A Fast Estimation of SRAM Failure Rate Using Probability Collectives

Fang Gong, Sina Basir-Kazeruni, Lara Dolecek, Lei He
Electrical Engineering
University of California, Los Angeles
Los Angeles, CA, 90095
{fang08, sinabk, dolecek, lhe}@ee.ucla.edu *

ABSTRACT

Importance sampling is a popular approach to estimate rare event failures of SRAM cells. We propose to improve importance sampling by probability collectives. First, we use "Kullback-Leibler (KL) distance" to measure the distance between the optimal sampling distribution and the original sampling distribution of variable process parameters. Further, the probability collectives (PC) technique using immediate sampling is adapted to analytically minimize the KL distance and to obtain a sampling distribution as close to the optimal as possible. The proposed algorithm significantly accelerates the convergence of importance sampling. Experiments demonstrate that proposed algorithm is $5200X$ faster than the Monte Carlo approach and achieves more than $40X$ speedup over other existing state-of-the-art techniques without compromising estimation accuracy.

Categories and Subject Descriptors: B.7.[Hardware]: - Integrated Circuits-Design Aids
General Terms: Algorithms, Verification
Keywords: SRAM, Failure probability, Importance Sampling, Kullback-Leibler distance

1. INTRODUCTION

It has become increasingly challenging to estimate the failure probability of SRAM cells under large-scale process variations, because SRAM bit-cell needs to be copied millions or billions of times as an array for higher integration density and the failure of a few cells could be catastrophic. Therefore, SRAM cell designs need to have extremely small failure probability [1, 2]. This failure is a rare event[3] that can only be captured with millions of samples through extremely long Monte Carlo (MC) simulations.

To avoid the expensive MC runs, importance sampling has been proposed based on the insight that only the "important samples" capturing relevant rare event (i.e., cell failure) can improve the estimation accuracy and further speed up the estimation convergence. This approach has been extensively used for rare event estimation problems [4, 5, 6, 7, 8, 9]. However, one critical issue that affects the efficiency of importance sampling is how to build an "optimal sampling distribution" so that more "important samples" of the relevant rare event can be chosen.

Many statistical methodologies have been developed to build the optimal sampling distribution for importance sampling and have been applied to failure rate estimation of SRAM cells [4, 5, 6, 7, 8, 9]. For example, [4] approximates the optimal sampling distribution by mixing a uniform distribution, the given sampling distribution and a "shifted" distribution centering around the failure region. Works in [5, 6] simply shift the mean values and keep the shape of original sampling distributions, and minimize the norm value of the shift vectors to find the optimal sampling distribution. The approach in [7] makes use of a "classifier" to block the Monte Carlo samples that are likely to satisfy the given performance constraints and runs simulations on remaining samples. In addition, "particle filtering"-based approach was proposed in [8] to tilt more samples towards the failure region. Moreover, it was recently proposed to adapt "Gibbs Sampling" in order to draw more failure region samples directly for improved performance [9]. While recent works made important advancements over the Monte Carlo approach, an efficient and low-complexity approach is still urgently needed to accurately estimate the failure rate of SRAM cells.

In this paper, we present a fast algorithm based on probability collectives (PC) method for the failure rate estimation of SRAM cells. First, "Kullback-Leibler (KL) distance" from probability theory [10] and information theory [11] is used to quantitatively measure the distance between the optimal sampling distribution and the given distribution of variable parameters. Then, a set of parameterized sampling distributions is analytically derived by minimizing the KL distance with a probability collective (PC) method using immediate sampling [12, 13], which is as close to the optimal sampling distribution as possible. Therefore, the convergence of the importance sampling approach can be significantly improved. The experimental results show that the proposed algorithm not only provides extremely high accuracy but also achieves $5200X$ speed-up over Monte Carlo. Moreover, the proposed method can be more than $40X$ faster than other state-of-the-art techniques (i.e., mixture importance sampling method [4] and spherical sampling method [6]).

Although the probability collective approach was initially

*This work was partially supported by Cisco, ICscape Inc, and UC Discovery Program.

developed in the statistics field [12, 13], it was previously unknown how to interface it to the importance sampling method for the failure analysis of SRAM cells. In fact, there are three major issues that need to be resolved: first, one particular type of parameterized distribution should be chosen in order to approximate the optimal sampling distribution. Second, it is important but difficult to initialize the parameterized sampling distribution. Third, the minimization of the KL distance involves complicated optimization problems and usually requires expensive computational efforts. To resolve these issues, we select a set of Gaussian distributions parameterized by mean and sigma, and adapt the "norm minimization" from [5, 6] to initialize the distribution by shifting the given sampling distribution towards the failure region. Moreover, the immediate sampling-based probability collectives method [12, 13] is used to analytically solve for the optimal parameterized sampling distributions for importance sampling. To the best of our knowledge, this is the first work on successfully developing the probability collectives based importance sampling method for the failure probability estimation of SRAM cells.

