Graduate student
ACM Student Member Number: 6578924

Advisor: Prof. Xin Li
Member of ACM SIGDA: yes

ICCAD : G : Fast Statistical Analysis of Rare Circuit Failure Events in High-Dimensional Variation Space

Shupeng Sun, shupengs@ece.cmu.edu
Carnegie Mellon University, Pittsburgh, PA 15213 USA

I. PROBLEM AND MOTIVATION

As integrated circuit (IC) technology advances, the ever increasing process variation has become a growing concern [1]. A complex IC, containing numerous circuit components (e.g., millions of SRAM bit-cells integrated in an advanced microprocessor), 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 circuit component must be extremely small. For instance, the failure rate of an SRAM bit-cell must be less than 10^{-8}~10^{-6} for a typical SRAM design [2]-[3]. Due to this reason, efficiently analyzing the rare failure events for individual circuit components becomes an important task for the IC design community.

II. BACKGROUND AND RELATED WORK

The simple way to estimate the failure probability is to apply the well-known brute-force Monte Carlo (MC) technique. MC 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 random sample. Theoretically, $1/P_F$ random samples are required on average to obtain a failure sample [22]. Here, P_F denotes the failure rate. When MC is applied to estimate an extremely small failure rate (e.g., 10^{-8}~10^{-6}), a large number of (e.g., 10^7~10^9) random samples are needed to accurately estimate the small failure probability. Since generating each random sample requires a transistor-level simulation, 10^7~10^9 transistor-level simulations are needed to collect 10^7~10^9 random samples, implying that MC 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 [4], [7], [10], [12]-[13]. 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 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, called statistical blockade, has recently been proposed [9]. This approach first builds a classifier with a number of transistor-level simulations, and then draws random samples from the original PDF. Unlike MC 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 this method quickly intractable.

In addition to the aforementioned statistical methods, several deterministic approaches have also been proposed to efficiently estimate the rare failure probability [6], [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 the low-dimensional 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 [4]-[14] have been successfully applied to the SRAM cell-level design to estimate the rare failure rates for SRAM bit-cells where only a small number of (e.g., 6~20) 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 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 that

must be designed with extremely low failure rates. Taking D flip-flop (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 the rare failure events in a high-dimensional variation space with low computational cost.

III. APPROACH AND UNIQUENESS

A. Subset Simulation (SUS)

To accurately capture the rare failure events in a high-dimensional variation space, a novel subset simulation (SUS) technique is proposed in this work [21]. The key idea of SUS, borrowed from the statistics community [17]-[19], 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.

To intuitively understand the key idea of SUS, let us consider a simple 2-D example with two random variables $\mathbf{x} = [x_1\ x_2]$ to model the device-level process variations. In Figure 1, Ω denotes the failure region in the variation space, and is far away from the origin **0** for our application of rare failure rate estimation. The failure rate P_F can be written as $P_F = \Pr(x \in \Omega)$. Here, Pr(•) denotes the probability that an event occurs. Without loss of generality, we assume that $P_F = 10^{-6}$ in this simple example. Unlike MC that runs millions of transistor-level simulations to directly estimate the rare failure probability P_F, SUS first defines an intermediate failure region Ω_1 in the variation space, as shown in Figure 1 (b). Given Ω_1, the original rare failure probability P_F can be rewritten as $P_F = \Pr(x \in \Omega_1) \times \Pr(x \in \Omega \mid x \in \Omega_1)$. If the intermediate failure region Ω_1 is appropriately defined, the two probabilities $\Pr(x \in \Omega_1)$ and $\Pr(x \in \Omega \mid x \in \Omega_1)$ are much larger than P_F. For instance, $\Pr(x \in \Omega_1) = 10^{-3}$, and $\Pr(x \in \Omega \mid x \in \Omega_1) = 10^{-3}$. Hence, we can accurately estimate $\Pr(x \in \Omega_1)$ and $\Pr(x \in \Omega \mid x \in \Omega_1)$ with a small number of simulations. Restating in words, SUS by estimating $\Pr(x \in \Omega_1)$ and $\Pr(x \in \Omega \mid x \in \Omega_1)$ is much more efficient than MC by directly estimating P_F in this simple 2-D example.

