Chapter 12
Fast Statistical Analysis of Rare Circuit Failure Events

Jun Tao, Shupeng Sun, Xin Li, Hongzhou Liu, Kangsheng Luo, Ben Gu, and Xuan Zeng

12.1 Introduction

As integrated circuit (IC) technology advances, the ever increasing process variation has become a growing concern [5]. A complex IC, containing numerous memory components, is required to meet the design specification not only at the nominal process corner, but also under large-scale process variations. To achieve sufficiently high yield, the failure rate of each individual memory component must be extremely small. For instance, the failure rate of an SRAM bit-cell must be less than

J. Tao (✉) · X. Zeng (✉)
State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai, China
e-mail: taojun@fudan.edu.cn; xzeng@fudan.edu.cn

S. Sun
Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: shupengs@ece.cmu.edu

X. Li (✉)
Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
e-mail: xinli.ece@duke.edu

H. Liu · K. Luo
Cadence Design Systems, Inc., Pittsburgh, PA, USA
e-mail: hliu@cadence.com; ksluo@cadence.com

B. Gu
Cadence Design Systems, Inc., Austin, TX, USA
e-mail: gxin@cadence.com

I. M. Elfadel et al. (eds.), *Machine Learning in VLSI Computer-Aided Design*,
https://doi.org/10.1007/978-3-030-04666-8_12

$10^{-8}\sim10^{-6}$ for a typical SRAM design [2, 12]. Due to this reason, efficiently analyzing the rare failure event for the individual memory component becomes an important task for the IC design community.

The simple way to estimate the failure probability is to apply the well-known crude Monte Carlo (CMC) technique [3]. CMC directly draws random samples from the probability density function (PDF) that models device-level variations, and performs a transistor-level simulation to evaluate the performance value for each sample. When CMC is applied to estimate an extremely small failure rate (e.g., $10^{-8}\sim10^{-6}$), most random samples do not fall into the failure region. Hence, a large number of (e.g., $10^{7}\sim10^{9}$) samples are needed to accurately estimate the small failure probability, which implies that CMC can be extremely expensive for our application of rare failure rate estimation.

To improve the sampling efficiency, importance sampling (IS) methods have been proposed in the literature [7, 13, 15, 17, 20]. Instead of sampling the original PDF, IS samples a distorted PDF to get more samples in the important failure region. The efficiency achieved by IS highly depends on the choice of the distorted PDF. The traditional IS methods apply several heuristics to construct a distorted PDF that can capture the most important failure region in the variation space. Such a goal, though easy to achieve in a low-dimensional variation space, is extremely difficult to fulfill when a large number of random variables are used to model process variations.

Another approach to improving the sampling efficiency, referred to as statistical blockade, has recently been proposed [18]. This approach first builds a classifier with a number of transistor-level simulations, and then draws random samples from the original PDF. Unlike CMC where all the samples are evaluated by transistor-level simulations, statistical blockade only simulates the samples that are likely to fall into the failure region or close to the failure boundary based on the classifier. The efficiency achieved by this approach highly depends on the accuracy of the classifier. If the variation space is high-dimensional, a large number of transistor-level simulations are needed to build an accurate classifier, which makes the statistical blockade method quickly intractable.

In addition to the aforementioned statistical methods, several deterministic approaches have also been proposed to efficiently estimate the rare failure probability [10, 14]. These methods first find the failure boundary, and then calculate the failure probability by integrating the PDF over the failure region in the variation space. Though efficient in a low-dimensional variation space, it is often computationally expensive to accurately determine the failure boundary in a high-dimensional space especially if the boundary has a complicated shape (e.g., non-convex or even discontinuous).

Most of these traditional methods [7, 9, 10, 13–15, 17, 18, 20, 22, 23] have been successfully applied to SRAM bit-cells to estimate their rare failure rates where only a small number of (e.g., $6\sim20$) independent random variables are used to model process variations and, hence, the corresponding variation space is low-dimensional. It has been demonstrated in the literature that estimating the rare failure probability

in a high-dimensional space (e.g., hundreds of independent random variables to model the device-level variations for SRAM) becomes increasingly important [21]. Unfortunately, such a high-dimensional problem cannot be efficiently handled by most traditional methods. It, in turn, poses an immediate need of developing a new CAD tool to accurately capture the rare failure events in a high-dimensional variation space with low computational cost.

To address this technical challenge, we first describe a novel subset simulation (SUS) technique. The key idea of SUS, borrowed from the statistics community [1, 6, 11], is to express the rare failure probability as the product of several large conditional probabilities by introducing a number of intermediate failure events. As such, the original problem of rare failure probability estimation is cast to an equivalent problem of estimating a sequence of conditional probabilities via multiple phases. Since these conditional probabilities are relatively large, they are substantially easier to estimate than the original rare failure rate.

When implementing the SUS method, it is difficult, if not impossible, to directly draw random samples from the conditional PDFs and estimate the conditional probabilities, since these conditional PDFs are unknown in advance. To address this issue, a modified Metropolis (MM) algorithm is adopted from the literature [1] to generate random samples by constructing a number of Markov chains. The conditional probabilities of interest are then estimated from these random samples. Unlike most traditional techniques [7, 9, 10, 13–15, 17, 18, 20, 22, 23] that suffer from the dimensionality issue, SUS can be efficiently applied to high-dimensional problems, which will be demonstrated by the experimental results in Sect. 12.2.

To define the intermediate failure events required by SUS, the performance of interest (PoI) must be continuous. In other words, SUS can only analyze a continuous PoI. For many rare failure events, however, PoIs are discrete (e.g., the output of a voltage-mode sense amplifier). Realizing this limitation, we further describe a scaled-sigma sampling (SSS) approach to efficiently estimate the rare failure rates for discrete PoIs in a high-dimensional space. SSS is particularly developed to address the following two fundamental questions: (1) how to efficiently draw random samples from the rare failure region, and (2) how to estimate the rare failure rate based on these random samples. Unlike CMC that directly samples the variation space and therefore only few samples fall into the failure region, SSS draws random samples from a distorted 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 derived from the theorem of "soft maximum" is optimally fitted by applying maximum likelihood estimation (MLE). Next, the failure rate can be efficiently estimated from the fitted model.

The remainder of this chapter is organized as follows. In Sect. 12.2, we will summarize the SUS approach and, next, the SSS approach will be presented in Sect. 12.3. Finally, we conclude in Sect. 12.4.

12.2 Subset Simulation

Suppose that the vector

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_M \end{bmatrix}^T \tag{12.1}$$

is an M-dimensional random variable modeling device-level process variations. In a process design kit, the random variables $\{x_m; m = 1, 2, \ldots, M\}$ in (12.1) are typically modeled as a jointly Normal distribution [7, 9, 10, 13–15, 17, 18, 20, 22, 23]. Without loss of generality, we further assume that $\{x_m; m = 1, 2, \ldots, M\}$ are mutually independent and standard Normal (i.e., with zero mean and unit variance) and its joint PDF is

$$f(\mathbf{x}) = \prod_{m=1}^{M} p_m(x_m) = \prod_{m=1}^{M} \left[\frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{x_m^2}{2} \right) \right] = \frac{\exp\left(-\|\mathbf{x}\|_2^2 \big/ 2 \right)}{\left(\sqrt{2\pi} \right)^M}, \tag{12.2}$$

where $p_m(x_m)$ is the 1-D PDF for x_m, and $\| \bullet \|_2$ 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 [3]. Then, the failure rate of a circuit can be mathematically represented as:

$$P_F = \Pr(\mathbf{x} \in \Omega) = \int_{\mathbf{x} \in \Omega} f(\mathbf{x}) \cdot d\mathbf{x}, \tag{12.3}$$

where Ω denotes the failure region, i.e., the subset of the variation space where the PoI does not meet the specification.

