UNIVERSITY OF CALIFORNIA

Los Angeles

# A Fast Method for SRAM Failure Estimation

A thesis submitted in partial satisfaction

of the requirements for the degree of Master of Science

in Electrical Engineering

by

Min Gao

2012

ABSTRACT OF THE THESIS

# A Fast Method for
# SRAM Failure Estimation

by

Min Gao

Master of Science in Electrical Engineering

University of California, Los Angeles, 2012

Professor Lei He, Chair

The SRAM cell is an important memory component that is widely used in integrated circuit design. Its performance is crucial to the entire circuit. However, inevitable process variations have introduced significant changes in the performance of fabricated SRAM cells and led to severe circuit failure. Consequently, the failure probability of an SRAM cell must be kept extremely small. These extremely small probability events are considered to be “rare events”. The most straightforward method of estimating failure probability is using the classical Monte

Carlo method. However, this method is extremely impractical in the case of rare events because of its drastically long run time. Therefore, a method to efficiently estimate the failure probability of rare events is strongly desired.

In this thesis, a novel and fast failure analysis for SRAM cells is proposed to efficiently and accurately estimate the failure probability of a rare event. The proposed approach is based on Probability Collective based Importance Sampling. This approach increases the convergence rate by finding the closest approximation of the optimal distribution used in Importance Sampling. In order to find the closest approximation of the optimal sampling distribution, the proposed method minimizes the Figure of Merit of the estimated probability in each iteration. Experimental results show that the proposed method can have an average of 5x speed up compared to Probability Collective based Importance Sampling. Moreover, multiple trials between these two methods show that the proposed method offers a faster convergence rate and greater stability.

The thesis of Min Gao is approved.

Lara Dolecek

Puneet Gupta

Lei He, Committee Chair

University of California, Los Angeles

2012

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Professor Lei He, for his guidance and advice as I worked on this project. I would like to acknowledge the work of Fang Gong, Sina Basir-Kazeruni and Lara Dolecek [GBD12] who developed the theory of the Probability Collectives based Importance Sampling algorithm and developed an early implementation so that I could have a very nice foundation to make some improvements. I would also like to acknowledge Qirong Wang from Fudan University for her hard work and creative thoughts on this project. Finally, I would like to thank Rahul Krishnan and Wenyao Xu for their kind suggestions and comments on this thesis.

# CHAPTER 1

# INTRODUCTION

Static Random Access Memory (SRAM) cells are widely used by a vast amount of semiconductor devices. SRAM cells have many advantages compared to other types of memory cells. For example, they do not need to periodically refresh data and they also consume less power but operate at faster speeds. However, as technology pushes the limit for transistor sizes, process variations caused by random fluctuations during fabrication in sub-micron technology can significantly affect the performance of SRAM cells. If millions of copies of SRAM cells are connected together, an error in a single SRAM cell could trigger a catastrophic error in the system [HW04] [GBD09]. Therefore, the failure probability of SRAM cells should be extremely small in order to achieve a very robust design. Consequently, evaluating SRAM failure rate is becoming more and more critical.

Clearly, the failure of an SRAM cell should be classified as a rare event [AN06]. A rare event is generally estimated by transistor level Monte Carlo simulation with the help of a numerical solver such as SPICE. However, simply using Monte Carlo simulation to directly estimate failure rate is extremely slow because the failure region for a transistor is very small and not close to the expected value of transistor performance. Consequently, only a handful of results from the millions of Monte Carlo simulated samples fall into the failure region. This means that it is very time-inefficient to gather enough Monte Carlo results that fall into the failure region to give us an accurate prediction.

In order to overcome the drawback of extremely long simulation times, many statistical methods have been proposed to quickly estimate the failure rate without losing much accuracy. Some methods attempt to find ways to avoid unnecessary sampling regions [SR07] while others attempt to use an importance sampling method to shift the original distribution to an alternative distribution such that more samples can be sampled in the failure region of the alternative distribution [DQS08] [DL11]. However, all of these methods tend to be high in complexity and are often inefficient for high-dimension cases. Therefore, an efficient and low-complexity method is still needed to estimate the failure rate of SRAM cells in an accurate manner.

