Breaking the Simulation Barrier: SRAM Evaluation Through Norm Minimization

Lara Dolecek, Masood Qazi, Devavrat Shah, Anantha Chandrakasan
Department of Electrical Engineering and Computer Science,
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
email: {dolecek, mqazi,devavrat}@mit.edu, anantha@mtl.mit.edu

Abstract— **With process variation becoming a growing concern in deep submicron technologies, the ability to efficiently obtain an accurate estimate of failure probability of SRAM components is becoming a central issue. In this paper we present a general methodology for a fast and accurate evaluation of the failure probability of memory designs. The proposed statistical method, which we call *importance sampling through norm minimization principle*, reduces the variance of the estimator to produce quick estimates. It builds upon the importance sampling, while using a novel norm minimization principle inspired by the classical theory of Large Deviations. Our method can be applied for a wide class of problems, and our illustrative examples are the data retention voltage and the read/write failure tradeoff for 6T SRAM in 32 nm technology. The method yields computational savings on the order of 10000x over the standard Monte Carlo approach in the context of failure probability estimation for SRAM considered in this paper.**

I. Introduction

The semiconductor chip market is worth well over $200 billion [1], [8]. The significant majority of these chips have a high demand for embedded memory to enable evermore computationally intensive applications. Take as a representative example a microprocessor chip for personal computers [19]: roughly half of its area is dedicated to on-chip memory, comprised of 48 million static random access memory (SRAM) cells. For each metric of cost, performance, and minimum standby power of such multi-megabit SRAMs, the worst case cell sets the performance. In this paper we adress this multi-billion dollar question associated with guaranteeing acceptable failure rates of SRAM cells.

In the deep sub-micron technology, the random variations introduced by semiconductor processes poses a new set of challenges for an efficient cell sizing and design [2]. Therefore, a priori it is hard to meet the requirement of *one-in-million* cell failure rate while being *economically viable* – poor prediction can lead to severe mis-estimation of yield. In the context of SRAM applications, typical interest is in maximizing the density of the cells, and thus minimizing their size, while maintaining an acceptable level of the failure probability [7]. However, as previously observed [10], the variations due to dopant fluctuations increase with the decrease in the gate area. This situation has created the need for very accurate estimation of failure probability where neither the worst case analysis suffices, nor are the analytic models accurate enough [9], [10], [15].

A *de facto* way of evaluating the performance under the statistical variability, or under inadequate analytical description, is to use extensive Monte Carlo simulations. While Monte Carlo approach is general, simple to implement, and can produce accurate estimates, its drawback is in the number of simulation runs one needs to execute to achieve accurate estimate, since the number of simulations increases with the decrease in the probability of the event of interest. Due to high replication of individual cells in the memory block, when evaluating the feasibility of a memory design, the major concern is how to achieve a very low probability of failure on the per cell basis. There is thus a pressing need for an efficient statistical method to evaluate *very low* probabilities.

Previous work. Previous work on using statistical methods for SRAM performance evaluation includes [12], [6], and [15]. In particular, our work is motivated by the results presented in [12], where an importance sampling-based approach is developed, and the work in [6] where a related approach is used as a part of the evaluation of contending memory architectures. We improve on this method in terms of mathematical formalization of the approach, resulting accuracy and computational savings.

Related work. A related line of work was presented in [15], and recently extended in [16]. The approach in [15] uses a different statistical tool, known as the Extreme Value Theory (EVT), to produce the estimates of the probability of very rare failures. The examples in [15] consider the write time problem where it is useful to obtain a continuum of values. We view IS and EVT as complementary approaches, in the sense that one may be better suited for a specific problem, –e.g. an IS-based method could be used when pass/failure differentiation is clear, such as in the case of the static noise margin (SNM), and an EVT-based method could be used when the tradeoff between a physically meaningful output value (e.g. delay) and the probability of failure is of interest over a continuum of values – whereas these methods can jointly provide a comprehensive evaluation of circuit design and performance using statistical ideas.

Our contributions and results. In this work, we focus on developing a complete and general importance sampling based method, which may even be viewed as “black-box” that does not need special tweaking or ad-hoc argument adjustment for a specific example for it to work. Our method is an importance sampling approach based on a novel *norm minimization principle*. This principle is derived through structural insights from the classical theory of large deviations. We provide a sufficient theoretical background and the intuition behind the method, from an added insight into this approach.

This method can be used for a quick and accurate evaluation of memory designs that successfully addresses the need for multiple sizings evaluated in an iterative fashion. It provides computational speed-ups on the order of up to 10000x while maintaining the same level of accuracy as the traditional Monte Carlo approach. We illustrate the method via representative examples: the data retention problem, and the read and write stability trade-offs.

Specifically, in the experimental section we present the analysis of a cell failing at a rate of 10^{-5} to 10^{-6}: the corresponding Monte Carlo run length of 40 million took two months to run, while our method produced the same estimate of the probability of failure in under *two hours*. By providing a quicker way to assess the failure probability, not only can a single cell sizing be qualified sooner, but successively more refined choices of device sizes can be tested in the allocated time frame. The modification of device sizings allows the SRAM cell designer to obtain lower failure rate at the cost of larger area. At the same time, the SRAM cell designer must iterate several times to ensure that area is not inefficiently wasted for too low of a failure probability.

Another important application of statistical methods in high performance circuit designs is in evaluation of circuit delays. Typical integrated circuits contain 10^2 to 10^3 timing paths that could potentially limit performance. An early work [11] in statistical timing analysis showed the significance of considering delay distributions in analyzing and optimizing performance of digital circuits. Subsequent work in the statistical analysis of timing includes [3], [14], [17]. We are optimistic that the methodology developed here can be also suitably applied to the statistical timing analysis.

Organization. In Section II we provide the necessary background on Monte Carlo and importance sampling-based approaches. Section III contains the detailed description of the proposed algorithm. The generality and the applicability of the algorithm is demonstrated through examples provided in Section IV which simultaneously highlight both the accuracy of the resulting predictions as well as the significant computational speed-ups obtained by our approach. Section V summarizes the main contributions of this paper and proposes future extensions.