Instead of directly estimating the rare failure probability P_F, SUS expresses P_F as the product of several large conditional probabilities by introducing a number of intermediate failure events in the variation space. Without loss of generality, we define K intermediate failure events $\{\Omega_k; k = 1, 2, \ldots, K\}$ as:

$$\Omega_1 \supset \Omega_2 \supset \cdots \supset \Omega_{K-1} \supset \Omega_K = \Omega. \tag{12.4}$$

Based on (12.4), we can express P_F in (12.3) as:

$$P_F = \Pr(\mathbf{x} \in \Omega) = \Pr\left(\mathbf{x} \in \Omega_K, \mathbf{x} \in \Omega_{K-1} \right). \tag{12.5}$$

Equation (12.5) can be re-written as:

$$P_F = \Pr\left(\mathbf{x} \in \Omega_K \,\middle|\, \mathbf{x} \in \Omega_{K-1} \right) \cdot \Pr\left(\mathbf{x} \in \Omega_{K-1} \right). \tag{12.6}$$

Similarly, we can express $\Pr\left(\mathbf{x} \in \Omega_{K-1}\right)$ as:

$$\Pr\left(\mathbf{x} \in \Omega_{K-1}\right) = \Pr\left(\mathbf{x} \in \Omega_{K-1} \left| \mathbf{x} \in \Omega_{K-2}\right.\right) \cdot \Pr\left(\mathbf{x} \in \Omega_{K-2}\right). \tag{12.7}$$

From (12.4), (12.6), and (12.7), we can easily derive:

$$P_F = \Pr\left(\mathbf{x} \in \Omega_1\right) \cdot \prod_{k=2}^{K} \Pr\left(\mathbf{x} \in \Omega_k \left| \mathbf{x} \in \Omega_{k-1}\right.\right) = \prod_{k=1}^{K} P_k, \tag{12.8}$$

where

$$P_1 = \Pr\left(\mathbf{x} \in \Omega_1\right), \tag{12.9}$$

$$P_k = \Pr\left(\mathbf{x} \in \Omega_k \left| \mathbf{x} \in \Omega_{k-1}\right.\right) \quad (k = 2, 3, \ldots, K). \tag{12.10}$$

If $\{\Omega_k; k = 1, 2, \ldots, K\}$ are properly chosen, all the probabilities $\{P_k; k = 1, 2, \ldots, K\}$ are large and can be efficiently estimated. Once $\{P_k; k = 1, 2, \ldots, K\}$ are known, the rare failure probability P_F can be easily calculated by (12.8).

Note that the failure events $\{\Omega_k; k = 1, 2, \ldots, K\}$ are extremely difficult to specify in a high-dimensional variation space. For this reason, we do not directly define $\{\Omega_k; k = 1, 2, \ldots, K\}$ in the variation space. Instead, we utilize their corresponding subsets $\{F_k; k = 1, 2, \ldots, K\}$ in the performance space:

$$F_k = \left\{y(\mathbf{x}); \mathbf{x} \in \Omega_k\right\} \quad (k = 1, 2, \ldots, K), \tag{12.11}$$

where $y(\mathbf{x})$ denotes the PoI as a function of $\mathbf{x}$. Since $y(\mathbf{x})$ is typically a scalar, $\{F_k; k = 1, 2, \ldots, K\}$ are just one-dimensional subsets of $\mathbb{R}$ and, therefore, easy to be specified. Once $\{F_k; k = 1, 2, \ldots, K\}$ are determined, $\{\Omega_k; k = 1, 2, \ldots, K\}$ are implicitly known. For instance, to know whether a given $\mathbf{x}$ belongs to Ω_k, we first run a transistor-level simulation to evaluate $y(\mathbf{x})$. If $y(\mathbf{x})$ belongs to F_k, $\mathbf{x}$ is inside Ω_k. Otherwise, $\mathbf{x}$ is outside Ω_k.

In what follows, we will use a simple 2-D example to intuitively illustrate the basic flow of SUS. Figure 12.1 shows this 2-D example where two random variables $\mathbf{x} = [x_1\ x_2]^T$ are used to model the device-level process variations, and Ω_1 and Ω_2 denote the first two subsets in (12.4). Note that Ω_1 and Ω_2 are depicted for illustration purposes in this example. In practice, we do not need to explicitly know Ω_1 and Ω_2, as previously explained.

Our objective is to estimate the probabilities $\{P_k; k = 1, 2, \ldots, K\}$ via multiple phases. Starting from the 1st phase, we simply draw L_1 independent random samples $\{\mathbf{x}^{(1,l)}; l = 1, 2, \ldots, L_1\}$ from the PDF $f(\mathbf{x})$ to estimate P_1. Here, the superscript "1" of the symbol $\mathbf{x}^{(1,l)}$ refers to the 1st phase. Among these L_1 samples, we identify a subset of samples $\{\mathbf{x}_F^{(1,t)}; t = 1, 2, \ldots, T_1\}$ that fall into Ω_1, where T_1 denotes the total number of samples in this subset. As shown in Fig. 12.1(a), the red

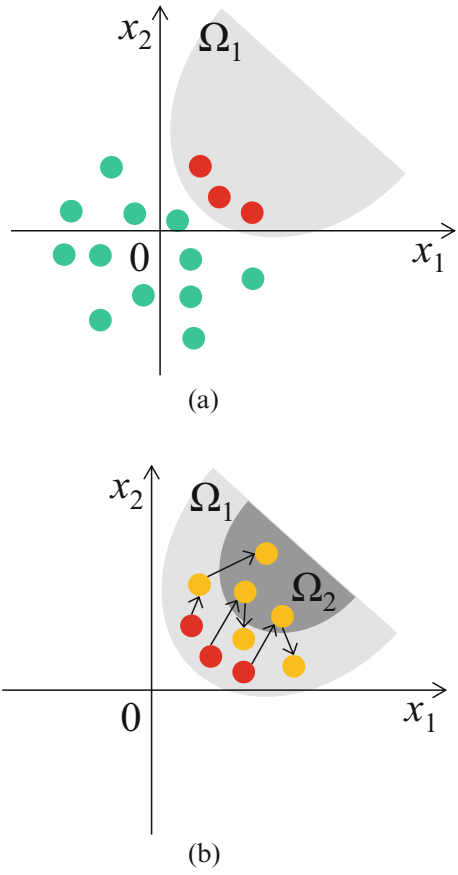

Fig. 12.1 A 2-D example is used to illustrate the procedure of probability estimation via multiple phases by using SUS: (**a**) generating MC samples and estimating P_1 in the 1st phase, and (**b**) generating MCMC samples and estimating P_2 in the 2nd phase

points represent the samples that belong to Ω_1 and the green points represent the samples that are out of Ω_1. In this case, P_1 can be estimated as:

$$P_1^{\mathrm{SUS}} = \frac{1}{L_1} \cdot \sum_{l=1}^{L_1} I_{\Omega_1}\left[\mathbf{x}^{(1,l)}\right] = \frac{T_1}{L_1}, \tag{12.12}$$

where P_1^{SUS} denotes the estimated value of P_1, and $I_{\Omega_1}(\mathbf{x})$ represents the indicator function

$$I_{\Omega_1}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_1 \\ 0 & \mathbf{x} \notin \Omega_1 \end{cases}. \tag{12.13}$$

If P_1 is large, it can be accurately estimated with a small number of random samples (e.g., L_1 is around $10^2 \sim 10^3$).

Next, in the 2nd phase, we need to estimate the conditional probability $P_2 = \Pr(\mathbf{x} \in \Omega_2 | \mathbf{x} \in \Omega_1)$. Towards this goal, one simple idea is to directly draw random samples from the conditional PDF $f(\mathbf{x}|\mathbf{x} \in \Omega_1)$ and then compute the mean of the indicator function $I_{\Omega_2}(\mathbf{x})$

$$I_{\Omega_2}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_2 \\ 0 & \mathbf{x} \notin \Omega_2 \end{cases}. \tag{12.14}$$

This approach, however, is practically infeasible since $f(\mathbf{x}|\mathbf{x} \in \Omega_1)$ is unknown in advance. To address this issue, we apply a modified Metropolis (MM) algorithm [1] to generate a set of random samples that follow the conditional PDF $f(\mathbf{x}|\mathbf{x} \in \Omega_1)$.

MM is a Markov chain Monte Carlo (MCMC) technique [3]. Starting from each of the samples $\{\mathbf{x}_F^{(1,t)}; t = 1, 2, \ldots, T_1\}$ that fall into Ω_1 in the 1st phase, MM generates a sequence of samples that form a Markov chain. In other words, there are T_1 independently generated Markov chains in total and $\mathbf{x}_F^{(1,t)}$ is the 1st sample of the t-th Markov chain. To clearly explain the MM algorithm, we define the symbol $\mathbf{x}^{(2,t,1)} = \mathbf{x}_F^{(1,t)}$, where $t \in \{1, 2, \ldots, T_1\}$. The superscripts "2" and "1" of $\mathbf{x}^{(2,t,1)}$ refer to the 2nd phase and the 1st sample of the Markov chain, respectively.