In this thesis, a method based on [GBD12] is proposed. The proposed method aims to increase the convergence rate of failure probability estimation. First, an initial distribution is constructed by uniformly sampling and optimization technology. Then, a set of parameterized sampling distributions is analytically derived by minimizing the value of the FOM (Figure of Merit) obtained in every iteration, making the distributions as close to the optimal sampling distributions as possible. Consequently, the convergence rate for importance sampling is greatly increased. Finally conventional Importance Sampling is implemented to obtain failure probability. Experiments show that the proposed method achieves an average of 5x speed-up over probability collectives based importance sampling while maintaining the same accuracy [GBD12].

The contribution for the proposed method is that it successfully accelerates the convergence rate of failure probability estimation without loss of accuracy. At the same time, multiple trials between the proposed method and Probability Collective based Importance Sampling (PCIS) show that the proposed method has a faster convergence rate and greater stability.

The rest of the thesis is organized as follows. Chapter 2 provides background information on SRAM cells and its reading failure, Monte Carlo Simulation, importance sampling, Figure of Merit and Kullback-Leibler Distance. Chapter 3 contains a brief introduction on previous works in rare event estimation. Chapter 4 gives the proposed algorithm for SRAM failure estimation. Chapter 5 gives experimental results and further discussion. Chapter 6 concludes the thesis.

# CHAPTER 2

# BACKGROUND

## 2.1 SRAM and SRAM Reading Failure

Static Random Access Memory (SRAM) is a semiconductor memory device that is now widely used in the electrical manufacturing field. A typical design of a 6-transistor SRAM cell is shown in Figure 1. In the SRAM reading process, two bit lines will be charged with a logic 1, then two word lines will be charged to open up M5 and M6. If logic 1 is stored in $Q$ and 0 in $\overline{Q}$, then $M1$ and $M4$ will open, making $BL$ retain logic 1 while $\overline{BL}$ will be discharged to logic 0. On the other hand, if 0 is stored in $Q$, then $M2$ and $M3$ will open, making $BL$ discharge to logic 0 and $\overline{BL}$ retain logic 1 [SRA]. There is also a sense amplifier connected to the SRAM cells to sense the voltage between two bit lines. An SRAM reading failure means that this difference is less than a specified threshold because the sense amplifier has an inherent and randomly distributed offset voltage. The amplifier can only sense a voltage greater than this offset voltage [YAL09]. This kind of error will lead to a circuit working in a wrong state or a severe error.

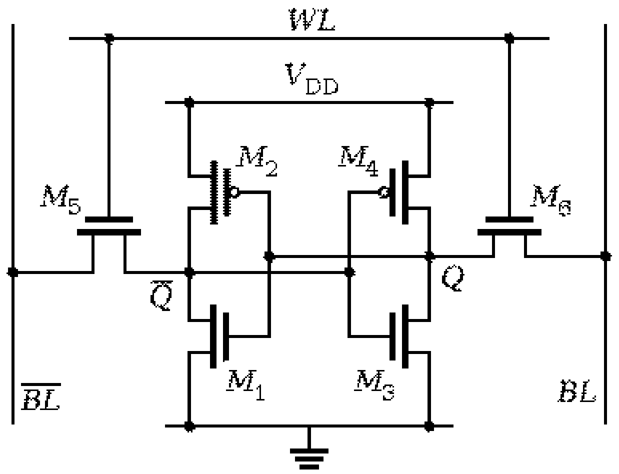


Figure 1 Schematic: 6-transistor SRAM cell

## 2.2 Monte Carlo Simulation

Monte Carlo simulation is a method that performs repeated random sampling and simulation to estimate the probability of a certain event. A typical Monte Carlo simulation has four steps. The first step is certifying the sample domain and deciding the input variable and confirming the input distribution. Then, a large number of input samples are randomly generated based on the distribution in the previous step. Next, these samples are simulated by a numerical solver. For example, we calculate the difference between two bit lines in a SRAM cell by generating random inputs of threshold voltage and evaluating them using a closed form circuit model in SPICE. The final step is getting the expected probability based on the numerical results obtained above[MCM].