II. Background: Monte Carlo and Importance Sampling

Monte Carlo. Monte Carlo (MC) is a popular statistical method to estimate probability of certain event A under an unknown distribution. Usually, such an event is characterized by a random variable X of interest (whose distribution is unknown) taking value in a certain range. For example, in Section IV we will be interested in finding the probability of failure of an SRAM element, an event characterized by the Static Noise Margin (SNM) being less than 0. Formally, the goal is to find p where

$$p = \Pr(X \in A).$$

The MC method generates N independent samples of X, say $X_1, \dots, X_N$ and forms an estimate of p, denoted by $\hat{p}_{\text{MC}}$ as the fraction of samples showing event A. For the SRAM example, it will be the fraction of samples for which the failure occurred, i.e. SNM < 0. Formally,

$$\hat{p}_{\text{MC}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(X_i \in A),$$

where the truth-indicator $\mathbf{1}(ST)$ takes value 1 if *ST* is true, and 0 if *ST* is false. The MC generates unbiased and asymptotically correct estimate of p (e.g. see [18]). That is,

$$\mathbb{E}[\hat{p}_{\text{MC}}] = p, \quad \text{for any } N, \qquad \text{and} \quad \hat{p}_{\text{MC}} \to p, \quad \text{as } N \to \infty\ .$$

The quantity of interest is the number of samples required to obtain an estimate of certain accuracy with desired confidence. Specifically, for MC it is known that to obtain an estimate with $(1-\varepsilon)100\%$ accuracy with $(1-\delta)100\%$ confidence the number of samples required, $N(\varepsilon, \delta)$ scales as

$$N(\varepsilon, \delta) \approx \frac{\log(1/\delta)}{p\varepsilon^2}.$$

Thus, for 90% accuracy ($\varepsilon = 0.1$) and 90% confidence ($\delta = 0.1$) we need roughly $100/p$ samples. Therefore, the MC estimator requires too many samples when the event is *rare* (i.e. p very small) – which is indeed the case for the probability of failure of an SRAM element.

Importance Sampling. The MC takes too many samples to estimate small p because it requires $1/p$ samples on average just to observe the event A (or failure) even once ! Consider the following scenario: we have a random variable $\hat{X}$ with distribution such that $\hat{p} = \Pr(\hat{X} \in A)$ is not *rare* (say, $\hat{p}$ is close to 0.5). Though we do not know distribution of X and the new variable $\hat{X}$, we do know relation between them in the following sense: let f be the density of X and $\hat{f}$ be the density of $\hat{X}$, where we do know w, and where

$$w(x) = \frac{f(x)}{\hat{f}(x)}, \quad \text{for all} \quad x.$$

In such a scenario, we can indeed *speed-up* the MC method by first finding $\hat{p}$ quickly using samples of $\hat{X}$ and then using the knowledge of function w appropriately. This is the key idea behind *Importance Sampling* (IS).

Formally, under importance sampling we obtain N independent samples of $\hat{X}$, say $\hat{X}_1, \dots, \hat{X}_N$. Then, the estimator of p, denoted by $\hat{p}_{\text{IS}}$ is

$$\hat{p}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N} w(\hat{X}_i)\mathbf{1}(\hat{X}_i \in A).$$

It can be easily verified that $\hat{p}_{\text{IS}}$ is unbiased and asymptotically correct estimator of p, as in

$$\mathbb{E}[\hat{p}_{\text{IS}}] = p, \quad \text{for any } N, \qquad \text{and} \quad \hat{p}_{\text{IS}} \to p, \quad \text{as } N \to \infty\ .$$

However, the IS estimator is useful only if the number of samples required for the given accuracy and confidence is less than that required by the MC estimator. Of course, MC is a special case of IS (i.e. choose $\hat{X} = X$). However, the challenge lies in systematic design of $\hat{X}$ so that the function $w(\cdot)$ is well-known, and the computational speedup is significant. Addressing this challenge requires problem dependent creative solution. In this paper, we shall put-forth a novel method, based on a new *norm minimization principle*, to address this challenge for a class of problems. We establish the effectiveness of our solution in the context of evaluating the probability of failure of SRAM elements in Section IV. Before presenting the algorithm in Section III, we establish some useful definitions.

Figure of merit. An important question concerned with the simulation setup is the following: given the requirement of $(1-\varepsilon)100\%$ accuracy with $(1-\delta)100\%$ confidence, when should we stop the simulation? For this, we use the notion of *figure of merit*. Specifically, given an estimator $\hat{p}$, let its variance at the end of an N-sample simulation run be $VAR(\hat{p})$. For MC, it is

$$VAR(\hat{p}_{MC}) = \frac{1}{N}\left(\hat{p}_{MC} - \hat{p}_{MC}^2\right) \ . \tag{1}$$

For IS, it is given by

$$VAR(\hat{p}_{\text{IS}}) = \frac{1}{N^2}\left(\sum_{i=1}^{N} w(\hat{X}_i)^2 \mathbf{1}(\hat{X}_i \in A) - N\hat{p}_{\text{IS}}^2\right). \tag{2}$$

Given the variance VAR$(\hat{p})$, figure of merit $\rho(\hat{p})$ is defined as

$$\rho(\hat{p}) = \frac{\sqrt{VAR(\hat{p})}}{\hat{p}} \ . \tag{3}$$

This figure of merit can be used as follows: suppose we stop when the figure of merit, $\rho(\hat{p}) \leq \varepsilon\sqrt{\log(1/\delta)}$. Then, we can declare that the estimate of p is $(1-\varepsilon)100\%$ accurate with confidence at least $(1-\delta)100\%$. For example, if we stop our algorithm when $\rho(\hat{p}) = 0.1$, then we have both the accuracy and confidence of 90% each.