For our 2-D example, we start from $\mathbf{x}^{(2,1,1)} = [x_1^{(2,1,1)}\ x_2^{(2,1,1)}]^T$ to form the 1st Markov chain. To generate the 2nd sample $\mathbf{x}^{(2,1,2)}$ from $\mathbf{x}^{(2,1,1)}$, we first randomly sample a new value x_1^{NEW} from a 1-D transition PDF $q_1[x_1^{NEW}|x_1^{(2,1,1)}]$ that must satisfy the following condition [1]:

$$q_1\left[x_1^{NEW}\left|x_1^{(2,1,1)}\right.\right] = q_1\left[x_1^{(2,1,1)}\left|x_1^{NEW}\right.\right] \tag{12.15}$$

There are many possible ways to define $q_1[x_1^{NEW}|x_1^{(2,1,1)}]$ in (12.15) [1]. For example, a 1-D Normal PDF can be used

$$q_1\left[x_1^{NEW}\left|x_1^{(2,1,1)}\right.\right] = \frac{1}{\sqrt{2\pi}\cdot\sigma_1}\cdot\exp\left\{-\frac{\left[x_1^{NEW} - x_1^{(2,1,1)}\right]^2}{2\cdot\sigma_1^2}\right\}. \tag{12.16}$$

where $x_1^{(2,1,1)}$ and σ_1 are the mean and standard deviation of the distribution, respectively. Here, σ_1 is a parameter that usually be empirically chosen[19].

Next, we compute the ratio

$$r = \frac{p_1\left(x_1^{NEW}\right)}{p_1\left(x_1^{(2,1,1)}\right)}, \tag{12.17}$$

where $p_1(x_1)$ is the original PDF of the random variable x_1 shown in (12.2). A random sample u is then drawn from a 1-D uniform distribution with the following PDF:

$$f(u) = \begin{cases} 1 & 0 \le u \le 1 \\ 0 & \text{Otherwise} \end{cases}, \tag{12.18}$$

and the value of $x_1^{(2,1,2)}$ is set as

$$x_1^{(2,1,2)} = \begin{cases} x_1^{NEW} & u \le \min(1, r) \\ x_1^{(2,1,1)} & u > \min(1, r) \end{cases}. \tag{12.19}$$

A similar procedure is applied to generate $x_2^{(2,1,2)}$. Once $x_1^{(2,1,2)}$ and $x_2^{(2,1,2)}$ are determined, we form a candidate $\mathbf{x}^{NEW} = [x_1^{(2,1,2)}\ x_2^{(2,1,2)}]^T$ and use it to create the sample $\mathbf{x}^{(2,1,2)}$

$$\mathbf{x}^{(2,1,2)} = \begin{cases} \mathbf{x}^{NEW} & \mathbf{x}^{NEW} \in \Omega_1 \\ \mathbf{x}^{(2,1,1)} & \mathbf{x}^{NEW} \notin \Omega_1 \end{cases}. \tag{12.20}$$

By repeating the aforementioned steps, we can create other samples to complete the Markov chain $\{\mathbf{x}^{(2,1,l)}; l = 1, 2, \ldots, L_2\}$, where L_2 denotes the length of the Markov chain in the 2nd phase. In addition, all other Markov chains can be similarly formed. Since there are T_1 Markov chains and each Markov chain contains L_2 samples, the total number of the MCMC samples is $T_1 \cdot L_2$ for the 2nd phase. Figure 12.1(b) shows the sampling results in the 2nd phase for our 2-D example. In Fig. 12.1(b), the red points represent the initial samples $\{\mathbf{x}^{(2,t,1)}; t = 1, 2, \ldots, T_1\}$ of the Markov chains and they are obtained from the 1st phase. The yellow points represent the MCMC samples created via the MM algorithm in the 2nd phase. It has been proved in [1] that all these MCMC samples $\{\mathbf{x}^{(2,t,l)}; t = 1, 2, \ldots, T_1; l = 1, 2, \ldots, L_2\}$ in Fig. 12.1(b) approximately follow $f(\mathbf{x}|\mathbf{x} \in \Omega_1)$. In other words, we have successfully generated a number of random samples that follow our desired distribution for the 2nd phase.

Among all the MCMC samples $\{\mathbf{x}^{(2,t,l)}; t = 1, 2, \ldots, T_1; l = 1, 2, \ldots, L_2\}$, we further identify a subset of samples $\{\mathbf{x}_F^{(2,t)}; t = 1, 2, \ldots, T_2\}$ that fall into Ω_2, where T_2 denotes the total number of the samples in this subset. The conditional probability P_2 can be estimated as:

$$P_2^{\text{SUS}} = \frac{1}{T_1 \cdot L_2} \cdot \sum_{t=1}^{T_1} \sum_{l=1}^{L_2} I_{\Omega_2}\left[\mathbf{x}^{(2,t,l)}\right] = \frac{T_2}{T_1 \cdot L_2}, \tag{12.21}$$

where P_2^{SUS} denotes the estimated value of P_2.

By following the aforementioned idea, we can estimate all the probabilities $\{P_k; k = 1, 2, \ldots, K\}$. Once the values of $\{P_k; k = 1, 2, \ldots, K\}$ are estimated, the rare failure rate P_F is calculated by

$$P_F^{\mathrm{SUS}} = \prod_{k=1}^{K} P_k^{\mathrm{SUS}}, \tag{12.22}$$

where P_F^{SUS} represents the estimated value of P_F by using SUS. If we have more than two random variables, estimating the probabilities $\{P_k; k = 1, 2, \ldots, K\}$ can be pursued in a similar way [19].

To efficiently apply SUS, we must carefully choose the subset $\{F_k; k = 1, 2, \ldots, K\}$ so that the probability P_k will be close to 0.1, where $k \in \{1, 2, \ldots, K\}$. In this case, even if the failure rate P_F is extremely small (e.g., $10^{-8}\sim10^{-6}$), SUS only needs a small number of (e.g., $K = 6\sim8$) phases to complete. Furthermore, it only requires a few hundred samples during each phase to accurately estimate the probability P_k.

In addition, to quantitatively assess the accuracy of the proposed SUS estimator, we must estimate its confidence interval (CI). To this end, we need to know the distribution of P_F^{SUS}. Since P_F^{SUS} is equal to the multiplication of $\{P_k^{\mathrm{SUS}}; k = 1, 2, \ldots, K\}$, we must carefully study the statistical property of P_k^{SUS} in order to derive the distribution for P_F^{SUS}.

To be specific, P_1^{SUS} is calculated by using (12.12) with L_1 independent and identically distributed (i.i.d.) samples drawn from $f(\mathbf{x})$. Hence, according to the central limit theorem (CLT) [16], P_1^{SUS} approximately follows a Normal distribution

$$P_1^{\mathrm{SUS}} \sim N\left(P_1, v_1\right), \tag{12.23}$$

where the mean value P_1 is defined in (12.9) and the variance value v_1 can be approximated as [16]

$$v_1 \approx \frac{1}{L_1} \cdot P_1^{\mathrm{SUS}} \cdot \left(1 - P_1^{\mathrm{SUS}}\right). \tag{12.24}$$

On the other hand, the conditional probability P_k^{SUS}, where $k \in \{2, 3, \ldots, K\}$, can be estimated by using the MCMC samples $\{\mathbf{x}^{(k,t,l)}; t = 1, 2, \ldots, T_{k-1}; l = 1, 2, \ldots, L_k\}$ created by MM:

$$P_k^{\mathrm{SUS}} = \frac{1}{T_{k-1} \cdot L_k} \cdot \sum_{t=1}^{T_{k-1}} \sum_{l=1}^{L_k} I_{\Omega_k}\left[\mathbf{x}^{(k,t,l)}\right], \tag{12.25}$$

where $I_{\Omega_k}[\mathbf{x}]$ represents the indicator function

$$I_{\Omega_k}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_k \\ 0 & \mathbf{x} \notin \Omega_k \end{cases}. \tag{12.26}$$

Since the MCMC samples $\{\mathbf{x}^{(k,t,l)}; l = 1, 2, \ldots, L_k\}$, where $t \in \{1, 2, \ldots, T_{k-1}\}$, are strongly correlated, they cannot be considered as i.i.d. samples. For this reason, we cannot directly apply CLT to derive the distribution for the estimator P_k^{SUS} in (12.25).