The rest of this paper is organized as follows. In Section 2, we provide the necessary background on importance sampling, KL distance and probability collectives methods. Section 3 contains more details of the required techniques in the proposed method for SRAM failure analysis. The experiments and further discussion are provided in Section 4 to validate the accuracy and efficiency of the proposed method. The paper is concluded in Section 5.

2. BACKGROUND

2.1 Importance Sampling

Let ξ_i $(i = 1, \cdots, m)$ be independent random variables with probability density function (PDF) given by $p(\xi_i)$, characterizing circuit parameters under process variations, such as the threshold voltage and effective channel length of transistors. The joint PDF of ξ_i is denoted as $p(\boldsymbol{\xi})$ and can be expressed as follows due to the independence property:

$$p(\boldsymbol{\xi}) = \prod_{i=1}^{m} p(\xi_i). \tag{1}$$

The j-th Monte Carlo sample $\boldsymbol{\xi}^j = (\xi_1^j, \cdots, \xi_m^j)$ consists of one sample from each random variable distribution.

Let $f(\boldsymbol{\xi})$ be the performance merit of interest, such as static noise margin of SRAM cell. This quantity typically needs to be evaluated with expensive transistor-level circuit simulation.

Let f_0 be the performance constraint; the circuit failure $\{f(\boldsymbol{\xi}) < f_0\}$ event is designed to be "rare". Thereby, the indicator function $I(\boldsymbol{\xi})$ is defined to identify pass/fail of $f(\boldsymbol{\xi})$ as:

$$I(\boldsymbol{\xi}) = \begin{cases} 0 & \text{if } f(\boldsymbol{\xi}) \geq f_0 \text{ (pass)}, \\ 1 & \text{if } f(\boldsymbol{\xi}) < f_0 \text{ (fail)}. \end{cases} \tag{2}$$

Therefore, the probability of failure ($\mathcal{P}_r$) is estimated in (3):

$$\mathcal{P}_r = \int I(\boldsymbol{\xi}) \cdot p(\boldsymbol{\xi}) d\boldsymbol{\xi}. \tag{3}$$

In general, $p(\boldsymbol{\xi})$ is known but $I(\boldsymbol{\xi})$ is unknown since the indicator function $I(\boldsymbol{\xi})$ cannot be evaluated explicitly. When $I(\boldsymbol{\xi})$ characterizes a failure region, extremely long Monte Carlo simulations on million samples of $\boldsymbol{\xi}$ are required.

To avoid massive Monte Carlo simulations, importance sampling has been proposed to sample from one "distorted" sampling distribution $g(\boldsymbol{\xi})$ that tilts towards the failure region where failures become more likely to happen. Then:

$$\mathcal{P}_r = \int I(\boldsymbol{\xi}) \cdot \frac{p(\boldsymbol{\xi})}{g(\boldsymbol{\xi})} \cdot g(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int w(\boldsymbol{\xi}) \cdot I(\boldsymbol{\xi}) \cdot g(\boldsymbol{\xi}) d\boldsymbol{\xi}. \tag{4}$$

where $w(\boldsymbol{\xi})$ re-weights each sample of $\boldsymbol{\xi}$ to convert the sample into the original sampling distribution. Theoretically, the optimal sampling distribution $g^{opt}(\boldsymbol{\xi})$ [9], where only one sample is needed to provide the accurate estimation of failure probability, can be expressed as:

$$g^{opt}(\boldsymbol{\xi}) = \frac{I(\boldsymbol{\xi}) \cdot p(\boldsymbol{\xi})}{\mathcal{P}_r} \tag{5}$$

However, $g^{opt}(\boldsymbol{\xi})$ cannot be evaluated with (5) directly because $I(\boldsymbol{\xi})$ is unknown and $\mathcal{P}_r$ is indeed the desired failure rate. Instead, another sampling distribution $h(\boldsymbol{\xi})$ should be created to provide an approximation as close to $g^{opt}(\boldsymbol{\xi})$ as possible. For example, the Kullback-Leibler distance can be used to define the distance between $h(\boldsymbol{\xi})$ and $g^{opt}(\boldsymbol{\xi})$.

2.2 Kullback-Leibler Distance

The Kullback-Leibler (KL) distance was proposed in probability theory [10] and information theory communities [11] to measure the *directional* distance from one distribution to another.

The KL distance from distribution $g^{opt}(\boldsymbol{\xi})$ in (5) to $h(\boldsymbol{\xi})$ is expressed as:

$$\mathbb{D}_{KL}(g^{opt}(\boldsymbol{\xi}), h(\boldsymbol{\xi})) = \mathbb{E}_{g^{opt}}\left[log\left(\frac{g^{opt}(\boldsymbol{\xi})}{h(\boldsymbol{\xi})}\right)\right]. \tag{6}$$

Note that both distributions g^{opt} and h should be defined over the same random variable $\boldsymbol{\xi}$. In addition, $\mathbb{E}[\cdot]$ denotes the expectation operator and the subscript g^{opt} indicates that $\mathbb{E}[\cdot]$ is taken with respect to distribution g^{opt}.