One example to illustrate the typical steps for Monte Carlo simulation is calculating the value of $\pi$ . The first step is to draw a square on the ground and then draw a circle to inscribe the square. Here, the sample domain would be the square. The next step is to uniformly scatter some objects of the same size inside the square. This means we assume the input sample would follow a uniform distribution because all scattered objects are the same size. Then, the number of objects falling into the circle and falling into the square is counted. This step is a deterministic calculation for the inputs. Finally the number of samples in the circle divided by the number of samples in the whole square, all multiplied by 4 approximates the value of $\pi$ .

Let $\xi_i(i=1,...,m)$ be the independent random variables which have a probability density function (PDF) already given by $p(\xi_i)$ . The joint PDF of $\xi_i$ , denoted as $p(\xi)$ , can be expressed as

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

Let $f_0$ be the performance constraint and $f(\xi)$ be the set of performance values after deterministic calculation for every sample point. The indicator function $I(\xi)$ is used to identify each sample point as pass or fail

$$I(\xi)=\begin{cases}1 & if\{f(\xi)>f_0\} \quad pass\\ 0 & if\{f(\xi)\le f_0\} \quad fail\end{cases} \tag{2}$$

Therefore the event probability from Monte Carlo simulation is given by

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

## 2.3 Importance Sampling

Although Monte Carlo simulation is a trustworthy method and widely used in various areas to serve as a standard method for estimation, there are still some drawbacks on it. The most significant drawback is for a rare event, an event with the probability of 1e-5 or lower, Monte Carlo simulation needs to sample an extremely large number of samples to get a relatively accurate estimation. This process results in a very long runtime for simulation. Importance Sampling has been proposed with the aim to increase the sampling speed without loss of much accuracy [IS].

Importance Sampling samples the input variables with another distribution $g(\xi)$ instead of the original distribution $p(\xi)$ . The new distribution $g(\xi)$ is formed so that a larger number of samples will fall into the region where the specific event will happen [GBD12]. In this case,

$$P = \int I(\xi) \cdot \frac{p(\xi)}{g(\xi)} \cdot g(\xi) d\xi = \int I(\xi) \cdot \omega(\xi) \cdot g(\xi) d\xi \tag{4}$$

where $\omega(\xi)$ is the weight function and will convert the samples from the new distribution utilizing Importance Sampling to the original sampling type. Theoretically, only one sample is needed to estimate the probability of a rare event if a proper distribution $g_{opt}(\xi)$ is found, and can be expressed as

$$g_{opt}(\xi) = \frac{I(\xi) \cdot p(\xi)}{P} \tag{5}$$

However, it is usually impossible to directly evaluate $g_{opt}(\xi)$ because the indicator function $I(\xi)$ is unknown and $P$ is the probability that we want to evaluate from Importance Sampling. In practice, we will use a different distribution $h(\xi)$ which optimally approximates $g_{opt}(\xi)$ as the distribution to perform sampling.

## 2.4 Figure of Merit (FOM)

Figure of Merit is a very important measurement variable in all simulation methods. It decides when to stop the simulation. It is represented by equation (6)

$$\rho(P) = \frac{\sqrt{VAR(P)}}{P} \tag{6}$$

Where $P$ is the event probability and $VAR$ is the variance at the end of $N$ samples of the estimator.