To address this issue, we define a set of new random variables

$$s^{(k,t)} = \frac{1}{L_k} \cdot \sum_{l=1}^{L_k} I_{\Omega_k}\left[\mathbf{x}^{(k,t,l)}\right], \tag{12.27}$$

where $t \in \{1, 2, \ldots, T_{k-1}\}$. Studying (12.27) reveals two important observations. First, $s^{(k,t)}$ only depends on the t-th Markov chain $\{\mathbf{x}^{(k,t,l)}; l = 1, 2, \ldots, L_k\}$. Since different Markov chains are created from different initial samples $\{\mathbf{x}^{(k,t,1)}; t = 1, 2, \ldots, T_{k-1}\}$, the random variables $\{s^{(k,t)}; t = 1, 2, \ldots, T_{k-1}\}$ are almost statistically independent. Second, since all initial samples $\{\mathbf{x}^{(k,t,1)}; t = 1, 2, \ldots, T_{k-1}\}$ follow the same conditional PDF $p(\mathbf{x} \,|\, \mathbf{x} \in \Omega_{k-1})$ and all the Markov chains are generated by following the same procedure, the random variables $\{s^{(k,t)}; t = 1, 2, \ldots, T_{k-1}\}$ must be identically distributed. For these reasons, we can consider $\{s^{(k,t)}; t = 1, 2, \ldots, T_{k-1}\}$ as a set of i.i.d. random variables.

Based on (12.27), P_k^{SUS} in (12.25), where $k \in \{2, 3, \ldots, K\}$, can be re-written as

$$P_k^{\mathrm{SUS}} = \frac{1}{T_{k-1}} \cdot \sum_{t=1}^{T_{k-1}} s^{(k,t)} \tag{12.28}$$

and, as a result, approximately follows a Normal distribution according to CLT:

$$P_k^{\mathrm{SUS}} \sim N\left(P_k, v_k\right), \tag{12.29}$$

where P_k is defined in (12.10) and

$$v_k \approx \frac{1}{\left(T_{k-1} - 1\right) \cdot T_{k-1}} \cdot \sum_{t=1}^{T_{k-1}} \left[s^{(k,t)} - P_k^{\mathrm{SUS}}\right]^2. \tag{12.30}$$

To further derive the distribution for P_F^{SUS} in (12.22) based on (12.23) and (12.29), we take logarithm on both sides of (12.22) because it is much easier to handle summation than multiplication

$$\log\left(P_F^{\mathrm{SUS}}\right) = \sum_{k=1}^{K} \log\left(P_k^{\mathrm{SUS}}\right). \tag{12.31}$$

To derive the distribution of $\{log(P_k^{SUS}); k = 1, 2, \ldots, K\}$, we approximate the nonlinear function $\log(\bullet)$ by the first-order Taylor expansion around the mean value

P_k of the random variable P_k^{SUS}:

$$\log\left(P_k^{\mathrm{SUS}}\right) \approx \log\left(P_k\right) + \frac{P_k^{\mathrm{SUS}} - P_k}{P_k} \approx \log\left(P_k\right) + \frac{P_k^{\mathrm{SUS}} - P_k}{P_k^{\mathrm{SUS}}}. \tag{12.32}$$

According to the linear approximation in (12.32), $\log(P_k^{\mathrm{SUS}})$ follows a Normal distribution

$$\log\left(P_k^{\mathrm{SUS}}\right) \sim N\left[\log\left(P_k\right), v_{\log,k}\right], \tag{12.33}$$

where

$$v_{\log,k} = \frac{v_k}{\left(P_k^{\mathrm{SUS}}\right)^2}, \tag{12.34}$$

and $k \in \{1, 2, \ldots, K\}$.

Since $\log(P_F^{\mathrm{SUS}})$ is the summation of several "approximately" Normal random variables $\{\log(P_k^{\mathrm{SUS}}); k = 1, 2, \ldots, K\}$, $\log(P_F^{\mathrm{SUS}})$ also approximately follows a Normal distribution [16]

$$\log\left(P_F^{\mathrm{SUS}}\right) \sim N\left\{\mathrm{MEAN}\left[\log\left(P_F^{\mathrm{SUS}}\right)\right], \mathrm{VAR}\left[\log\left(P_F^{\mathrm{SUS}}\right)\right]\right\} \tag{12.35}$$

Based on (12.8), (12.31), and (12.33), MEAN[$\log(P_F^{\mathrm{SUS}})$] can be expressed as

$$\mathrm{MEAN}\left[\log\left(P_F^{\mathrm{SUS}}\right)\right] = \sum_{k=1}^{K} \log\left(P_k\right) = \log\left(\prod_{k=1}^{K} P_k\right) = \log\left(P_F\right), \tag{12.36}$$

and VAR[$\log(P_F^{\mathrm{SUS}})$] can be calculated as

$$\begin{aligned} \mathrm{VAR}\left[\log\left(P_F^{\mathrm{SUS}}\right)\right] &= \mathrm{VAR}\left[\sum_{k=1}^{K} \log\left(P_k^{\mathrm{SUS}}\right)\right] \\ &= \sum_{k=1}^{K} v_{\log,k} + 2 \cdot \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \mathrm{COV}\left[\log\left(P_i^{\mathrm{SUS}}\right), \log\left(P_j^{\mathrm{SUS}}\right)\right] \end{aligned}, \tag{12.37}$$

where COV(•, •) denotes the covariance of two random variables.

When applying MCMC, we often observe that an MCMC sample is strongly correlated to its adjacent sample. However, the correlation quickly decreases as the distance between two MCMC samples increases. Therefore, we can assume that the

samples used to estimate $\log(P_i^{\mathrm{SUS}})$ are weakly correlated to the samples used to estimate $\log(P_j^{\mathrm{SUS}})$, if the distance between i and j is greater than 1 (i.e., $|i-j|>1$). Based on this assumption, (12.37) can be approximated as

$$\mathrm{VAR}\left[\log\left(P_F^{\mathrm{SUS}}\right)\right] \approx \sum_{k=1}^{K} v_{\log,k} + 2\cdot\sum_{k=1}^{K-1}\mathrm{COV}\left[\log\left(P_k^{\mathrm{SUS}}\right),\log\left(P_{k+1}^{\mathrm{SUS}}\right)\right]. \tag{12.38}$$

Accurately estimating the covariance between $\log(P_k^{\mathrm{SUS}})$ and $\log(P_{k+1}^{\mathrm{SUS}})$ is nontrivial. Here, we derive an upper bound for $\mathrm{COV}[\log(P_k^{\mathrm{SUS}}),\log(P_{k+1}^{\mathrm{SUS}})]$ [16]:

$$\mathrm{COV}\left[\log\left(P_k^{\mathrm{SUS}}\right),\log\left(P_{k+1}^{\mathrm{SUS}}\right)\right] \le \sqrt{v_{\log,k}\cdot v_{\log,k+1}}, \tag{12.39}$$

where $k\in\{1,2,\ldots,K-1\}$. Substituting (12.39) into (12.38) yields

$$\mathrm{VAR}\left[\log\left(P_F^{\mathrm{SUS}}\right)\right] \le \sum_{k=1}^{K} v_{\log,k} + 2\cdot\sum_{k=1}^{K-1}\sqrt{v_{\log,k}\cdot v_{\log,k+1}} = v_{\log,\mathrm{SUS}}. \tag{12.40}$$

In this chapter, we approximate $\mathrm{VAR}[\log(P_F^{\mathrm{SUS}})]$ by its upper bound $v_{\log,\mathrm{SUS}}$ defined in (12.40) to provide a conservative estimation for the CI. Based on (12.36) and (12.40), (12.35) can be re-written as

$$\log\left(P_F^{\mathrm{SUS}}\right) \sim N\left[\log\left(P_F\right), v_{\log,\mathrm{SUS}}\right]. \tag{12.41}$$