Therefore, it is desired to minimize $\mathbb{D}_{KL}(g^{opt}(\boldsymbol{\xi}), h(\boldsymbol{\xi}))$ in order to achieve $\hat{h}(\boldsymbol{\xi})$ as the best approximation of $g^{opt}(\boldsymbol{\xi})$. To this end, the probability collective method can be adapted to solve the minimization problem efficiently.

2.3 Probability Collectives

In general, probability collectives (PC) method is an efficient optimization framework [12, 13], which can search for the optimal probability distributions of variable parameters in order to optimize the objective function.

As an illustration, we consider random variables $\boldsymbol{\xi}$=(ξ_1, $\cdots$, ξ_m) and aim to minimize the KL distance as:

$$\arg\min \mathbb{E}_{g^{opt}}\left[log\left(\frac{g^{opt}(\boldsymbol{\xi})}{h(\boldsymbol{\xi})}\right)\right]. \tag{7}$$

By change of measure, the above minimization problem is equivalent to the statement in (8):

$$\arg\max \mathbb{E}_h\left[I(\boldsymbol{\xi}) \cdot log(h(\boldsymbol{\xi}))\right]. \tag{8}$$

It is highly prohibitive to perform exhaustive search for $h(\boldsymbol{\xi})$ since the search space is extremely large and contains arbitrary distributions. The PC method simplifies the search problem by utilizing a set of parameterized sampling distributions $h(\boldsymbol{\xi}, \theta)$ with additional parameters θ =

$(\theta_1, \cdots, \theta_m)$. As such, the maximization problem in (8) becomes:

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_h \left[I(\boldsymbol{\xi}) \cdot log(h(\boldsymbol{\xi}, \theta)) \right]. \quad (9)$$

where $\hat{\theta}$ is the optimal parameter of the distribution $h(\boldsymbol{\xi}, \theta)$ that leads the minimum KL distance in (7).

Note that the expectation value $\mathbb{E}_h[\cdot]$ in (9) cannot be evaluated with analytical formula and thereby sampling techniques must be used. In fact, several sampling based PC methods such as delay sampling based PC, and immediate sampling based PC were proposed in [12, 13].

In this paper, we adapt the immediate sampling based PC method as summarized in Algorithm (1). Interested readers are referred to [12, 13] for other PC methods.

Algorithm 1 Immediate Sampling based PC Algorithm

1: Choose the initial parameter $\theta^{(1)}$ to build parameterized sampling distributions $h(\boldsymbol{\xi}, \theta^{(1)})$.
2: Draw random samples from $h(\boldsymbol{\xi}, \theta^{(1)})$ and set iteration index number $t = 2$.
3: **repeat**
4: Evaluate values of indicator function $I(\boldsymbol{\xi})$ with chosen samples.
5: Solve for $\theta^{(t)}$ by:

$$\theta^{(t)} = \arg\max_{\theta} \mathbb{E}_h \left[I(\boldsymbol{\xi}) \cdot log(h(\boldsymbol{\xi}, \theta^{(t-1)})) \right].$$

6: Draw random samples from the parameterized distribution $h(\boldsymbol{\xi}, \theta^{(t)})$ and set $t = t + 1$.
7: **until** Converged (e.g., $\theta^{(t)}$ does not change for several subsequent iterations)
8: Obtain the optimum parameter $\hat{\theta}$ for the optimal sampling distribution.
9: Sample the final $h(\boldsymbol{\xi}, \hat{\theta})$ to get solution(s) in order to optimize the objective function.

Since the updated distribution $h(\boldsymbol{\xi}, \theta^{(t)})$ at the t-th iteration is sampled immediately, the procedure is called "immediate sampling" based PC method. However, there exist several issues that need to be resolved when immediate sampling PC method is used for failure analysis of SRAM cells:

- First, there exist many types of parameterized distributions (e.g., Gaussian distributions, Boltzmann distributions, etc.), and it remains unclear how to choose $h(\boldsymbol{\xi}, \theta)$ for the SRAM failure analysis.
- It is important and nontrivial to find $\theta^{(1)}$ which provides a "starting point" or a "heuristic initial solution" for the solution of (9). This quantity significantly affects the speed of convergence in Algorithm (1).
- The optimization problem in (9) is very difficult to solve and a closed-form solution is highly desired.

Therefore, it is of interest to develop an approach to use immediate sampling based PC method in a way that is suitable for SRAM failure analysis.

3. PROPOSED METHOD

In this section, we introduce several existing techniques and highlight our novel contributions that are needed to utilize the immediate sampling PC method for SRAM failure analysis.

3.1 Parameterized Distribution Selection

Before we move forward, let us first introduce the modeling of process variations in SRAM cells. In general, the variation sources of CMOS transistors can be threshold voltage V_{th}, effective channel length L_{eff} and other device parameters, but V_{th} variation is dominant so that the variability effects of other parameters are significantly dampened [2].