For Monte Carlo simulation, the variance is

$$VAR(P_{MC}) = \frac{1}{N}(P_{MC} - P_{MC}^2) \tag{7}$$

For importance sampling, the variance is

$$VAR(P_{IS}) = \frac{1}{N^2}(\sum_{i=1}^{N}\omega(\hat{X}_i)^2 I(\hat{X}_i)^2 - NP_{IS}^2) \tag{8}$$

A typical illustration for Figure of Merit is that stopping the simulation where $\rho(P) \le \varepsilon\sqrt{\log(1/\delta)}$ implies that the estimate of $P$ has $(1-\varepsilon)100\%$ accuracy with at least $(1-\delta)100\%$ confidence level. For example, if a simulation algorithm stops when the Figure of Merit $\rho(P) = 0.1$, we can say we have reached 90% accuracy together with a 90% confidence level.

## 2.5 Kullback-Leibler Distance

Kullback-Leibler(KL) Distance is a famous concept in information theory[RE00] [CT91]that measures the distance from one distribution to another distribution. In Importance Sampling, a new distribution $h(\xi)$ is obtained to approximate the optimal distribution $g_{opt}(\xi)$. To obtain the closest approximation of $g_{opt}(\xi)$ using $h(\xi)$, we minimize the K-L Distance between them as follows:

$$\min \quad D_{KL}(g_{opt}(\xi), h(\xi)) = E_{g_{opt}}[\log(\frac{g_{opt}(\xi)}{h(\xi)})] \tag{9}$$

Where $E$ denotes the expectation operator and the subscript means expectation over $g_{opt}(\xi)$.

# CHAPTER 3

# PREVIOUS WORKS

Many statistical methodologies have been proposed to approximate the optimal sampling distribution for Importance Sampling.

[DQS08] tries to find the distribution with the aim of reducing the variance of the estimator. It simply shifts the mean values and keeps the shape of the original sampling distributions. Then, it minimizes the L-2 norm value of the shift vectors to find the optimal sampling distribution $g_{opt}(\xi)$ . Finally, the conventional Importance Sampling technique is implemented to get the estimated failure probability.

[KJN06] is also a series of importance sampling based methods which tries to approximate the optimal sampling distribution $g_{opt}(\xi)$ by mixing three different distributions----a uniform distribution, the given sampling distribution, and a new distribution whose center is around the failure region.

[DL11] proposes an adaptive Gibbs sampling method which samples a sequence of single dimensional PDFs, instead of multi-dimensional joint PDFs, to generate many samples to allow the samples to follow the optimal distribution. Then, the mean and covariance matrix of these Gibbs samples are calculated to determine a multivariate distribution for approximating the optimal distribution $g_{opt}(\xi)$. Finally it generates samples following $g_{opt}(\xi)$ and then evaluates the failure probability.

[SR07] combines the ideas of data mining and Extreme Value Theory to efficiently filter unnecessary samples. This method blocks some samples that are not located in the desired event's region. The method first performs initial sampling, which can be either Monte Carlo or Importance Sampling, to start building a classifier. Then, a proper threshold with a safety margin using support vector machine (SVM) is chosen to complete the classifier. Next, simulation generates more samples using Monte Carlo method. Finally, calculation on the points that have not been filtered by the classifier is performed to speed up the overall method. The main process is illustrated in Figure 2.