A useful example of $\hat{X}$. Here, we present an example of $\hat{X}$ when we can evaluate function w even though we do not have a handle on distribution of X or $\hat{X}$. This example will be used in the algorithm described in Section III, as well as its applications in Section IV. Suppose X is some (possibly, non-linear and complicated) function of a collection of independent Gaussian variables. Specifically, let $X = F(Y_1, \ldots, Y_M)$ where each Y_i is an independent Gaussian random variable with mean μ_i and variance σ_i^2, i.e. $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, and X takes real values. In Section IV, X will be SNM, $M = 6$ and $Y_1, \ldots, Y_6$ will be certain input threshold voltages. Due to the lack of knowledge of F, we do not have information about distribution of X. To obtain $\hat{X}$, we will *shift* the input mean of $Y_1, \ldots, Y_M$. Specifically, we consider $\hat{Y}_1, \ldots, \hat{Y}_M$ where $\hat{Y}_i$ is a Gaussian random variable with mean $\mu_i + s_i$ and the same variance σ_i^2. Now, $\hat{X} = F(\hat{Y}_1, \ldots, \hat{Y}_M)$. Again, we do not know the distribution of $\hat{X}$. Fortunately, we know the function w since we are going to sample Ys or $\hat{Y}$s. Specifically, w is defined (with a little abuse of notation) as

$$\begin{aligned} w(\hat{y}_1, \ldots, \hat{y}_M) &= \frac{f(\hat{y}_1, \ldots, \hat{y}_M)}{\hat{f}(\hat{y}_1, \ldots, \hat{y}_M)} \\ &= \frac{\exp\left(-\sum_{i=1}^{M} \frac{(\hat{y}_i - \mu_i)^2}{2\sigma_i^2}\right)}{\exp\left(-\sum_{i=1}^{M} \frac{(\hat{y}_i - \mu_i - s_i)^2}{2\sigma_i^2}\right)} \\ &= \exp\left(-\sum_{i=1}^{M} \frac{s_i(2(\hat{y}_i - \mu_i) - s_i)}{2\sigma_i^2}\right) \end{aligned} \tag{4}$$

In this case, the IS estimator for $p = \Pr(X \in A)$ is as follows: obtain N independent samples of the M-vector, $\hat{\mathbf{Y}}^1, \ldots, \hat{\mathbf{Y}}^N$ where $\hat{\mathbf{Y}}^k = (\hat{Y}_1^k, \ldots, \hat{Y}_M^k)$ with $\hat{Y}_i^k \sim \mathcal{N}(\mu_i + s_i, \sigma_i^2)$ for $1 \leq k \leq N$. Then, the estimator of p is

$$\hat{p}_{\text{IS}}(\mathbf{s}) = \frac{1}{N}\sum_{k=1}^{N} w(\hat{\mathbf{Y}}^k)\mathbf{1}(F(\hat{\mathbf{Y}}^k) \in A), \tag{5}$$

with value of $w(\hat{\mathbf{Y}}^k)$ as in (4). Here we note the dependence of an estimator on the shift-vector $\mathbf{s} = (s_1, \ldots, s_M)$ by denoting it as $\hat{p}_{\text{IS}}(\mathbf{s})$.

III. Importance Sampling through Norm minimization

Here, we present our novel algorithm for importance sampling based on a "norm minimization" principle. We consider the setup as described in the example above. We wish to find $p = \Pr(X \in A)$ for some *rare* event A with $X = F(Y_1, \ldots, Y_M)$ of unknown distribution (or function F), with $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. We will use importance sampling estimator $\hat{p}_{\text{IS}}(\mathbf{s})$ with certain shift-vector $\mathbf{s}$. The key question is: what should be the value of $\mathbf{s}$, say $\mathbf{s}^*$, to achieve good *speed-up*?

In what follows, we present an algorithm to find such $\mathbf{s}^*$. The algorithm is based on the insight from the classical theory of *Large Deviations* [4]: When a *rare* event happens, it happens in the most likely manner; therefore the probability of a rare event can be estimated by that of this most likely aspect of it. In the case of our setup, this insight translates into choosing $\mathbf{s}$ that minimizes a certain norm (see discussion later in this Section). We present the algorithm below, followed by cost of algorithm and a brief discussion about its theoretical justification.

IS through Norm Minimization Algorithm. The algorithm has two main steps: (1) find a good shift-vector $\mathbf{s}^*$, and (2) run IS based on $\mathbf{s}^*$. We describe them separately as follows.

Step 1: Find $\mathbf{s}^*$. This step includes two sub-steps: (a) *filtering* for *reasonable* shifts to obtain a collection $\mathcal{F}$; and (b) obtaining the *minimal norm* shift in $\mathcal{F}$.

(a) By a reasonable shift, $\mathbf{s}$, we mean that the probability of occurrence of the event A is approximately 0.5 under the associated values. In order to look for all such reasonable $\mathbf{s}$, we will do the following. Sample $\mathbf{s}$ uniformly from a region R, where

$$R = \{(s_1, \ldots, s_M) : -L \leq s_i - \mu_i \leq L, \ \text{ for all } 1 \leq i \leq M\},$$

for some large L[1]. For a given sampled $\mathbf{s}$, include it in $\mathcal{F}$ if we find that probability of A is ≈ 0.5[2]. The number of samples to calibrate $\mathcal{F}$ can be chosen to be some large fixed number, or, more generally, it can be adaptively improved as discussed under *cost of algorithm.*

(b) Once $\mathcal{F}$ is chosen as per (a), choose $\mathbf{s}^*$ as the following minimizer of the L_2 norm:

$$\mathbf{s}^* = \arg\min \sum_{i=1}^{M} \frac{s_i^2}{\sigma_i^2} \qquad \text{over} \quad \mathbf{s} \in \mathcal{F}. \tag{6}$$

Step 2: IS using $\mathbf{s}^*$. Given the $\mathbf{s}^*$, run IS to obtain the estimate $\hat{p}_{\text{IS}}(\mathbf{s}^*)$ as explained in (5) with function w given in (4). Run the algorithm for N large enough so that the figure of merit $\rho(\hat{p}_{\text{IS}}(\mathbf{s}^*))$, evaluated as per (3) is less than 0.1 – here (2) need to be used. This method produces an estimate with 90% accuracy of 90% confidence. More generally, run the algorithm until $\rho(\hat{p}_{\text{IS}}(\mathbf{s}^*))$ becomes less than $\varepsilon\sqrt{\log(1/\delta)}$ for $(1-\varepsilon)100\%$ accuracy with $(1-\delta)100\%$ confidence.