Moreover, V_{th} variations are typically modeled as independent random variables of Gaussian distributions [4, 5, 6, 7, 9]. As such, it is a natural choice to deploy a family of Gaussian distributions parameterized by mean (μ) and standard deviation (σ). In fact, parameterized Gaussian distributions can lead to a closed-form solution to the optimization problem in (9) as shown in following sections.

As an illustration, let ξ_i be the independent Gaussian random variable for i-th V_{th} variation source, which has the mean $\mu_i^{(0)}$ and the standard deviation $\sigma_i^{(0)}$. To build the parameterized Gaussian distribution for ξ_i, we shift the mean to $\hat{\mu}_i$ and reducing the standard deviation to $\hat{\sigma}_i$. This approach is motivated by the following insights:

- **Mean-shift** can tilt the sampling distribution towards the failure region where the rare failures are more likely to happen. This approach is similar to the finding in [5] and has been extensively used in previous works [5, 6, 8, 14, 15, 16].
- **σ-reduction** can concentrate the samples around a much smaller region where rare failures can happen with higher probability.

Therefore, the samples drawn from the parameterized Gaussian distribution $h(\xi_i, \hat{\mu}_i, \hat{\sigma}_i)$ are more likely to fail, and can thereby expedite the convergence of the failure probability estimation in the importance sampling. However, it is still unknown how to find the optimal parameters $\hat{\mu}_i$ and $\hat{\sigma}_i$ efficiently; this question will be investigated in following sections.

3.2 Parameterized Distribution Initialization

As discussed in Section 2.3, the first step is to initialize the parameters (μ_i, σ_i), which, in fact, provides a "starting point" or "heuristic initial solution" to search for the optimal parameters $(\hat{\mu}_i, \hat{\sigma}_i)$. As such, the initial parameters $\mu_i^{(1)}$ and $\sigma_i^{(1)}$ can significantly affect the efficiency of the iterative search in PC method or even lead to completely misleading results.

To this end, we propose an efficient initial parameter selection method inspired by the insights of "norm minimization" in [5], which can rapidly shift the given sampling distribution towards the failure region and make rare failures most likely to happen.

Assume random variables ξ_i follow Gaussian distributions $N(\mu_i^{(0)}, \sigma_i^{(0)})$. The proposed initial parameter selection can be summarized as following: first, a few hundred *uniformly-distributed* samples of ξ_i are generated using Quasi Monte Carlo method [17] in order to evenly cover the entire parameter range, such as the eight-sigma range from $(\mu_i^{(0)} - 4\sigma_i^{(0)})$ to $(\mu_i^{(0)} + 4\sigma_i^{(0)})$. Then, transistor level simulations are run on these samples and the failed samples are identified with given performance constraints. We can further choose one failed sample with the *minimum* L_2-norm and use its val-

ue as the initial parameter for $\mu_i^{(1)}$. In addition, the initial sigma parameter $\sigma_i^{(1)}$ can be the same as $\sigma_i^{(0)}$.

It is worthwhile to point out that the above "norm minimization" based method is a heuristic for obtaining an *initial* parameterized Gaussian distribution but cannot provide the *optimal* sampling distribution $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$ in (9) by any means. The optimization problem in (9) should be solved for $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$ and an efficient closed-form approach is needed.

3.3 Closed-Form Optimization Solution

Before we present the closed-form solution, it should be noted that the optimization in (9) must be revised as (10) because samples are generated from the parameterized distributions $h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ rather than from distributions $h(\boldsymbol{\xi})$:

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \arg\max_{\boldsymbol{\mu}} \mathbb{E}_h[I(\boldsymbol{\xi}) \cdot w(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \cdot log(h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma}))], \\ \hat{\boldsymbol{\sigma}} &= \arg\max_{\boldsymbol{\sigma}} \mathbb{E}_h[I(\boldsymbol{\xi}) \cdot w(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \boldsymbol{\sigma}) \cdot log(h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \boldsymbol{\sigma}))] \end{aligned} \quad (10)$$

where $w(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ denotes the weights to unbias the samples from the parameterized distribution $h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ and can be expressed as:

$$w(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{h(\boldsymbol{\xi})}{h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma})}. \quad (11)$$

For the illustration purpose, let us consider following example:

- $\boldsymbol{\xi} = (\xi_1, \cdots, \xi_m)$: independent random Gaussian variables.
- $h(\boldsymbol{\xi}) = (h(\xi_1), \cdots, h(\xi_m))$: the given Gaussian sampling distributions of ξ.
- $h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = (h(\xi_1, \mu_1, \sigma_1), \cdots, h(\xi_m, \mu_m, \sigma_m))$: the chosen parameterized Gaussian distributions for $\boldsymbol{\xi}$.
- $\xi_i^1, \cdots, \xi_i^j, \cdots, \xi_i^N$: the samples of ξ_i drawn from the parameterized Gaussian distribution $h(\xi_i, \mu_i, \sigma_i)$.