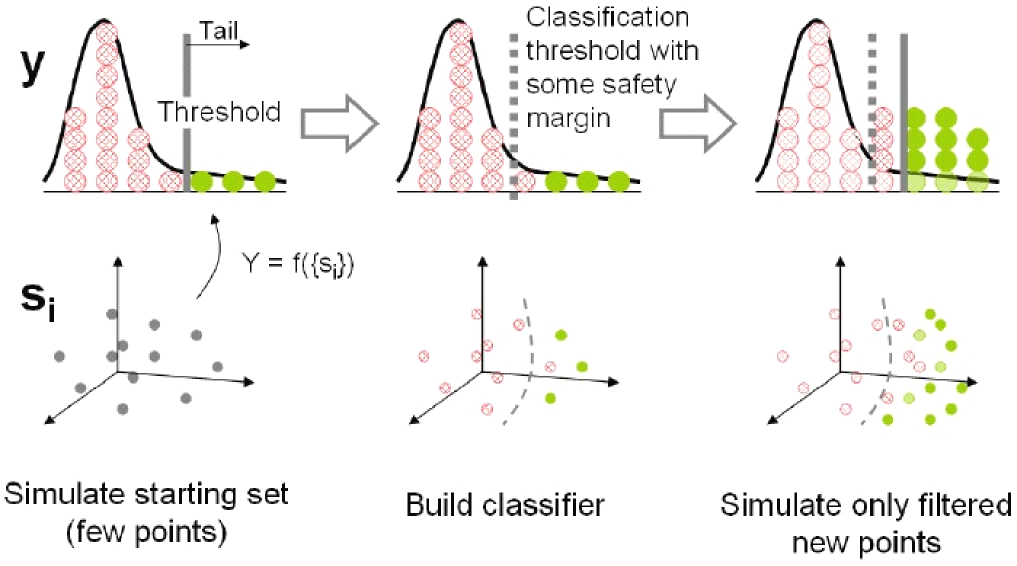

y
Tail
Threshold
Classification threshold with some safety margin
$s_i$
Y = f({$s_i$})
Simulate starting set (few points)
Build classifier
Simulate only filtered new points

Figure 2 Illustration: construction of EVT filter

[KHT10] applies a particle (sample) filter to search for a failure region regardless of the shape and probability distribution. Then, it uses Importance Sampling to estimate failure probability. It has three stages: the first stage is to predict each particle's position by using the particles past data. The next stage is to calculate the weight of each particle. The final stage is to perform resampling to eliminate particles with lower weight, meaning these sample points are far away from the failure region. Using these three stages, particles iteratively tilt toward the failure region.

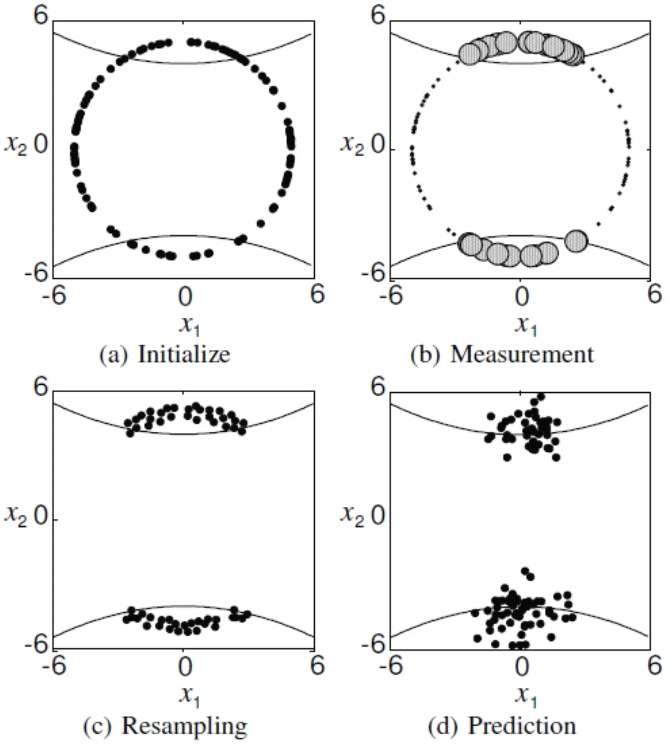


Figure 3 Illustration: particle filter

[GBD12] is the latest method to perform Importance Sampling. Similar to other Importance

Sampling methods, this method has three stages. It first chooses an initial distribution to serve as a starting point for finding an approximation of $g_{opt}(\xi)$ with uniform sampling and L-2 norm minimization. Then, by minimizing the Kullback-Leibler distance using probability collectives, a closed form representation is found to search for the proper distribution. The final step is to perform conventional Importance Sampling based on the distribution obtained in the previous step and evaluate the probability for a rare event. It is reported to have 5200x speed up over traditional Monte Carlo simulation without loss of accuracy and is the fastest method among all the existing work.