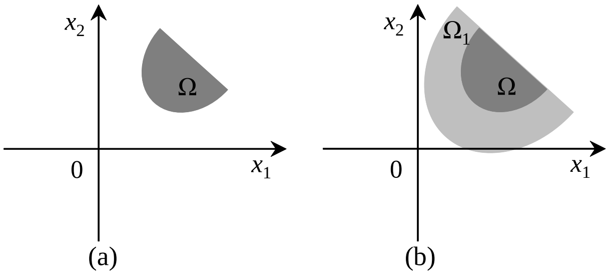


Figure 1. A two-dimensional example is used to illustrate the key idea of SUS: (a) the failure region Ω is far away from the origin and hard to capture; and (b) the intermediate failure region Ω_1 is close to the origin and easy to estimate.

When implementing the proposed SUS method, it is difficult, if not impossible, to directly draw random samples from the conditional PDFs (e.g., $\Pr(x \in \Omega \mid x \in \Omega_1)$ in the 2-D example shown in Figure 1) 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 [17] 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 [4]-[14] that suffer from the dimensionality issue, SUS can be efficiently applied to high-dimensional problems, which will be demonstrated by the experimental results in Section IV.

It is important to emphasize that though the key idea of SUS was first proposed in the statistics community [17]-[19], how to estimate the confidence interval of SUS remains an open question. Due to this reason, the SUS method has not been successfully applied to many practical applications in the literature. In this work, we propose a statistical methodology to accurately estimate the confidence interval of SUS based on the theory of Markov chain Monte Carlo simulation, thereby making our proposed SUS method of practical utility.

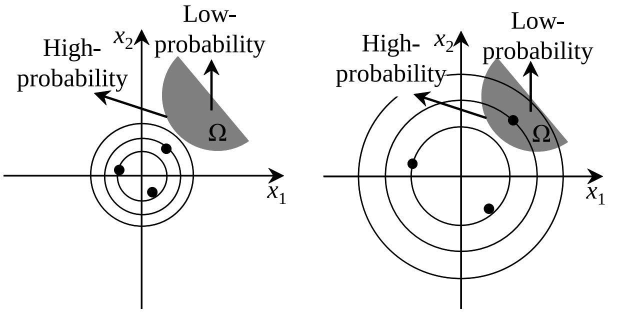
B. Scaled-Sigma Sampling (SSS)

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 binary (e.g., the output of a voltage-mode sense amplifier in SRAM circuit). Realizing this limitation, a novel scaled-sigma sampling (SSS) approach is proposed to efficiently estimate the rare failure rates for binary PoIs in a high-dimensional space [20]. 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 rare failure rate based on these random samples. Unlike MC 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. As a result, a large number of samples can now fall into the failure region.

To understand the key idea of SSS, let us consider a simple 2-D example with two random variables $\mathbf{x} = [x_1\ x_2]$ to model the device-level process variations. Without loss of generality, we further assume that x_1 and x_2 are mutually independent and follow standard Gaussian distribution [20]. In Figure 2, the grey area Ω denotes the failure region and the circles represent the contour lines of the PDF. For our application of rare failure event analysis, Ω is typically far away from the origin $\mathbf{0}$. Therefore, most samples drawn from the original PDF do not fall into the failure region Ω, as shown in Figure 2 (a). Once the standard deviation of $\mathbf{x}$ is scaled up, we conceptually increase the magnitude of process variations. Hence, the scaled PDF widely spreads over a large region and the probability for a random sample to reach the faraway failure region increases, as shown in Figure 2 (b). Studying Figure 2, we can observe that the "scaled" failure rate associated with the scaled PDF is larger than the original failure rate. If the scaling factor is sufficiently large, the "scaled" failure rate can be easily estimated from a small number of "scaled" random samples.