As such, the weights of j-th sample $\boldsymbol{\xi}^j = (\xi_1^j, \cdots, \xi_m^j)$ can be expressed as:

$$w(\boldsymbol{\xi}^j, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{h(\xi_1^j) \times \cdots \times h(\xi_m^j)}{h(\xi_1^j, \mu_1, \sigma_1) \times \cdots \times h(\xi_m^j, \mu_m, \sigma_m)}. \quad (12)$$

Moreover, the expectation value $\mathbb{E}_h[\cdot]$ in (10) cannot be evaluated directly in general, because there is no analytical formula for the integral operation, and sampling methods must be used. For instance, with the samples $\xi_i^j, (j = 1, \cdots, N)$, the optimization problem for μ_i becomes the sampled form as (13). Similar expression can be derived for σ_i.

$$\hat{\mu}_i = \arg\max_{\mu} \frac{1}{N} \sum_{j=1}^{N} \Big(I(\boldsymbol{\xi}^j) w(\boldsymbol{\xi}^j, \boldsymbol{\mu}, \boldsymbol{\sigma}) \log(h(\xi_i^j, \mu_i, \sigma_i)) \Big). \quad (13)$$

As proposed in [12], the above optimization problem is a convex optimization problem that can be solved with closed-form formula, because the parameterized distribution $h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$, following Gaussian distribution, is a log-concave distribution.

Specifically, the optimal parameters $\hat{\mu}_i$ and $\hat{\sigma}_i$ can be analytically solved with closed-form formulae as [12, 13]:

$$\hat{\mu}_i = \frac{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \boldsymbol{\mu}, \boldsymbol{\sigma}) \times \xi_i^j}{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \boldsymbol{\mu}, \boldsymbol{\sigma})}. \quad (14)$$

where $\hat{\mu}_i$ can be asymptotically approached by iteratively updating the parameter $\boldsymbol{\mu}$ and evaluating the above formula. In practice, the iterative process can converge very fast within only a few iterations. Note that [11, 14, 15, 16] use the identical analytical formula to find the optimal parameter for mean shift.

Similarly, the closed-form formula can be derived to analytically compute $\hat{\sigma}_i$ as:

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \hat{\boldsymbol{\mu}}, \boldsymbol{\sigma}) \times (\xi_i^j - \hat{\mu}_i)^2}{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \hat{\boldsymbol{\mu}}, \boldsymbol{\sigma})}}. \quad (15)$$

It is obvious that the calculation of $\hat{\sigma}_i$ depends on the optimization result $\hat{\boldsymbol{\mu}}$ from (14). In other words, the potential error from the optimization of $\hat{\boldsymbol{\mu}}$ can propagate into the computation of $\hat{\sigma}_i$ and lead to completely misleading results, which is especially undesired because the performance of importance sampling is highly sensitive to the sampling distribution. This observation can further validate the necessity of the initial parameter selection presented in previous section.

Therefore, the optimal sampling distribution is obtained as $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$, which can be finally sampled to estimate the probability of SRAM rare event failures in the importance sampling to provide significant improvement on both accuracy and efficiency.

3.4 Overall Algorithm Flow

The proposed algorithm for the SRAM failure analysis is based on the above techniques. The overall algorithm flow is described in Algorithm (2), which consists of three stages:

(1) **Parameterized distribution initialization**: The first stage initializes the parameterized sampling distribution $h(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ as a "heuristic initial solution" to search for the optimal parameterized sampling distribution $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$. Initialization adopts the insight of "norm minimization" from [5] and shifts the given sampling distribution towards the failure region where SRAM failures are more likely to happen.

(2) **Optimal parameter evaluation**: This stage starts with the initial parameterized sampling distribution and analytically solves the optimization problems in (14) and (15) to achieve the optimal parameterized sampling distribution $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$.

(3) **Failure probability estimation**: The conventional importance sampling method is performed with the obtained optimal sampling distribution $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$ to estimate the failure rate of SRAM cells.

As shown in Section 4, the proposed approach in Algorithm (2) can provide more than $40X$ speedup over the existing state-of-the-art techniques and be up to $5200X$ faster than Monte Carlo method without compromising any accuracy.

4. EXPERIMENTAL RESULTS

We have implemented our proposed algorithm using MATLAB and Hspice with BSIM4 model. Also, Monte Carlo (MC), spherical sampling (SS) [6] and mixture importance sampling (MixIS) [4] are all implemented. As an illustration, the threshold voltages of all MOSFETs are considered as variation sources and static noise margin (SNM) failure

Algorithm 2 Overall Algorithm for SRAM Failure Analysis

Input: random variables $\boldsymbol{\xi} = (\xi_1, \cdots, \xi_M)$ with given Gaussian distributions $h(\boldsymbol{\xi}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{(0)})$, and sample counts (N_1, N_2, N_3).

Output: the estimation of failure probability $\mathcal{P}_r$.

