^{Up}, respectively. For only 10 runs out of 200 runs in total, the 95% confidence interval $[P_f^{Low}, P_f^{Up}]$ does not overlap with the golden failure rate (i.e., 10^{-6}), as shown in Figure 4. In other words, the probability for the golden failure rate to fall out of the estimated confidence interval is exactly 10/200 = 5%. It, in turn, demonstrates that our proposed confidence interval estimation based on re-sampling (i.e., Algorithm 1) is highly accurate and it is practically more attractive than the traditional MNIS method based on importance sampling.

5.2 DFF Delay

Shown in Figure 5 is the simplified circuit schematic for a DFF designed in a commercial 32nm CMOS process. The DFF consists of 20 transistors and the random mismatch of each

transistor is modeled by 14 independent random variables in the process design kit. Thus, there are 280 independent random variables in total. In this example, we consider the delay from the clock signal *CLK* to the output *Q* as the performance of interest. In particular, there are two different delay metrics: (i) the delay from *CLK* to *Q* when the input data *D* is "ZERO", and (ii) the delay from *CLK* to *Q* when the input data *D* is "ONE". Both delay values must be less than the pre-defined specifications so that the DFF circuit is considered as "PASS". Otherwise, the DFF circuit is considered as "FAIL".

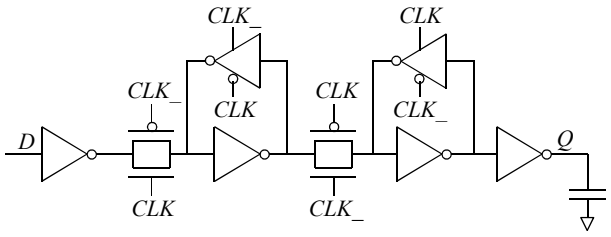


Figure 5. The simplified schematic is shown for a DFF designed in a 32nm CMOS process where the delay from the clock signal *CLK* to the output *Q* is considered as the performance of interest.

Table 2. Failure rate and 95% confidence interval $[P_f^{Low}, P_f^{Up}]$ estimated by MNIS and SSS ("golden" failure rate = 10^{-5})

# of Sims	MNIS [10]			SSS (Proposed)		
	P_f^{Low}	P_f^{MNIS}	P_f^{Up}	P_f^{Low}	P_f^{SSS}	P_f^{Up}
4×10^3	0	5.2×10^{-6}	1.5×10^{-5}	4.3×10^{-7}	2.6×10^{-5}	5.2×10^{-4}
5×10^3	0	3.3×10^{-6}	9.6×10^{-6}	1.7×10^{-6}	3.0×10^{-5}	2.6×10^{-4}
6×10^3	0	2.2×10^{-7}	5.8×10^{-7}	8.3×10^{-7}	1.3×10^{-5}	1.1×10^{-4}
7×10^3	0	1.7×10^{-7}	4.6×10^{-7}	1.9×10^{-7}	3.6×10^{-6}	2.3×10^{-5}
8×10^3	0	1.8×10^{-7}	4.3×10^{-7}	5.2×10^{-6}	6.9×10^{-5}	5.6×10^{-4}
9×10^3	0	1.6×10^{-7}	3.7×10^{-7}	1.0×10^{-6}	1.0×10^{-5}	5.5×10^{-5}
1×10^4	0	1.5×10^{-7}	3.4×10^{-7}	1.7×10^{-6}	1.6×10^{-5}	8.9×10^{-5}

We first run a brute-force Monte Carlo analysis with 10^7 random samples and the estimated failure rate is 10^{-5}. Table 2 compares the failure rate and the 95% confidence interval estimated by MNIS [10] and SSS. Similar to the SRAM example in the previous sub-section, MNIS cannot predict the failure rate or the confidence interval accurately. On the other hand, the proposed SSS method estimates both of them accurately, even if the number of simulations is as small as 4×10^3. From this point of view, the DFF example again demonstrates that our proposed SSS method is superior over the traditional MNIS approach in a high-dimensional variation space.