To further recover the original rare failure rate from the scaled failure rate, we derive an analytical model between the scaled failure rate and the scaling factor based on the theorem of "soft maximum" [23], which is the key contribution of this work. To fit the model, we first choose a set of scaling factors, and estimate their corresponding scaled failure rates from a small number of scaled random samples. The model is then optimally fitted by applying maximum likelihood estimation (MLE). Next, the original rare failure rate can be efficiently estimated from the fitted model by setting the scaling factor to 1.

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 SSS estimator is calculated by linear regression with nonlinear exponential/logarithmic transformation [20]. Therefore, accurately estimating the confidence interval of SSS is not a trivial task. To address the aforementioned challenge, we apply bootstrap technique [24]. 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 failure rate by SSS for multiple times. Based on these repeated runs, the statistics (hence, the confidence interval) of the proposed SSS estimator can be accurately estimated.



(a) (b)

Figure 2. The proposed SSS method 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 and the failure region is far away from the origin $\mathbf{x} = \mathbf{0}$. (b) The scaled PDF widely spreads over a large region and the scaled samples are likely to reach the faraway failure region.

C. Bayesian Scaled-Sigma Sampling (BSSS)

Though only a few thousand simulations are required by SUS and SSS for rare failure probability estimation [20]-[21], tens of thousands of simulations in total may be needed when designing a circuit component. To understand the reason, let us assume that we are designing a sense amplifier (SA), and applying SSS to estimate the failure probability. If the estimated failure probability meets our design specification, the design process is complete. Otherwise, we need to improve our SA design, and re-run SSS. Generally speaking, we need a few iterations before we converge to the final design. Since SSS spends a few thousand simulations for each design candidate, tens of thousands of simulations may be required over the entire design process, which can be extremely expensive.

To further reduce the simulation cost, we propose a novel Bayesian scaled-sigma sampling (BSSS) approach which can be considered as an extension of SSS. As mentioned in Section III.B, SSS is a model-based approach. By studying SSS, we can observe that a number of coefficients of the SSS model for an early design candidate are similar to those for a late design candidate. Motivated by this observation, we propose to explore the "similarity" between different SSS models fitted for different design candidates, and encode such "similarity" as our *prior* distribution for the SSS model coefficients. Next, we apply Bayesian model fusion (BMF) [25] to fit the SSS model with the *prior*.

The key difference between BSSS and SSS lies in the fact that BSSS maximizes the product of the prior distribution and the data likelihood by maximum-a-posteriori (MAP) estimation, while SSS only maximizes the data likelihood by MLE. As long as the prior distribution is properly defined, MAP can reduce the amount of required simulation data and, hence, the model fitting cost without surrendering any accuracy, as demonstrated in [25]. In other words, BSSS can be more efficient than SSS if the prior distribution is appropriately defined.

Similar to SSS, we apply bootstrap technique [24] to accurately estimate the confidence interval of our BSSS estimator. Our numerical experimental results in Section IV demonstrate that BSSS can achieve superior accuracy over SSS when the prior distribution is appropriately defined.

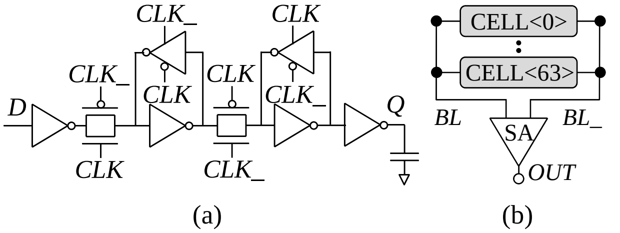
IV. RESULTS AND CONTRIBUTIONS

Example 1: D Flip-Flop (DFF)