According to (12.41), we can derive the CI for any given confidence level. For instance, the 95% CI is expressed as

$$\left[\exp\left(\log(P_F^{\mathrm{SUS}}) - 1.96\cdot\sqrt{v_{\log,\mathrm{SUS}}}\right), \exp\left(\log(P_F^{\mathrm{SUS}}) + 1.96\cdot\sqrt{v_{\log,\mathrm{SUS}}}\right)\right]. \tag{12.42}$$

To demonstrate the efficacy of SUS, we consider an SRAM column example designed in a 45 nm CMOS process, as shown in Fig. 12.2. In this example, our PoI is the read current I_{READ}, which is defined as the difference between the bit-line currents I_{BL} and $I_{\mathrm{BL\_}}$ (i.e., $I_{\mathrm{READ}} = I_{\mathrm{BL}} - I_{\mathrm{BL\_}}$) when we start to read CELL <0>. If I_{READ} is greater than a pre-defined specification, we consider the SRAM circuit as "PASS". For process variation modeling, the local V_{TH} mismatch of each transistor is considered as an independent Normal random variable. In total, we have 384 independent random variables (i.e., 64 bit-cells $\times$ 6 transistors per bit-cell = 384).

We first run CMC with 10^9 random samples, and the estimated failure rate is 1.1×10^{-6}, which is considered as the "golden" failure rate in this example. Next, we compare SUS with the traditional importance sampling technique: MNIS [17], where 2000 simulations are used to construct the distorted PDF. We repeatedly run

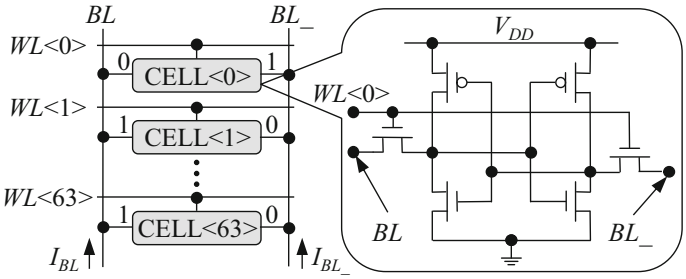

Fig. 12.2 The simplified schematic is shown for an SRAM column consisting of 64 bit-cells designed in a 45 nm CMOS process

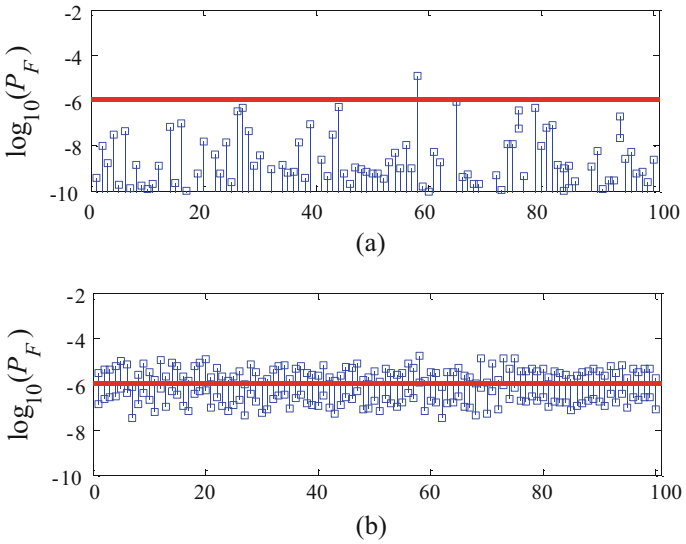

Fig. 12.3 The 95% confidence intervals (blue bars) of the SRAM read current example are estimated from 100 repeated runs with 6000 transistor-level simulations in each run for: (**a**) MNIS and (**b**) SUS. The red line represents the "golden" failure rate

MNIS and SUS for 100 times with 6000 transistor-level simulations in each run. Figure 12.3 shows the 100 estimated 95% CIs for each method, where each blue bar represents the CI of a single run, and the red line represents the "golden" failure rate.

In this example, only a single CI estimated from 100 repeated runs by MNIS can cover the "golden" failure rate, implying that MNIS fails to estimate the CIs accurately. This is an important limitation of MNIS, and generally most of the importance sampling techniques, since the user cannot reliably know the actual "confidence" of the estimator in practice. For the SUS approach, however, there are 95 CIs out of 100 runs that cover the "golden" failure rate. More importantly, the CIs

estimated by SUS are relatively tight, which implies that SUS achieves substantially better accuracy than the traditional MNIS approach in this example.

Before ending this section, we would like to emphasize that to define the subsets $\{F_k; k = 1, 2, \ldots, K\}$ required by SUS, PoI must be continuous. Realizing this limitation, we further describe a scaled-sigma sampling (SSS) approach to efficiently estimate the rare failure rates for discrete PoIs in a high-dimensional space, which will be presented in the next section.

12.3 Scaled-Sigma Sampling

Unlike the traditional importance sampling methods that must explicitly identify the high-probability failure region, SSS takes a completely different strategy to address the following questions: (1) how to efficiently draw random samples from the high-probability failure region, and (2) 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.

For the application of rare failure rate estimation, a failure event often occurs at the tail of the PDF $f(\mathbf{x})$. Given (12.2), it implies that the failure region Ω is far away from the origin $\mathbf{x} = \mathbf{0}$, as shown in Fig. 12.4(a). Since the failure rate is extremely small, the traditional CMC analysis cannot efficiently draw random samples from the failure region. Namely, many samples cannot reach the tail of $f(\mathbf{x})$.

To address the aforementioned sampling issue, SSS applies a simple idea. Given $f(\mathbf{x})$ in (12.2), we scale up the standard deviation of $\mathbf{x}$ by *a scaling factor* s ($s > 1$), yielding the following distribution:

$$g(\mathbf{x}) = \prod_{m=1}^{M} \left[\frac{\exp\left(-x_m^2 \Big/ 2s^2\right)}{\sqrt{2\pi}\, s} \right] = \frac{\exp\left(-\|\mathbf{x}\|_2^2 \Big/ 2s^2\right)}{\left(\sqrt{2\pi} \cdot s\right)^M}. \tag{12.43}$$

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 Fig. 12.4(b).

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 Fig. 12.4(a) and (b). Scaling the PDF from $f(\mathbf{x})$ to $g(\mathbf{x})$

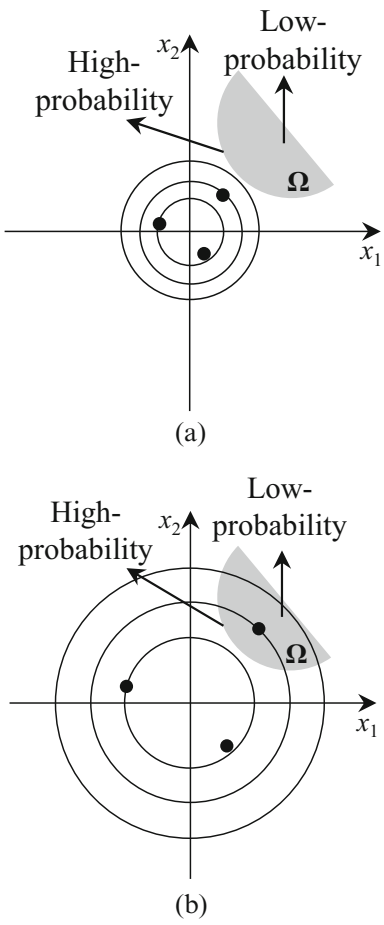

Fig. 12.4 The proposed SSS is illustrated by a 2-D example where the grey area Ω denotes the failure region and the circles represent the contour lines of the PDF. (**a**) Rare failure events occur at the tail of the original PDF $f(\mathbf{x})$ and the failure region is far away from the origin $\mathbf{x} = \mathbf{0}$. (**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

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 (12.43), we need to further estimate the failure rate P_F defined in (12.3). To this end, one straightforward way is to apply the importance sampling method [3]. Such a simple approach, however, has been proved to be intractable when the dimensionality (i.e., M) of the variation space is high [21]. Namely, it does not fit the need of high-dimensional failure rate estimation in this chapter.