6. CONCLUSIONS

In this paper, a novel statistical analysis method, referred to as SSS, is developed to accurately estimate the rare failure rates for nanoscale ICs in high-dimensional variation space. The proposed SSS approach is based upon an analytical model derived from the theorem of "soft maximum". It is statistically formulated as a regression modeling problem and optimally solved by MLE. Our numerical experiments demonstrate that SSS achieves superior accuracy over the traditional importance sampling technique when the dimensionality of the variation space is more than a few hundred. In addition, the proposed SSS method can be easily extended to estimate the rare failure rate as a function of the performance specification (i.e., the tail of the cumulative distribution function, instead of a single failure rate only) without running any additional transistor-level simulations. Finally, even though we assume that process variations are modeled as a multivariate Normal distribution in this paper, SSS can be possibly extended to handle other statistical distributions (i.e., uniform distribution). More details along these directions will be studied in our future work.

7. ACKNOWLEDGEMENTS

This work has been supported in part by the National Science Foundation under contract CCF–1016890.

8. REFERENCES

[1] B. Calhoun, Y. Cao, X. Li, K. Mai, L. Pileggi, R. Rutenbar, and K. Shepard, "Digital circuit design challenges and opportunities in the era of nanoscale CMOS," *Proc. IEEE*, vol. 96, no. 2, pp. 343-365, Feb. 2008.

[2] A. Bhavnagarwala, X. Tang and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE JSSC*, vol. 36, pp. 658-665, Apr. 2001.

[3] R. Heald and P. Wang, "Variability in sub-100nm SRAM designs," *IEEE ICCAD*, pp. 347-352, 2004.

[4] 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," *IEEE DAC*, pp. 69-72, 2006.

[5] 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," *IEEE ASP-DAC*, pp. 754-761, 2008.

[6] M. Abu-Rahma, K. Chowdhury, J. Wang, Z. Chen, S. Yoon and M. Anis, "A methodology for statistical estimation of read access yield in SRAMs," *IEEE DAC*, pp. 205-210, 2008.

[7] L. Dolecek, M. Qazi, D. Shah and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," *IEEE ICCAD*, pp. 322-329, 2008.

[8] J. Wang, S. Yaldiz, X. Li and L. Pileggi, "SRAM parametric failure analysis," *IEEE DAC*, pp. 496-501, 2009.

[9] A. Singhee and R. Rutenbar, "Statistical blockade: very fast statistical simulation and modeling of rare circuit events, and its application to memory design," *IEEE Trans. on CAD*, vol. 28, no. 8, pp. 1176-1189, Aug. 2009.

[10] M. Qazi, M. Tikekar, L. Dolecek, D. Shah and A. Chandrakasan, "Loop flattening and spherical sampling: highly efficient model reduction techniques for SRAM yield analysis," *IEEE DATE*, pp. 801-806, 2010.

[11] R. Fonseca, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel and N. Badereddine, "A statistical simulation method for reliability analysis of SRAM core-cells," *IEEE DAC*, pp. 853-856, 2010.

[12] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi and T. Sato, "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis," *IEEE ICCAD*, pp. 703-708, 2010.

[13] R. Kanj, R. Joshi, Z. Li, J. Hayes and S. Nassif, "Yield estimation via multi-cones," *IEEE DAC*, pp. 1107-1112, 2012.

[14] S. Sun, Y. Feng, C. Dong and X. Li, "Efficient SRAM failure rate prediction via Gibbs sampling," *IEEE Trans. on CAD*, vol. 31, no. 12, pp. 1831-1844, Dec. 2012.

[15] R. Joshi, S. Mukhopadhyay, D. Plass, Y. Chan, C. Chuang and Y. Tan, "Design of sub-90nm low-power and variation tolerant PD/SOI SRAM cell based on dynamic stability metrics," *IEEE JSSC*, vol. 44, no. 3, pp. 965-976, Mar. 2009.

[16] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, Addison-Wesley, 2010.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2009.

[18] C. Bishop, *Pattern Recognition and Machine Learning*, Prentice Hall, 2007.

[19] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Process*, McGraw-Hill, 2001.