/* Stage 1: Initial Parameter Selection */
Draw uniformly-distributed samples $\xi^j (j = 1, \cdots, N_1)$ from the given distributions $h(\boldsymbol{\xi})$ and run simulations on these samples.
Identify samples that fail with given performance constraints and calculate their L_2-norm values.
Choose the failed sample with the minimum L_2 norm and use the value of this sample as the initial $\boldsymbol{\mu}^{(1)}$.
Set the initial sigma $\boldsymbol{\sigma}^{(1)}$ to be the same as given $\boldsymbol{\sigma}^{(0)}$.

/* Stage 2: Optimal Parameter Finding */
Draw N_2 samples ξ^j from the initial parameterized distribution $h(\boldsymbol{\xi}, \boldsymbol{\mu}^{(1)}, \boldsymbol{\sigma}^{(1)})$ and set the iteration index number $t = 2$.
repeat
  Evaluate the indicator function $I(\xi^j)$ in (14) and (15) with these samples.
  for $i = 1 \rightarrow M$ **do**
    Solve for $\mu_i^{(t)}$ and $\sigma_i^{(t)}$ with

$$\mu_i^{(t)} = \frac{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\sigma}^{(t-1)}) \times \xi_i^j}{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\sigma}^{(t-1)})}.$$

$$\sigma_i^{(t)} = \sqrt{\frac{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\sigma}^{(t-1)}) \times (\xi_i^j - \mu_i^{(t)})^2}{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\sigma}^{(t-1)})}}.$$

  end for
  Draw N_2 samples from the updated parameterized distribution $h(\boldsymbol{\xi}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)})$ and set $t = t + 1$.
until Converged; when $\boldsymbol{\mu}^{(t)}$ and $\boldsymbol{\sigma}^{(t)}$ do not change for several subsequent iterations.
Obtain the optimal parameter $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$ for parameterized sampling distribution.

/* Stage 3: Failure Probability Estimation */
Draw N_3 samples from the obtained optimal sampling distribution $h(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$.
Run simulations on these samples ξ_j and evaluate the indicator function $I(\xi^j), (j = 1, \cdots, N_3)$.
Solve for the failure probability, $\mathcal{P}_r$, with sampled form:

$$\mathcal{P}_r = \frac{1}{N_3} \sum_{i=1}^{N_3} I(\xi^j) \times w(\xi^j, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}).$$

where $w(\xi^j, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$ is the weight for sample ξ^j and is defined as

$$w(\xi^j, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}) = \frac{\prod_{i=1}^{M} h(\xi_i^j)}{\prod_{i=1}^{M} h(\xi_i^j, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})}.$$

is studied. Note that the same algorithm can be applied to other variation sources (i.e. L_{eff}, T_{ox}, etc.) and other rare failures (i.e. reading/writing failures) as well.

4.1 SRAM Cell and Static Noise Margin

The typical design of a 6-transistor SRAM cell is shown in Fig.1. We introduce process variations to threshold voltage V_{th} of all MOSFETs as *independent* random variables of Gaussian distributions. Specifically, the nominal mean values of the threshold voltages for NMOS and PMOS are $0.466V$ and $-0.4118V$, respectively. The standard deviations (σ) of threshold voltage variations are 10% of nominal threshold voltage values.

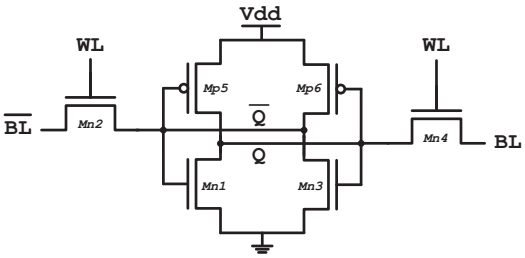


Figure 1: The schematic of the 6T SRAM cell.

The SRAM cell consists of six transistors: $Mn2$ and $Mn4$ control the access of the cell during reading, writing and standby operations; the remaining four transistors form two inverters and use two stable states (either '0' or '1') to store the data in this memory cell.

Static Noise Margin (SNM) is used to evaluate the stability of SRAM cell by describing the noise voltage that is needed to flip the stored data. More specifically, SNM can be measured by the length of maximum embedded square in the butterfly curves, which consist of the voltage transfer curve (VTC) of the two inverters in SRAM cell [18]. As such, when SNM is less than zero, the butterfly curve is collapsed and the data retention failure happens.