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

$$P_G = \int_{\mathbf{x}\in\Omega} g(\mathbf{x}) \cdot d\mathbf{x} = \int_{-\infty}^{+\infty} I_\Omega(\mathbf{x}) \cdot g(\mathbf{x}) \cdot d\mathbf{x}, \tag{12.44}$$

where $I_\Omega(\mathbf{x})$ represents the indicator function:

$$I_\Omega(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega \\ 0 & \mathbf{x} \notin \Omega \end{cases}. \tag{12.45}$$

Our objective is to study the relation between the scaled failure rate P_G in (12.44) and the original failure rate P_F in (12.3). 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 (12.44) can be approximated as:

$$P_G \approx \sum_k I_\Omega \left[\mathbf{x}^{(k)}\right] \cdot g\left[\mathbf{x}^{(k)}\right] \cdot \Delta\mathbf{x}, \tag{12.46}$$

where $\Delta\mathbf{x}$ denotes the volume of a hyper-rectangle. The approximation in (12.46) is accurate, if each hyper-rectangle is sufficiently small. Given the definition of $I_\Omega(\mathbf{x})$ in (12.45), Eq. (12.46) can be re-written as:

$$P_G \approx \sum_{k\in\Omega} g\left[\mathbf{x}^{(k)}\right] \cdot \Delta\mathbf{x}, \tag{12.47}$$

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

Substituting (12.43) into (12.47), we have

$$P_G \approx \frac{\Delta\mathbf{x}}{\left(\sqrt{2\pi}\cdot s\right)^M} \cdot \sum_{k\in\Omega} \exp\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]. \tag{12.48}$$

Taking the logarithm on both sides of (12.48) yields:

$$\log P_G \approx \log \frac{\Delta\mathbf{x}}{(2\pi)^{M/2}} - M\cdot\log s + \operatorname*{lse}_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right], \tag{12.49}$$

where

$$\operatorname*{lse}_{k\in\Omega}\left[\frac{-\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right] = \log\left\{\sum_{k\in\Omega}\exp\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]\right\} \tag{12.50}$$

stands for the log-sum-exp function. The function lse(•) in (12.50) is also known as the "soft maximum" from the mathematics [4]. It can be bounded by

$$\max_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]+\log(T)\geq\operatorname*{lse}_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]\geq\max_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right], \tag{12.51}$$

where T denotes the total number of hyper-rectangles in Ω.

In general, there exist a number of (say, T_0) dominant hyper-rectangles that are much closer to the origin $\mathbf{0}$ than other hyper-rectangles in the set $\{\mathbf{x}^{(k)}; k \in \Omega\}$. Without loss of generality, we assume that the first T_0 hyper-rectangles $\{\mathbf{x}^{(k)}; k = 1, 2, \ldots, T_0\}$ are dominant. Hence, we can approximate the function lse(•) in (12.50) as

$$\operatorname*{lse}_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]\approx\log\left\{\sum_{k=1}^{T_0}\exp\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]\right\}. \tag{12.52}$$

We further assume that these dominant hyper-rectangles $\{\mathbf{x}^{(k)}; k = 1, 2, \ldots, T_0\}$ have similar distances to the origin $\mathbf{0}$. Thus, Eq. (12.52) can be approximated by

$$\operatorname*{lse}_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]\approx\max_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2s^2}\right]+\log(T_0). \tag{12.53}$$

Substituting (12.53) into (12.49) yields

$$\log P_G\approx\alpha+\beta\cdot\log s+\frac{\gamma}{s^2}, \tag{12.54}$$

where

$$\begin{aligned}\alpha&=\log\frac{\Delta\mathbf{x}}{(2\pi)^{M/2}}+\log(T_0)\\ \beta&=-M\\ \gamma&=\max_{k\in\Omega}\left[-\frac{\left\|\mathbf{x}^{(k)}\right\|_2^2}{2}\right]\end{aligned}. \tag{12.55}$$

Equation (12.54) reveals the important relation between the scaled failure rate P_G and the scaling factor s. The approximation in (12.54) does not rely on any specific assumption of the failure region. It is valid, even if the failure region is non-convex or discontinuous.

While (12.55) shows 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 fit the analytical model in (12.54) 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 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 (12.3) can be predicted by extrapolation. Namely, we substitute $s = 1$ into the analytical model in (12.54):

$$\log P_F^{\mathrm{SSS}} = \alpha + \gamma, \tag{12.56}$$

where P_F^{SSS} denotes the estimated value of P_F by SSS. Apply the exponential function to both sides of (12.56) and we have

$$P_F^{\mathrm{SSS}} = \exp\left(\alpha + \gamma\right). \tag{12.57}$$

To make the SSS method of practical utility, maximum likelihood estimation is applied to fit the model coefficients in (12.54). 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 $\{\mathbf{x}^{(n)}; n = 1, 2, \ldots, N_q\}$ are collected for the scaling factor s_q. The scaled failure rate $P_{G,q}$ can be estimated by MC

$$P_{G,q}^{\mathrm{MC}} = \frac{1}{N_q} \cdot \sum_{n=1}^{N_q} I_\Omega\left(\mathbf{x}^{(n)}\right), \tag{12.58}$$

where $I_\Omega(\mathbf{x})$ is the indicator function defined in (12.45). The variance of the estimator $P_{G,q}^{\mathrm{MC}}$ in (12.58) can be approximated as [16]

$$v_{G,q}^{\mathrm{MC}} = P_{G,q}^{\mathrm{MC}} \cdot \frac{1 - P_{G,p}^{\mathrm{MC}}}{N_q}. \tag{12.59}$$

If the number of samples N_q is sufficiently large, the estimator $P_{G,q}^{\mathrm{MC}}$ in (12.58) follows a Gaussian distribution according to CLT [16]

$$P_{G,q}^{\mathrm{MC}} \sim \mathrm{Gauss}\left(P_{G,q}, v_{G,p}^{\mathrm{MC}}\right), \tag{12.60}$$

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

Note that the model template in (12.54) is expressed for $\log P_G$, instead of P_G. To further derive the probability distribution for $\log P_{G,q}^{\mathrm{MC}}$, we adopt the first-order delta method from the statistics community [16]. Namely, we approximate the nonlinear function $\log(\bullet)$ by the first-order Taylor expansion around the mean value $\log P_{G,q}$ of the random variable $\log P_{G,q}^{\mathrm{MC}}$

$$\log P_{G,q}^{\mathrm{MC}} \approx \log P_{G,q} + \frac{P_{G,q}^{\mathrm{MC}} - P_{G,q}}{P_{G,q}} \approx \log P_{G,q} + \frac{P_{G,q}^{\mathrm{MC}} - P_{G,q}}{P_{G,q}^{\mathrm{MC}}}. \tag{12.61}$$

Based on the linear approximation in (12.61), $\log P_{G,q}^{\mathrm{MC}}$ follows the Gaussian distribution

$$\log P_{G,q}^{\mathrm{MC}} \sim \mathrm{Gauss}\left[\log P_{G,q},\ \frac{v_{G,q}^{\mathrm{MC}}}{\left(P_{G,q}^{\mathrm{MC}}\right)^2}\right]. \tag{12.62}$$

Equation (12.62) 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 $\{P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$ are mutually independent. Therefore, the Q-dimensional random variable

$$\log \mathbf{P}_G^{\mathrm{MC}} = \left[\log P_{G,1}^{\mathrm{MC}} \quad \log P_{G,2}^{\mathrm{MC}} \quad \cdots \quad \log P_{G,Q}^{\mathrm{MC}}\right]^T \tag{12.63}$$

satisfies the following jointly Gaussian distribution:

$$\log \mathbf{P}_G^{\mathrm{MC}} \sim \mathrm{Gauss}\left(\boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G\right), \tag{12.64}$$

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

$$\boldsymbol{\mu}_G = \left[\log P_{G,1} \quad \log P_{G,2} \quad \cdots \quad \log P_{G,Q}\right]^T \tag{12.65}$$

$$\boldsymbol{\Sigma}_G = \mathrm{diag}\left[\frac{v_{G,1}^{\mathrm{MC}}}{\left(P_{G,1}^{\mathrm{MC}}\right)^2}, \frac{v_{G,2}^{\mathrm{MC}}}{\left(P_{G,2}^{\mathrm{MC}}\right)^2}, \cdots, \frac{v_{G,Q}^{\mathrm{MC}}}{\left(P_{G,Q}^{\mathrm{MC}}\right)^2}\right], \tag{12.66}$$

where $\mathrm{diag}(\bullet)$ denotes a diagonal matrix.