Cost of Algorithm. Here we discuss the cost of the algorithm in terms of the number of samples required. The algorithm, as described above, includes sampling in both steps. In Step 1, the sampling is required to calibrate $\mathcal{F}$. In Step 2, the sampling is required to obtain $\hat{p}_{\text{IS}}(\mathbf{s}^*)$ with small figure of merit. We would like the cost of Step 1 to be no more than of Step 2. To achieve this goal, our approach is to calibrate $\mathcal{F}$ adaptively. Specifically, we start with some large number, say N_0 of samples (of shifts) to calibrate $\mathcal{F}$. Then based on this $\mathcal{F}$, we obtain the best $\mathbf{s}^*$ and run Step 2. Either our algorithm stops within N_0 samples (of $\mathbf{s}^*$ shifted r.v.), as we reach small enough figure of merit, or not. In the latter case, we go back to Step 1, sample N_0 more shifts, obtain possibly large $\mathcal{F}$, find new $\mathbf{s}^*$ and then repeat Step 2 for N_0 more steps; and repeat the above. As experiments in Section IV suggest, this approach is very successful for evaluating the failure probability of a memory element. An important feature of our algorithm is its general applicability, and the excellent speed up in simulations.

Discussion. The classical Large Deviation Theory [4] deals with estimation of probabilities of rare events. Since these estimates are distribution dependent, they are not of much use in our context. However, the theory does provide the following important structural insight: *when a rare event happens, it happens in the most likely manner*. Next, we explain how this leads to the "norm minimization principle" used in Step 1(b) of the algorithm.

Our interest is in the rare event A. Conditioned on the event A happening, as per the above insight, X must be taking values around a most likely $x^* \in A$. Now, recall that X is a function of vector $\mathbf{Y} = (Y_1, \ldots, Y_M)$: $X = F(\mathbf{Y})$. Let $\mathcal{Y}(x^*)$ be the set of all $\mathbf{y} = (y_1, \ldots, y_M)$ so that $F(\mathbf{y}) = x^*$. Since all $Y_1, \ldots, Y_M$ are independent Gaussian random variables, the most probable $\mathbf{Y}$ value in $\mathcal{Y}(x^*)$ is the following minimizer of the L_2 norm:

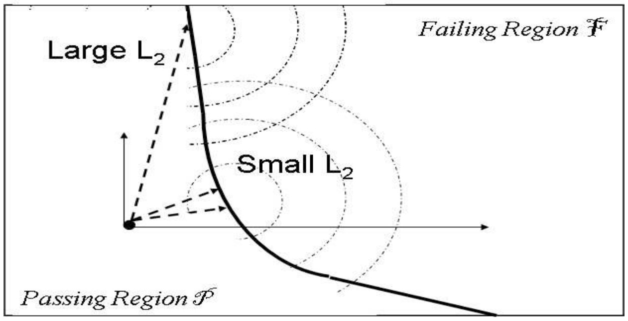
Large L_2
Failing Region $\mathcal{F}$
Small L_2
Passing Region $\bar{\mathcal{F}}$

Fig. 1. Illustration of the minimum norm principle. The circular contours represent the equiprobable elements under Gaussian distribution. The closest point (with respect to the mean-vector) on boundary of $\mathcal{F}$ represents the most likely way for the rare event A to happen.

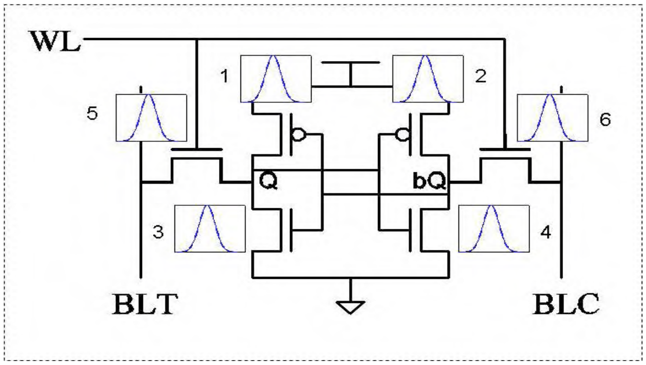

Fig. 2. The schematic of the 6T memory element along with the illustration of the variation of the individual threshold voltages. The BSIM VTH0 parameters in the simulation model are $-450mV$ and $509mV$ for the PMOS and NMOS devices, respectively. Each variation is modelled as an independent Gaussian random variable of zero mean and standard deviation of $\sigma = 36mV$ for PMOS devices, and $\sigma = 50mV$ for NMOS devices.

$$\arg\min \sum_i \frac{(y_i - \mu_i)^2}{\sigma_i^2} \quad \text{over} \quad \mathbf{y} = (y_1, \ldots, y_M) \in \mathcal{Y}(x^*).$$

If we call $s_i = y_i - \mu_i$, then the above minimization is the same as the (6) in Step 1(b) with the difference of $\mathcal{F}$ in place of $\mathcal{Y}(x^*)$. Even though the exact identification of $\mathcal{Y}(x^*)$ is computationally intractable, our Step 1(a) suggests a computationally feasible approximation to it. A pictorial description of this discussion is presented in Figure 1.