Shown in Figure 3 (a) is a simplified schematic for a DFF circuit designed in a 45 nm CMOS process. 200 independent random variables are used to model the process variations. In this example, our PoI is the data transfer delay $D_{CLK\rightarrow Q}$ from the clock *CLK* to the output *Q*. If the delay $D_{CLK\rightarrow Q}$ belongs to a pre-defined interval [D_{LOW}, D_{UP}], we consider the DFF circuit as "PASS". Otherwise, the DFF circuit is considered as "FAIL". For comparison purposes, we run our proposed SUS method and the traditional minimum-norm importance sampling (MNIS) method [10] for 100 times where 5500 transistor-level simulations are performed in each run. Figure 4 shows the 100 95% confidence intervals (CIs) estimated by these two methods, where each blue bar represents the 95% CI of a single run and the red line represents the "golden" failure rate. In this example, there are only 17 CIs out of 100 runs that cover the "golden" failure rate for MNIS, implying that MNIS fails to estimate the CIs accurately. This is an important limitation of MNIS since the user cannot reliably know the actual "confidence" of the estimator in practice. For our proposed SUS approach, however, there are 98 CIs out of 100 runs that cover the "golden" failure rate. More importantly, the CIs estimated by SUS are relatively tight, which implies that our proposed SUS approach achieves substantially better accuracy than the traditional MNIS approach in this DFF example.

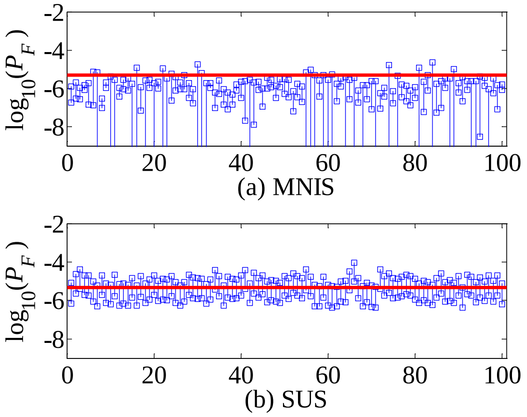
Example 2: SRAM Column

An SRAM column designed in a 45 nm CMOS process is shown in Figure 3 (b). It consists of 64 SRAM bit-cells and a sense amplifier (SA). 384 independent random variables are used to model the process variations. 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. We run SSS and MNIS for 100 times with 6000 transistor-level simulations in each run. As shown in Figure 5, there are 1 and 97 CIs out of 100 runs that cover the "golden" failure rate for MNIS and SSS respectively. MNIS, again, fails to accurately estimate the corresponding CIs. SSS, however, successfully estimates the CIs. These results demonstrate that our proposed SSS approach is superior to the traditional MNIS method in this SRAM example, where the dimensionality of the variation space is more than a few hundred.



(a) (b)

Figure 3. The simplified schematic is shown for: (a) a DFF circuit and (b) an SRAM column designed in a 45 nm CMOS process.



(a) MNIS

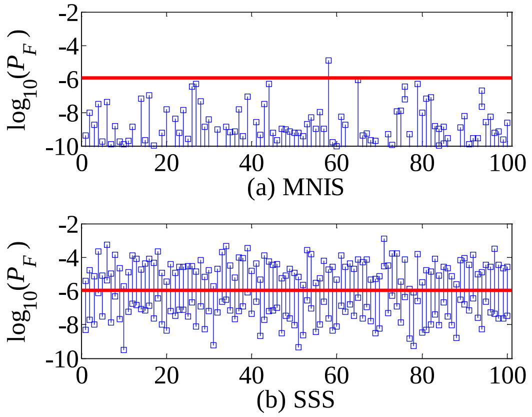
(b) SUS

Figure 4. The 95% confidence intervals (blue bars) of the DFF example are estimated from 100 repeated runs with 5500 transistor-level simulations in each run for: (a) MNIS and (b) SUS. The red line represents the "golden" failure rate.

Example 3: Sense Amplifier (SA)

In this sub-section, we consider another SA design that is deferent from the SA design used in Figure 3 (b). The SA considered here is also designed in a 45 nm CMOS process, and consists of 45 transistors. 536 independent random variables are used to model the process variations. The BL and BLB voltages are initially set to 1.1V and 1.2V respectively. If the output of SA is 0, the SA is considered as "PASS". Otherwise, it is considered as "FAIL".