The diagonal elements of the covariance matrix $\boldsymbol{\Sigma}_G$ in (12.66) can be substantially different. In other words, the accuracy of $\{\log P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$ 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 is difficult to accurately estimate $\log P_{G,q}$ from a small number of random samples. For this reason, instead of equally "trusting" the estimators $\{\log P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$, we must carefully model the "confidence" for each estimator $\log P_{G,q}^{\mathrm{MC}}$, as encoded by the covariance matrix $\mathbf{\Sigma}_G$ in (12.66). 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 (12.54), the mean vector $\boldsymbol{\mu}_G$ in (12.65) can be re-written as

$$\boldsymbol{\mu}_G = \alpha + \beta \cdot \begin{bmatrix} \log s_1 \\ \log s_2 \\ \vdots \\ \log s_Q \end{bmatrix} + \gamma \cdot \begin{bmatrix} s_1^{-2} \\ s_2^{-2} \\ \vdots \\ s_Q^{-2} \end{bmatrix} = \mathbf{A} \cdot \mathbf{\Theta}, \tag{12.67}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & \log s_1 & s_1^{-2} \\ 1 & \log s_2 & s_2^{-2} \\ \vdots & \vdots & \vdots \\ 1 & \log s_Q & s_Q^{-2} \end{bmatrix} \tag{12.68}$$

$$\mathbf{\Theta} = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T. \tag{12.69}$$

Equation (12.68) implies that the mean value of the Q-dimensional random variable $\log \mathbf{P}_G^{\mathrm{MC}}$ depends on the model coefficients α, β, and γ. Given $\{P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$, the key idea of MLE is to find the optimal values of α, β, and γ so that the likelihood of observing $\{P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$ is maximized.

Because the random variable $\log \mathbf{P}_G^{\mathrm{MC}}$ follows the jointly Gaussian distribution in (12.64), the likelihood associated with the estimated failure rates $\{P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$ is proportional to

$$L \sim \exp\left[-\frac{1}{2}\left(\log \mathbf{P}_G^{\mathrm{MC}} - \boldsymbol{\mu}_G\right)^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \left(\log \mathbf{P}_G^{\mathrm{MC}} - \boldsymbol{\mu}_G\right)\right]. \tag{12.70}$$

Taking the logarithm for (12.70) yields

$$\log L \sim -\left(\log \mathbf{P}_G^{\mathrm{MC}} - \boldsymbol{\mu}_G\right)^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \left(\log \mathbf{P}_G^{\mathrm{MC}} - \boldsymbol{\mu}_G\right). \tag{12.71}$$

Substitute (12.67) into (12.71), and we have

$$\log L \sim -\left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A} \cdot \mathbf{\Theta}\right)^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A} \cdot \mathbf{\Theta}\right). \tag{12.72}$$

Note that the log-likelihood $\log L$ in (12.72) depends on the model coefficients α, β, and γ, because the vector $\mathbf{\Theta}$ is composed of these coefficients as shown in (12.69). Therefore, the MLE solution of α, β, and γ can be determined by maximizing the log-likelihood function

$$\max_{\mathbf{\Theta}} \quad -\left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A}\cdot\mathbf{\Theta}\right)^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A}\cdot\mathbf{\Theta}\right). \tag{12.73}$$

Since the covariance matrix $\mathbf{\Sigma}_G$ is positive definite, the optimization in (12.73) is convex. In addition, since the log-likelihood $\log L$ is simply a quadratic function of $\mathbf{\Theta}$, the unconstrained optimization in (12.73) can be directly solved by inspecting the first-order optimality condition [4]

$$\begin{aligned} &\frac{\partial}{\partial \mathbf{\Theta}}\left[-\left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A}\cdot\mathbf{\Theta}\right)^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A}\cdot\mathbf{\Theta}\right)\right] \\ &\quad = 2\cdot \mathbf{A}^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \left(\log \mathbf{P}_G^{\mathrm{MC}} - \mathbf{A}\cdot\mathbf{\Theta}\right) = \mathbf{0} \end{aligned}. \tag{12.74}$$

Based on the linear equation in (12.74), the optimal value of $\mathbf{\Theta}$ can be determined by

$$\mathbf{\Theta} = \left(\mathbf{A}^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \mathbf{A}\right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{\Sigma}_G^{-1} \cdot \log \mathbf{P}_G^{\mathrm{MC}}. \tag{12.75}$$

Studying (12.75) reveals an important fact that the estimators $\{\log P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$ are weighted by the inverse of the covariance matrix $\mathbf{\Sigma}_G$. Namely, if the variance of the estimator $\log P_{G,q}^{\mathrm{MC}}$ is large, $\log P_{G,q}^{\mathrm{MC}}$ becomes non-critical when determining the optimal values of α, β, and γ. In other words, the MLE framework has optimally weighted the importance of $\{\log P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$ based on the "confidence" level of these estimators. Once α, β, and γ are solved by MLE, the original failure rate P_F can be estimated by (12.57).

To apply MLE, we need a set of pre-selected scaling factors $\{s_q; q = 1, 2, \ldots, Q\}$. In practice, appropriately choosing these scaling factors is a critical task due to several reasons. First, if these scaling factors are too large, the estimator P_F^{SSS} based on extrapolation in (12.57) would not be accurate, since the extrapolation point $s = 1$ is far away from the selected scaling factors. Second, if the scaling factors are too small, the scaled failure rates $\{P_{G,q}; q = 1, 2, \ldots, Q\}$ are extremely small and they cannot be accurately estimated from a small number of scaled random samples. Third, the failure rates for different performances and/or specifications can be quite different. To estimate them both accurately and efficiently, we should choose small scaling factors for the performance metrics with large failure rates, but large scaling factors for the performance metrics with small failure rates. Hence, finding an appropriate set of scaling factors for all performances and/or specifications can be extremely challenging.

In this chapter, a number of evenly distributed scaling factors covering a relatively large range are empirically selected. For the performance metrics with large failure rates, the scaled failure rates corresponding to a number of small scaling factors can be used to fit the model template in (12.54). On the other hand, the scaled failure rates corresponding to a number of large scaling factors can be used for the performance metrics with small failure rates. As such, a broad range of performances and/or specifications can be accurately analyzed by the SSS method.

While the MLE algorithm is able to optimally estimate the model coefficients α, β, and γ and then predict the failure rate P_F, it remains an open question how we can 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. Accurately estimating the confidence interval of P_F^{SSS} is not a trivial task.

To address the aforementioned challenge, a bootstrapping based technique [8] is developed to accurately estimate the CI of the SSS estimator. The key idea of bootstrap is to re-generate a large number of random samples based on a statistical model without running additional transistor-level simulations. These random samples are then used to repeatedly calculate the value of P_F^{SSS} in (12.57) 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 $\{P_{G,q}^{\mathrm{MC}}; q = 1, 2, \ldots, Q\}$. Each estimator $P_{G,q}^{\mathrm{MC}}$ follows the Gaussian distribution in (12.60). The actual mean $P_{G,q}$ in (12.60) is unknown; however, we can approximate its value by the estimated failure rate $P_{G,q}^{\mathrm{MC}}$. Once we know the statistical distribution of $P_{G,q}^{\mathrm{MC}}$, we can re-sample its distribution and generate N_{RS} sampled values $\{P_{G,q}^{MC(n)}; n = 1, 2, \ldots, N_{\mathrm{RS}}\}$. This re-sampling idea is applied to all scaling factors $\{s_q; q = 1, 2, \ldots, Q\}$, thereby resulting in a large data set $\{P_{G,q}^{MC(n)}; q = 1, 2, \ldots, Q; n = 1, 2, \ldots, N_{\mathrm{RS}}\}$. Next, we repeatedly run SSS for N_{RS} times and get N_{RS} different failure rates $\{P_F^{SSS(n)}; n = 1, 2, \ldots, N_{\mathrm{RS}}\}$. The confidence interval of P_F^{SSS} can then be estimated from the statistics of these failure rate values.