IV. Experimental Results

In this section we provide illustrative examples of our method. While the method itself works independent of the concrete application, or the exact details of the underlying parameters, we focus on the relevant examples of the SRAM cell analysis, since these elements are known to suffer from process variation in deeply scaled semiconductor technologies. This work employs 32nm bulk CMOS predictive technology

[1] The choice of L can be made judiciously depending on the magnitude of the probability of the rare event.

[2] This point can be easily checked with the help of few, say 100 trials, per choice of $\mathbf{s}$.

V_{DD} (mV)	MC est.	MC σ equiv.	MC no. of runs	IS est.	IS σ equiv.	IS no. of runs	speed-up	σ equiv. rel. error ($\times 100\%$)
275	5.4×10^{-3}	2.5491	2×10^{4}	4.9×10^{-3}	2.5828	10^{3}	$20x$	1.3
300	3.65×10^{-4}	3.3781	1.2×10^{6}	4.4×10^{-4}	3.3263	3×10^{3}	$100x$	1.53
400	3.1×10^{-6}	4.5195	4×10^{7}	3.0×10^{-6}	4.5264	10^{4}	$10000x$	0.15

TABLE I

COMPARISON OF THE MONTE CARLO - BASED ESTIMATOR AND THE PROPOSED IMPORTANCE SAMPLER. NOTE THE EXTREMELY CLOSE AGREEMENT IN THE ESTIMATED PROBABILITY OF FAILURE, WHILE THE NUMBER OF TRIALS IS REDUCED 10000 TIMES.

models [21]. The results elucidate the important features and benefits of the proposed method.

A. Example 1: data retention

Our first example considers the data retention voltage (DRV) for a six transistor memory element, shown in Figure 2. It is widely used in 6T and 8T embedded SRAM memories in microprocessors and other systems on a chip. Specifically we ask how low can the supply voltage, V_{DD}, be set such that the original data state on the two nodes Q and bQ does not flip. In an ideal process the DRV is arbitrarily low; however, the variation of the electrical behavior of each of the six transistors introduces an asymmetry that will favor one data state over another.

IS results. Our main results are summarized in Table I. Note the exceptionally close agreement between the estimates produced by the IS and MC methods, in particular the relative difference in terms of equivalent σ (defined as the deviation of the standard Gaussian random variable) is within 1.53%. As the table indicates, the computational savings also become more pronounced as the probability of failure gets smaller, which is a (known) feature of a well-designed variance reduction estimator. For consistency, we report the results at $\rho = 0.1$ throughout, which as previously argued guarantees 90% confidence with 90% accuracy.

For this problem, there are $M = 6$ input parameters, as indicated in Figure 2. Each is modelled as an independent Gaussian random variable, with the means and variances as indicated in the same figure. The rare event of interest is the failure $\{SNM \leq 0\}$, i.e. the collapse of the SNM curve (cf. Figures 4 for an illustration). By symmetry and geometrical construction of the butterfly curve it is sufficient to consider the collapse of a single lobe in both MC and IS. Using the algorithm outlined in the previous section we obtain the following:

Step 1(a) The boundary is determined by separating regions where $SNM \leq 0$ (fail) and where $SNM > 0$ (pass), and is obtained by uniformly sampling an appropriate space of V_T shifts. As an example, for $V_{DD} = 400mV$, a ten thousand point uniform random sampling of V_T shifts over $[0, 4\sigma_{V_T}]$ or $[-4\sigma_{V_T}, 0]$ (appropriately chosen for each device to degrade the corresponding butterfly lobe) produces roughly 3300 candidate shifts[3]. The resulting set of failing 6-dimensional vectors are sorted in the ascending order of the resulting SNM magnitude.

Step 1(b) Out of the subset of vectors yielding low SNM magnitude select *any* mean shift vector $\underline{m}$ that yields a low L_2. Some representative examples of such candidate vectors are given in Table II for the $V_{DD} = 400mV$ case.

A closer look at numbers. Details of the evolution of the simulation runs under $V_{DD} = 400mV$ are given in Figure 3, with the importance sampler run using the mean shifts given by the rows of Table II. Observe an extremely close agreement between the final predictions, whilst with the proposed approach the predictions settle around the final value 10000 times sooner than the Monte Carlo based approach. Recall that the algorithm can pick either one of the candidate shifts, and run the rest of the importance sampling on; as such it is robust to the precise details of the chosen vector, and is equipped with a flexibility to choose among several equally successful choices. Also note the suppressed variability of the IS plot relative to the MC plot. Note also that at the 10^3 and even 10^4 trials, a Monte Carlo run erroneously reports no errors – there is simply not enough gathered statistics to produce the estimate.

As a useful visual tool, Figure 4 illustrates the butterfly curves corresponding to the nominal values of threshold voltages (top graph) and corresponding to the values offset by the amount listed in the first row of Table II (bottom graph). Note that how in the latter case, the butterfly curve is on the verge of collapsing. The exact same butterfly curves, i.e. same as the bottom one in Figure 4, are obtained for the remaining rows of Table II. As expected, it is the joint effect of individual shifts that matters the most; as such can be achieved using different combinations of individual shifts. An interesting observation regarding the entries in Table II can be made:

Although all shifts exhibit the correct skew between symmetric device pairs—namely, pfet 1 is strengthened as pfet 2 is weakened, nfet 3 is weakened as nfet 4 is strengthened, and nfet 5 is strengthened as nfet 6 is weakened—the degree of skews between these individual pairs differs significantly across the three choices. At the same time, the norm stays similar across the three shifts because a smaller skew in one pair is balanced by a larger skew in the other two pairs.

Furthermore, when one considers the data state being upset (in this case $Q = 0V$ undesirably flipping to $Q = 400mV$), circuit operation insight highlights three key ratioed strengths of device pairs: 1 versus 3, 2 versus 4, and 5 versus 3. Table II shows that the amount of shifts in magnitude given to the

[3] While a 10 000 point sampling is an overkill at this stage, it also results in several, i.e. more than one, valid candidate shifts to be used in Step 2, and as we will see, these are all equally successful choices.