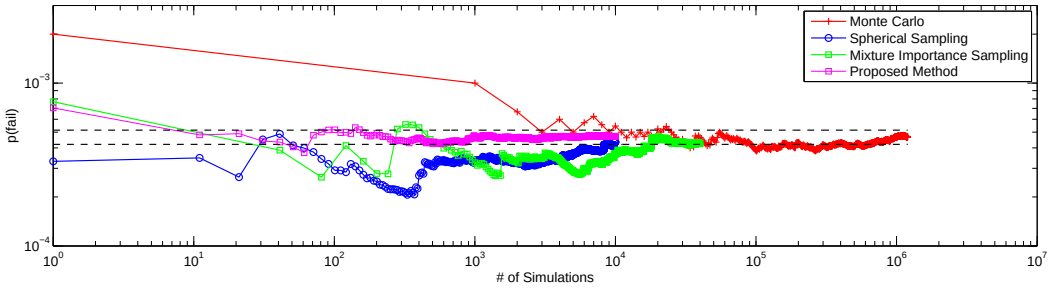
4.2 Accuracy Comparison

4.2.1 Comparison of Failure Rate Estimation

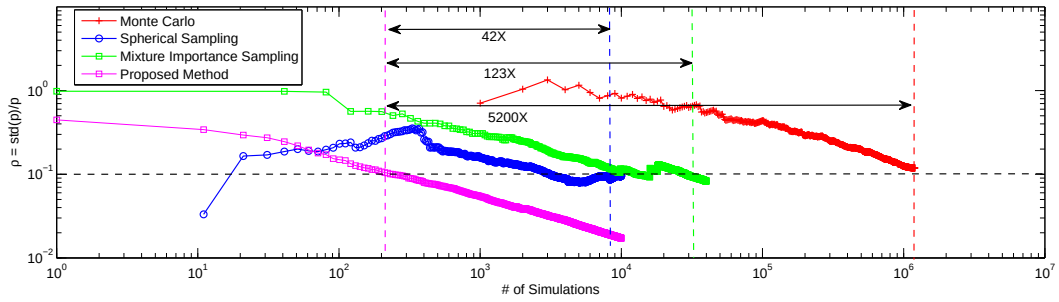
To validate the estimation accuracy of the proposed algorithm, we perform different methods, including Monte Carlo (MC), mixture importance sampling (MixIS) [4], spherical sampling (SS) [6], and the proposed algorithm on the same 6-T SRAM cell example in 45nm process to predict the probability of data retention failure due to SNM variation. Here, we choose $Vdd = 300mV$ as an example for comparison.

Evolutions of the probability estimation from different methods are plotted in Fig.2(a), the following observations can be made:

- First, the failure rate estimations from different methods closely match each other, which validates the estimation accuracy of our proposed method.
- Second, the proposed method in contrast to other methods starts with an estimation that is very close to the final accurate result, because it can find the *optimal* sampling distribution using probability collectives method for importance sampling.



(a) failure probability ($Vdd = 300mV$)



(b) figure of merit ($Vdd = 300mV$)

Figure 2: Evolution comparison of the failure probability estimation and figure of merit for different methods.

- The comparisons among MixIS, SS and proposed method also reveal that the importance sampling is highly sensitive to the sampling distribution, which can affect both the accuracy and efficiency. This is the very motivation behind this paper to exploit the optimal sampling distribution.

4.2.2 Comparison of Figure-Of-Merit (FOM)

As stated in [5, 6], Figure-Of-Merit (FOM), ρ, has been extensively used to quantify the accuracy of probability estimation, which is defined as:

$$\rho = \frac{\sqrt{\sigma^2_{\mathcal{P}_r}}}{\mathcal{P}_r}. \tag{16}$$

where $\mathcal{P}_r$ is the estimation of failure probability and $\sigma_{\mathcal{P}_r}$ is the standard deviation of $\mathcal{P}_r$. In fact, the FOM can be treated as a *relative error* so that a smaller figure of merit means higher accuracy.

Similarly, we further calculate the evolution of FOM for different methods which are plotted in Fig.2(b). To clearly compare the accuracy of different methods, we plot a dashed line to indicate the 90% accuracy level with 90% confidence interval ($\rho = 0.1$). Two important observations can be made:

- MixIS, SS and proposed method quickly reach higher accuracy level ($> 90\%$) while Monte Carlo can only closely approach the 90% accuracy. This is because importance sampling based methods can choose more failed samples from the failure region to efficiently improve the accuracy, while Monte Carlo method wastes a large number of samples that are far from the failure region.

Table 1: Results of all methods with $10,000$ samples.

	MC	MixIS [4]	SS [6]	Proposed
prob. of failure	5.455E-4	3.681E-4	4.342E-4	4.699E-4
ρ	0.8129	0.1111	0.9831	0.021
accuracy	18.71%	88.53%	90.42%	98.2%
#runs	1.0e+4	1.0e+4	1.0e+4	1.0e+4

Table 2: Accuracy and efficiency comparison for different methods.

	MC	MixIS [4]	SS [6]	Proposed
prob.(failure)	4.675E-4	4.332E-4	4.208E-4	4.7E-4
accuracy	88%	90%	90%	90%
#runs	1.2E+6	2.85E+4	9.771E+3	231
speedup	1X	42X	123X	5200X

- For the same number of samples, the proposed method outperforms existing approaches in terms of the estimator accuracy. For instance, we compare accuracy level of all different methods in Table(1) with only $10,000$ samples. In this table, the proposed method can provide 98.2% accuracy while other methods can only reach up to 90.42%, which is attributed to the choice of the optimal sampling distribution.