Note that to apply SSS, we only need a set of scaling factors and their corresponding scaled failure rates: $\{(s_q, P_{G,q}); q = 1, 2, \ldots, Q\}$. As long as $\{s_q; q = 1, 2, \ldots, Q\}$ are sufficiently large, $\{P_{G,q}; q = 1, 2, \ldots, Q\}$ are not small probability values and, therefore, can be efficiently estimated by CMC. When applying CMC, we only need to determine whether the random samples belong to the failure region. Namely, the PoI does not have to be continuous. Due to this reason, SSS can be applied to estimate the rare failure rates for both continuous and discrete PoIs. However, since SUS explores additional information from the continuous performance values, SUS is often preferred over SSS when we handle continuous PoIs.

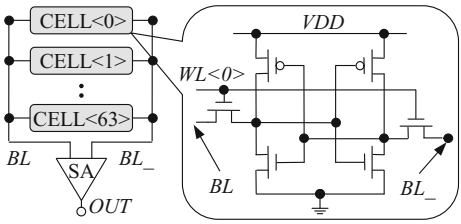

Fig. 12.5 The simplified schematic is shown for an SRAM column consisting of 64 bit-cells and a sense amplifier (SA) designed in a 45 nm CMOS process

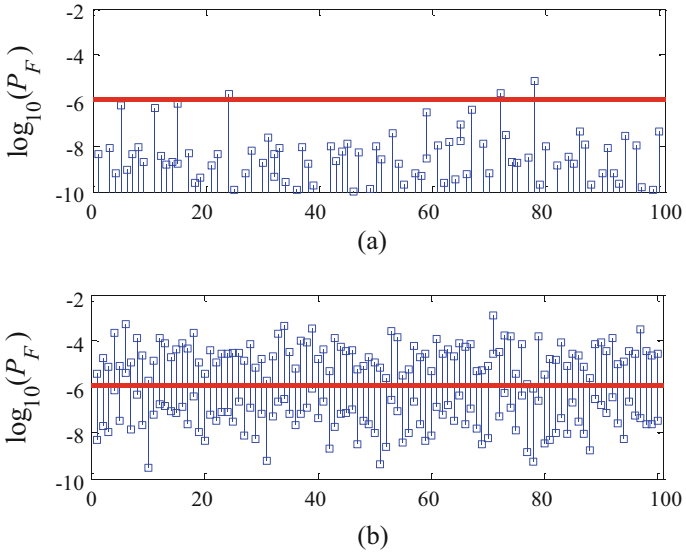

Fig. 12.6 The 95% confidence intervals (blue bars) of the SRAM example are estimated from 100 repeated runs with 6000 transistor-level simulations in each run for: (**a**) MNIS and (**b**) SSS. The red line represents the "golden" failure rate

To demonstrate the efficacy of SSS, we consider an SRAM column consisting of 64 bit-cells and a sense amplifier (SA) designed in a 45 nm CMOS process. Figure 12.5 shows the simplified circuit schematic of this SRAM column example. Similar to the SRAM read current example shown in Fig. 12.2, we consider the local V_{TH} mismatch of each transistor as an independent Normal random variable. In total, we have 384 independent random variables. In this example, the output of SA is considered as the PoI. If the output is correct, we consider the circuit as "PASS". Hence, the PoI is binary, and we cannot apply SUS in this example. For comparison purposes, we run MNIS [17] and SSS for 100 times with 6000 transistor-level simulations in each run. As shown in Fig. 12.6, there are 3 and 97 CIs out of 100 runs that cover the "golden" failure rate for MNIS and SSS, respectively. Here, the "golden" failure rate is estimated by CMC with 10^9 random samples. MNIS, again, fails to accurately estimate the corresponding CIs. SSS, however,

successfully estimates the CIs. These results demonstrate that SSS is superior to the traditional MNIS method in this SRAM example, where the dimensionality of the variation space is more than a few hundred.

12.4 Conclusions

Rare failure event analysis in a high-dimensional variation space has attracted more and more attention due to aggressive technology scaling. To address this technical challenge, we summarize two novel approaches: SUS and SSS. Several SRAM examples are used to demonstrate the efficacy of SUS and SSS. More experimental results of SUS and SSS can be found in the recent publications [19, 21]. Both SUS and SSS are based upon solid mathematical background and do not pose any specific assumption on the failure region. Hence, they can be generally applied to estimate the rare failure rates of a broad range of other circuits, e.g., DFF.

References

1. S. Au, J. Beck, Estimation of small failure probabilities in high dimensions by subset simulation. Probab. Eng. Mech. **16**(4), 263–277 (2001)
2. A. Bhavnagarwala, X. Tang, J. Meindl, The impact of intrinsic device fluctuations on CMOS SRAM cell stability. IEEE J. Solid-State Circuits **36**(4), 658–665 (2001)
3. C. Bishop, *Pattern Recognition and Machine Learning* (Prentice Hall, Upper Saddle River, 2007)
4. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2009)
5. B. Calhoun, Y. Cao, X. Li, K. Mai, L. Pileggi, R. Rutenbar, K. Shepard, Digital circuit design challenges and opportunities in the era of nanoscale CMOS. Proc. IEEE **96**(2), 343–365 (2008)
6. F. Cérou, P. Moral, T. Furon, A. Guyader, Sequential Monte Carlo for rare event estimation. Stat. Comput. **22**(3), 795–808 (2012)
7. L. Dolecek, M. Qazi, D. Shah, A. Chandrakasan, Breaking the simulation barrier: SRAM evaluation through norm minimization, in *International Conference on Computer-Aided Design* (2008), pp. 322–329
8. B. Efron, R. Tibshirnani, *An Introduction to the Bootstrap* (Chapman & Hall/CRC, London, 1993)
9. R. Fonseca, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel, N. Badereddine, A statistical simulation method for reliability analysis of SRAM core-cells, in *Design Automation Conference* (2010), pp. 853–856
10. C. Gu, J. Roychowdhury, An efficient, fully nonlinear, variability aware non-Monte-Carlo yield estimation procedure with applications to SRAM cells and ring oscillators, in *IEEE Asia and South Pacific Design Automation Conference* (2008), pp. 754–761
11. A. Guyader, N. Hengartner, E. Matzner-Løber, Simulation and estimation of extreme quantiles and extreme probabilities. Appl. Math. Optim. **64**(2), 171–196 (2011)
12. R. Heald, P. Wang, Variability in sub-100nm SRAM designs, in *International Conference on Computer-Aided Design* (2004), pp. 347–352

13. R. Kanj, R. Joshi, S. Nassif, Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events, in *Design Automation Conference* (2006), pp. 69–72
14. R. Kanj, R. Joshi, Z. Li, J. Hayes, S. Nassif, Yield estimation via multi-cones, in *Design Automation Conference* (2012), pp. 1107–1112
15. K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, T. Sato, Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis, in *International Conference on Computer-Aided Design* (2010), pp. 703–708
16. A. Papoulis, S. Pillai, *Probability, Random Variables and Stochastic Process* (McGraw-Hill, New York, 2001)
17. M. Qazi, M. Tikekar, L. Dolecek, D. Shah, A. Chandrakasan, Loop flattening and spherical sampling: highly efficient model reduction techniques for SRAM yield analysis, in *Design, Automation & Test in Europe* (2010), pp. 801–806
18. A. Singhee, R. Rutenbar, Statistical blockade: very fast statistical simulation and modeling of rare circuit events, and its application to memory design. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **28**(8), 1176–1189 (2009)
19. S. Sun, X. Li, Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space, in *International Conference on Computer-Aided Design* (2014), pp. 324–331
20. S. Sun, Y. Feng, C. Dong, X. Li, Efficient SRAM failure rate prediction via Gibbs sampling. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **31**(12), 1831–1844 (2012)
21. S. Sun, X. Li, H. Liu, K. Luo, B. Gu, Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space, in *International Conference on Computer-Aided Design* (2013), pp. 478–485
22. R. Topaloglu, Early, accurate and fast yield estimation through Monte Carlo-alternative probabilistic behavioral analog system simulations, in *IEEE VLSI Test Symposium* (2006), pp. 137–142
23. J. Wang, S. Yaldiz, X. Li, L. Pileggi, SRAM parametric failure analysis, in *Design Automation Conference* (2009), pp. 496–501