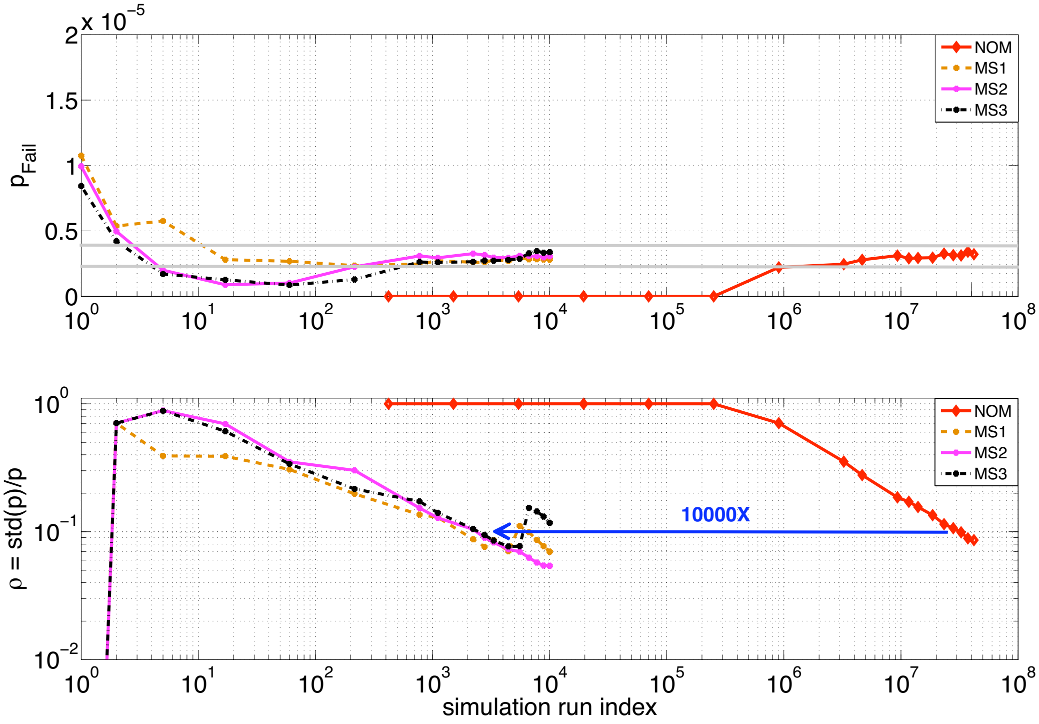

Fig. 3. Top: Zoomed-in plot of the evolutions of the estimates for the two approaches. Note how the IS curves very quickly settle around the convergent value. Bottom: Plot of ρ for the two approaches. Note the almost parallel nature of the curves.

	VT_1 shift	VT_2 shift	VT_3 shift	VT_4 shift	VT_5 shift	VT_6 shift	SNM value	L_2 norm
MS1	0.5091	-2.3923	1.7185	-3.5374	-0.7017	0.9689	-0.000265	4.7833
MS2	1.4651	-2.0932	0.8809	-3.8175	-0.7908	0.7313	-0.000023	4.7997
MS3	1.1703	-2.9488	0.02344	-3.5605	-0.7888	0.04651	-0.000567	4.8340

TABLE II

SEVERAL CANDIDATE MEAN-SHIFT VECTORS PRODUCED BY OUR METHOD FOR EXAMPLE 1. EACH ENTRY IS NORMALIZED TO BE A MULTIPLE OF SIGMA OF A STANDARD NORMAL VARIABLE. EACH ROW CORRESPONDS TO THE LOW SNM VALUE AND LOW L_2 NORM WHILE THE *individual* MEAN SHIFTS ON THE PER TRANSISTOR BASIS VARY SUBSTANTIALLY FROM ONE CHOICE TO ANOTHER.

individual members in a ratioed pair are not identical and, therefore, naïvely applying a common and large mean shift to all devices will not capture the typically rare events associated with Table II.

In contrast to applying the importance sampling estimator using the set of shifts based on the L_2 minimization, if one obliviously chooses the set of shifts based on the low SNM value alone (and thus possibly a larger L_2 norm), the computational speed-up and the benefit of using the importance sampling algorithm will be potentially diminishing. We have in fact experimented with various choices, and as the Large Deviations Theory suggests (and practice confirms), larger L_2 norms cannot accurately capture the failing region under the same number of trials, see also Figure 1.

Reporting p_{fail} vs. σ. While we have plotted the absolute p_{fail} estimates over time to contrast the time-wise evolution of the two approaches in Figure 3, the σ-equivalent results are reported in Table I. Note the remarkably close agreement between the two predictions. We also use this opportunity to emphasize the importance of quoting the absolute p_{fail} estimates, along with the conventional σ-style reporting [12], [15].

As a concrete example, consider a memory block consisting of N cells, and with the ability to overcome up to R errors, due to the additional layer of error-correcting capability. The quantity of interest is the overall yield, expressed as

$$Y = \sum_{k=0}^{R} \left(\begin{array}{c} N \\ k \end{array} \right) p^R (1-p)^{N-R} \ . \qquad (7)$$

Consider numerical values $N = 5 \times 10^5$ and $R = 3$. Suppose the Monte Carlo estimate of the per-cell probability of failure is 3.1×10^{-6}, as in the example considered here. In terms of the equivalent σ characterization, it corresponds to the value of 4.51. The yield Y is then 92%. Note that the importance