4.3 Efficiency Comparison

4.3.1 Comparison of Convergence Speed

Fig.2(b) illustrates the efficiency of proposed algorithm, which is shown to have the fastest speed of convergence among all the different methods. In this figure, the proposed

method chooses more failed samples and increasingly improves the accuracy to an extremely high level due to the optimal sampling distribution.

Similar observations can be made from Fig.2(a): the proposed method starts with the estimation that is very close to the final accurate results and quickly converges to the 95% confidence interval of the final Monte Carlo result (denoted by two dashed lines). Meanwhile, the estimations of other methods keep fluctuating before asymptotically approaching the final accurate results.

In fact, the proposed method can achieve 90% accuracy and 90% confidence interval with only 231 samples. In the contrast, MixIS and SS need 2.85e+4 and 9.77e+3 samples to reach the same accuracy level, respectively. Monte Carlo method cannot even reach 90% accuracy with up to 1.2e+6 samples. In other words, the proposed method can achieve $5200X$ speedup over Monte Carlo, $123X$ speedup over MixIS [4] and $42X$ speedup over SS [6].

4.3.2 Other Efficiency Comparison

It should be noted that all importance sampling based methods require some "extra" samples to find the new sampling distribution, called "extra" because Monte Carlo method does not need these extra samples in simulations. For example, the stage 1 and stage 2 in Algorithm (2) need some "extra" samples to construct the optimal sampling distribution before the failure probability can be estimated in stage 3.

Specifically, in our experiments, the MixIS needs 3000 samples to find the sampling distribution, because it mixes the uniform distribution, given sampling distribution and mean-shifted distribution together and requires more samples. The SS method needs 2000 samples to locate the failed samples with a minimum L_2-norm in a spherical manner. The proposed method also needs 2000 samples to find the optimal sampling distribution. However, these "extra" samples turn out to be negligible when compared to the Monte Carlo method.

5. CONCLUSION

In this paper, we presented an improved importance sampling algorithm based on probability collectives method to efficiently estimate the rare event failures of SRAM cells. This method adopts the "Kullback-Leibler (KL) distance" to represent the distance between the optimal sampling distribution and a given sampling distribution. The KL distance is further analytically minimized using immediate sampling based probability collectives method and a set of parameterized Gaussian distributions are obtained as the optimal sampling distribution. The experiments demonstrate that proposed algorithm can provide extremely high accuracy and dramatically improve the convergence of importance sampling. For instance, the proposed method can be $5200X$ faster than Monte Carlo method and offer more than $40X$ speedup over other existing state-of-the-art techniques (e.g., mixture importance sampling [4] and spherical sampling [6]) with the same accuracy.

6. REFERENCES

[1] R. Heald and P. Wang, "Variability in sub-100nm SRAM designs," in *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pp. 347–352, 2004.

[2] P. Girard, A. Bosio, L. Dilillo, P. S., and A. Virazel, "Advanced test methods for SRAMs: Effective solutions for dynamic fault detection in nanoscaled technologies," 2009.

[3] K. Agarwal and S. Nassif, "Statistical analysis of SRAM cell stability," in *Proceedings of the 43rd annual Design Automation Conference*, DAC '06, pp. 57–62, 2006.

[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," in *Proceedings of the 43rd annual Design Automation Conference*, DAC'06, pp. 69–72, 2006.

[5] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '08, pp. 322–329, 2008.

[6] 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," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, pp. 801 –806, 2010.

[7] A. Singhee and R. Rutenbar, "Statistical Blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application," in *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, pp. 1–6, 2007.

[8] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato, "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis," in *IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '10, 2010.

[9] C. Dong and X. Li, "Efficient SRAM failure rateprediction via Gibbs sampling," in *Proceedings of the 43rd annual Design Automation Conference*, DAC'11, 2011.

[10] V. K. Rohatgi and A. K. M. Ehsanes Saleh, "An introduction to probability and statistics," *Wiley-Interscience*, 2000.

[11] T. M. Cover and J. A. Thomas, "Elements of information theory," *John Wiley and Sons*, 1991.

[12] D. Rajnarayan, D. H. Wolpert, and I. Kroo, "Optimization under uncertainty using probability collectives," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006.

[13] D. Rajnarayan, I. Kroo, and D. H. Wolpert, "Probability collectives for optimization of computer simulations," *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007.

[14] A. Ridder and R. Y. Rubinstein, "Minimum cross-entropy methods for rare-event simulation," *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 83, pp. 769–784, 2007.

[15] T. H. de Mello, "A study on the cross-entropy method for rare event probability estimation," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 381–394, 2007.

[16] P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross entropy method," *Annals of Operations Research*, vol. 134, pp. 19–67, 2005.

[17] H. Niederreiter, "Random number generation and quasi-monte carlo methods," *Society for Industrial and Applied Mathematics*, 1992.

[18] D. Mukherjee, H. K. Mondal, and B. Reddy, "Static noise margin analysis of SRAM cell for high speed application," *International Journal of Computer Science Issues*, 2010.