We start from an initial SA design, and run SSS with 10^4 transistor-level simulations to estimate its failure rate. The estimated failure rate by SSS is 7.3×10^{-5}, which is relatively high and does not meet our specification. Hence, we need to further improve the SA design to reduce its failure rate. After tuning the transistor sizes, we obtain a second design. We run three different approaches (i.e., MNIS, SSS and BSSS) to estimate the failure rate of the second design. The estimated 95% confidence intervals by



(a) MNIS

(b) SSS

Figure 5. 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.

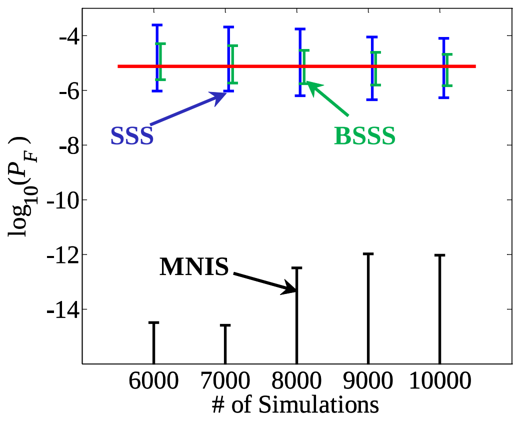


Figure 6. The 95% confidence intervals estimated by three different approaches with different numbers of simulations for the SA example: MNIS (black bars), SSS (blue bars) and BSSS (green bars). The red line represents the "golden" failure rate.

these three approaches with different numbers of simulations are shown in Figure 6. From Figure 6, we can observe that BSSS achieves significantly enhanced accuracy over MNIS and SSS especially when the number of simulations is small. This SA example demonstrates that BSSS is much preferable to SSS in the practical circuit design flow where a few iterations are required before we converge to the final design.

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 propose three novel approaches: SUS, SSS and BSSS. Three examples are used to demonstrate the efficacy of our proposed approaches. More experimental results can be found in our recent publications [20]-[21].

SUS, SSS and BSSS are all based upon solid mathematical background and do not pose any specific assumption on the failure region. Hence, they can be generally applied to many circuit examples. In the CAD community, SUS and SSS have been well recognized by both academia and industry. Our paper on SUS has been selected as one of the best paper candidates for ICCAD 2014. On the other hand, several major semiconductor and EDA companies are currently evaluating SSS for commercial applications. For instance, Cadence is now integrating SSS into their commercial EDA tool Virtuoso ADE GXL, as mentioned by the recent press release [26].

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, no. 4, 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] R. Topaloglu, "Early, accurate and fast yield estimation through Monte Carlo-alternative probabilistic behavioral analog system simulations," *IEEE VTS*, pp. 137-142, 2006.

[6] 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.

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

[14] R. Kanj, R. Joshi, Z. Li, J. Hayes and S. Nassif, "Yield estimation via multi-cones," *IEEE DAC*, pp. 1107-1112, 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. Au and J. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Eng. Mechanics*, vol. 16, no. 4, pp. 263-277, Oct. 2001.

[18] A. Guyader, N. Hengartner and E. Matzner-Løber, "Simulation and estimation of extreme quantiles and extreme probabilities," *Appl. Math. Optimization*, vol. 64, no. 2, pp. 171-196, Oct. 2011.

[19] F. Cérou, P. Moral, T. Furon and A. Guyader, "Sequential Monte Carlo for rare event estimation," *Stat. Computing*, vol. 22, no. 3, pp. 795-808, May 2012.

[20] S. Sun, X. Li, H. Liu, K. Luo and B. Gu, "Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space," *IEEE ICCAD*, pp. 478-485, 2013.

[21] S. Sun and X. Li, "Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space," *IEEE ICCAD*, pp. 324-331, 2014.

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

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

[24] B. Efron and R. Tibshirnani, *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

[25] F. Wang, W. Zhang, S. Sun, X. Li and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE DAC*, 2013.

[26] S. Lewis (2014, July 8), *EDA Plus Academia: A Perfect Game, Set and Match* [Online]. Available: http://community.cadence.com/cadence_blogs_8/b/cic/archive/2014/07/08/eda-plus-academia-a-perfect-game-set-and-match