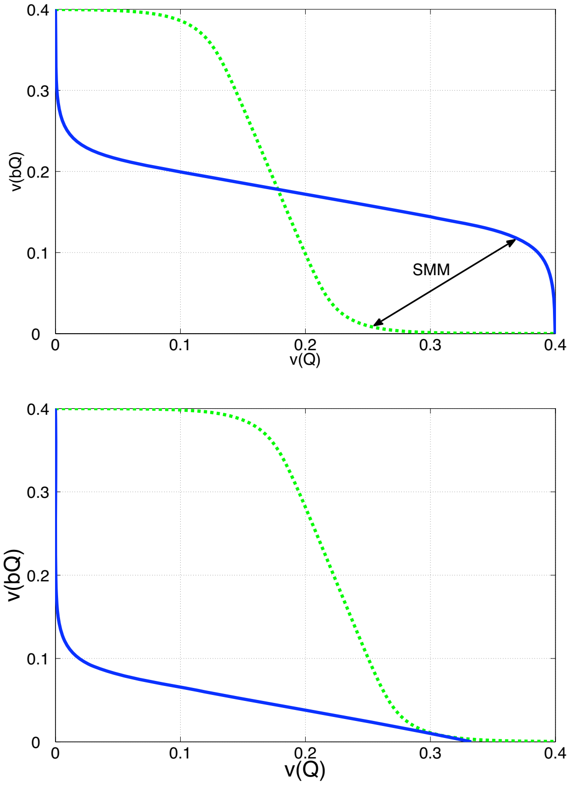

Fig. 4. Butterfly plots for the data retention condition at $V_{DD} = 400mV$. Top: Nominal V_T values exhibit two well separated and stable solutions at $Q = 0V$ and $Q = 400mV$. Bottom: Mean-shifted V_T values as obtained by the algorithm, produce a butterfly on the verge of collapsing to only one stable solution at . Note how under the latter values the butterfly curve is on the verge of collapsing to $Q = 0$.

sampling based prediction of p_{fail} has essentially the same yield estimate at 93%, and the equivalent σ is 4.52. If however one were to report the performance of a proposed estimator in terms of multiples of σ only, the estimate of the overall yield could be highly inaccurate.

For example, underestimating σ by only 5% at the failure rate of 3.1×10^{-6} results in the erroneous yield prediction of *only* 38%, which is very far from the 93% value of the actual yield. Such gross mis-estimates of the yield can severely impact the design cycles and the overall production cost.

We also point out that our predictions outperform [12] under the σ-style reporting; compare Table 5 in [12] and Table I above. The former lists 0.7σ discrepancy at MC σ equiv. = 4.15, whereas in our simulations the difference between the two is only 0.0069 at MC σ equiv. = 4.15, while the total number of runs needed to reach such an estimate is lower, and is moreover governed by the objective figure of merit.

Finally, our numerical result of $p_{fail} = 3 \times 10^{-6}$ at $V_{DD} = 400mV$ is a tolerable failure rate for many practical applications, and, therefore, predicts the DRV for 32nm technology to be around $400mV$ for high density bit cells.

B. Example 2: Read and write trade-off

While our previous example illustrated the success of the method where a single mode of failure is of interest – e.g. the data retention problem – in this example we demonstrate how the method can be equally successfully applied where multiple (and not necessarily mutually exclusive) modes of failure can occur. This is of particular interest in the design trade-off between the read and the write processes; our experiment addresses 4 different modes of failure: (I) read of 0, (II) read of 1, (III) write of 0, and (IV) write of 1.

Step 1(a). A thousand point uniform random sampling of V_T shifts over $[-4\sigma_{V_T}, 4\sigma_{V_T}]$ is run to simultaneously establish the boundaries separating pass/fail regions for each of the events (I) –(IV). *Step 2*(b). The four mean shifts, corresponding to the events (I) – (IV) are obtained based on the minimization of the individual L_2 norms of the points on the boundaries, and are listed in Table III. The time-wise evolution of IS and MC are plotted in Figure 5, highlighting the 150x speed-up of our method.

V. Conclusion

Process variability has become a growing issue for scaled technologies. Using the worst-case analysis is no longer suitable, and the analytical evaluations have become difficult. A way to estimate the performance is by using the standard Monte Carlo techniques. While the Monte Carlo approach is general enough in that it does not assume a particular structure of the functional description of the event of interest, its major drawback is that it is a very time- and resource-consuming exercise for estimating low probabilities of failure.

A present challenge is to design an efficient algorithm, which like Monte Carlo does not rely on the analytical description, while it also provides computational savings over the standard Monte Carlo approach.

In this paper we presented a highly efficient and general method for a quick evaluation of SRAM designs. The proposed methodology utilizes importance sampling as a suitable statistical tool for a quick evaluation of the circuit performance, equipped with a novel, large-deviations inspired, norm minimization approach. The computational advantage of the proposed methodology is illustrated through accompanying examples, which demonstrate up to the 10000X speed-up with no performance loss relative to the standard Monte Carlo based approaches, and significantly better accuracy, and the computational speed-up over the previous method [12].

While this work focused on very accurate evaluation of the functionality of a large memory block, the added benefit of the computational speed-ups obtained by the proposed stochastic method is that it can also enable exploratory study and design of SRAMs [6], and improve upon recently proposed Monte-Carlo based designs [5], [20]. An interesting and a fruitful future research direction is in developing statistical methods, along the lines of the large deviations approach presented here, for addressing the problem of accurate timing analysis. We envision the method presented here to become a key ingredient of a statistical tool for highly efficient circuit design and evaluation we will develop in the future.

Acknowledgement

The authors acknowledge the support of the DARPA ITMANET PROGRAM under grant 1105741-1-TFIND and the C2S2 program.

	VT_1 shift	VT_2 shift	VT_3 shift	VT_4 shift	VT_5 shift	VT_6 shift	SNM value	L_2 norm
Read 1	-0.7752	1.0799	-2.1426	2.5867	-0.9971	-1.9935	-0.00337	4.2447
Write 1	1.2629	1.5273	-1.2539	0.1306	1.743	3.0466	-0.00555	4.2234
Write 0	1.0105	-1.0915	-0.09699	-1.5308	2.7524	2.7007	-0.00518	4.4085
Read 0	1.1873	-1.9851	3.0305	-0.73	-1.8197	-0.4385	-0.00146	4.3094

TABLE III

MEAN-SHIFT VECTORS PRODUCED BY OUR METHOD FOR EXAMPLE 2 FOR VARIOUS MODES OF FAILURE UNDER $V_{DD} = 625mV$. WHILE THE FAILURE RATE IS IMPRACTICALLY HIGH (FOR REASONABLY SIZED MEMORIES), IT DOES ILLUSTRATE THE CONCEPT OF BALANCING FAILURE MODES.

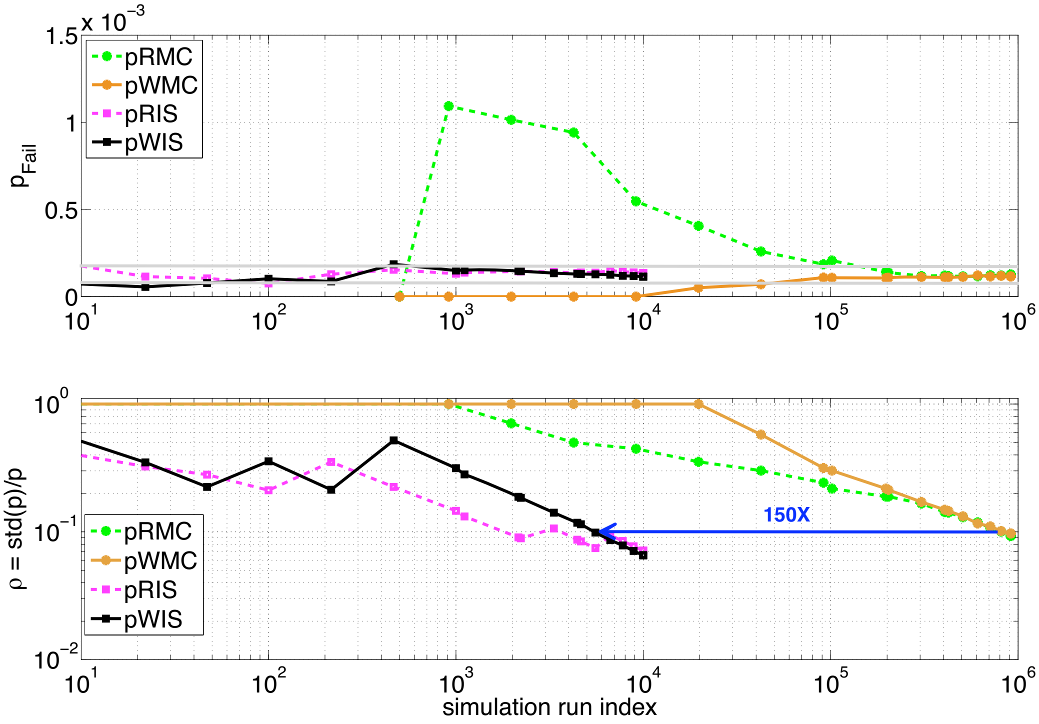

Fig. 5. Time evolution of IS and MC for Experiment 2. These results illustrate an efficient design where the probability of different mode failures are balanced.

REFERENCES

[1] Available at http://www.iSuppli.com

[2] K. Agarwal and S. Nassif, "Statistical analysis of SRAM cell stability," in *DAC*, 2006, pp. 57 –62.

[3] S. Bhardwaj, S. Vrudhula, and D. Blaauw, "τAU: timing analysis under uncertainty," in *ICCAD*, 2003, pp. 615-620.

[4] J. A. Bucklew, *Large Deviation Techniques in Decision, Simulation, and Estimation*, Wiley Series in Prob. and Math. Stat., 1990.

[5] B. H. Calhoun and A. P. Chandrakasan, "Static noise margin variation for sub-threshold SRAM in 65-nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 41, no. 7, July 2006, pp. 1673 – 79.

[6] G. K. Chen, D. Blaauw, T. Mudge, D. Sylvester, and N. S. Kim, "Yield-driven near-threshold SRAM design," in *ICCAD*, 2007, pp. 660 – 666.

[7] F. Duan, R. Castagnetti, R. Venkatraman, O. Kobozeva, and S. Ramesh, "Design and use of memory-specific test structures to ensure SRAM yield and manufacturability," in *ISQED*, 2003, pp. 119 – 203.

[8] S. A. Edwards, *The Nanotech Pioneers: Where Are They Taking Us*, Wiley, 2006.

[9] D. J. Frank, Y. Taur, M. Ieong and H. P. Wong, "Monte Carlo modeling of threshold variation due to dopant fluctuations," in *Symposium on VLSI Technology*, 1999, pp. 169 – 170.

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

[11] H.-F. Jyu, S. Malik, S. Devadas, and K. Keutzer, "Statistical timing analysis of combinatorial logic circuits," *IEEE Trans. on VLSI Systems*, vol. 1, no. 2, June 1993, pp. 126 – 137.

[12] R. Kanj, R. Joshi and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," in *DAC*, 2006, pp. 69 – 72.

[13] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Statistical design and optimization of SRAM cell for yield enhancement," in *ICCAD*, 2004, pp. 10-13.

[14] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," in *DAC*, 2002, pp. 556 – 561.

[15] A. Singhee and R. Rutenbar, "Statistical blockade: a novel method for very fast Monte Carlo simulation of rare circuit events, and its application," in *DATE* 2007, pp. 1–6.

[16] A. Singhee, J. Wang, B. H. Calhoun and R. Rutenbar "Recursive statistical blockade: an enhanced technique for rare event simulation with application to SRAM circuit design," in *VLSI Design Conference*, 2008, pp. 131 – 136.

[17] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*, Springer 2005.

[18] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*, Springer, 2002.

[19] G. Varghese et al. "Penryn: 45-nm next generation Intel®core^{TM} 2 processor," in *ASSCC*, 2007, pp. 14 – 17.

[20] N. Verma and A. P. Chandrakasan, "A 65nm 8T sub-Vt SRAM employing sense-amplifier redundancy," in *ISSCC*, 2007, pp. 328 –329.

[21] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," in *ISQED*, 2006, pp. 585 – 590.