Table 1 is a performance comparison among Monte Carlo, Mixture Importance Sampling, Spherical Sampling and PCIS in evaluating the SRAM Static Noise Margin in [GBD12]. We see that PCIS evaluation only needs 231 samples to achieve 90% accuracy with 90% confidence whereas Monte Carlo needs over 1 million samples to reach 88% accuracy. At the same time, PCIS has the fastest speed with the same accuracy and confidence level compared to the other two methods whose sample numbers are approximately 30000 and 10000 respectively.

|  | Monte Carlo | Mixture Importance Sampling | Spherical Sampling | Probability Collectives based Importance Sampling |
|---|---|---|---|---|
| Accuracy | 88% | 90% | 90% | 90% |
| # of samples | 1.2E+6 | 2.85E+4 | 9.77E+3 | 231 |
| Speedup | 1x | 42x | 123x | 5200x |

Table 1 Performance comparison between Monte Carlo, Mixture Importance Sampling, Spherical Sampling and Probability Collectives based Importance Sampling

# CHAPTER 4

# PROPOSED ALGORITHM

In this section, we will introduce the proposed method. This method aims to improve [GBD12] to estimate SRAM failure probability by increasing the convergence rate without loss of any accuracy. Here we assume the failure region is a single continuous region.

## 4.1 Motivation

The key problem in the framework of importance sampling is to shift the original, given distribution towards the failure region in order to build a new distribution. In particular, all of different methods try to search for the approximate distribution of optimal distribution $g_{opt}(\xi)$. We want to shift the mean value without changing the type of the original distribution such that the new distribution is tilted toward the failure region where the rare event is more likely to occur. Moreover, we want to reduce the standard deviation of the distribution so that the samples are more concentrated in the failure region, and thus the rare event in the original distribution will have a higher probability of occurring.

The above implies that samples drawn from the new distribution for importance sampling are more likely to fail, and therefore the convergence rate for the failure estimation will be accelerated.

Moreover, since the closed form function in [GBD12] is linear in both the numerator and denominator, we want to know if it is possible to make an assumption that we can add a high order representation in order to better approximate the optimal distribution.

## 4.2 Distribution Initialization

### 4.2.1 Parameter Selection

The first step of all sampling simulation problems is to decide how many parameters should be considered to model the process variations as the input parameters. In circuit modeling for SRAM cell, variation parameters for a CMOS transistor are mainly threshold voltage $V_{th}$ and effective channel length $L_{eff}$. Among all parameters, $V_{th}$ is the most significant and the effects of all other parameters are negligible.

Based on the discussion above, we will choose $V_{th}$ as the random variable to perform simulation. Note that in practice, $V_{th}$ variations are typically modeled by Gaussian random variables which will benefit mathematical analysis since this distribution type has a closed-form representation and benefits the solution of the optimal distribution problem.

### 4.2.2 Initial Distribution Selection

After the variable parameters have been fixed, the next step is to find an initial distribution to iteratively approximate the optimal distribution for importance sampling. The heuristic goal of searching for the initial distribution or even the optimal distribution is to pick more failed samples for SRAM cells which can speed up the convergence of failure probability estimation. Here, we propose a method that will quickly shift the distribution to a starting point that will allow a failure event to be more likely to happen. The first step is to sample a few uniformly distributed samples (about 100-1000) generated from the Quasi Monte Carlo method so that the whole sample space is uniformly covered. The next step is to run transistor level circuit simulation to screen out failed samples under the given performance constraints.

Let $\xi$ be the independent variable for threshold voltage $V_{th}$ with a Gaussian distribution of mean $\mu$ and standard deviation $\sigma$. The mean position of the initial distribution is based on all the failed samples while the initial standard deviation does not change. To get the mean value $\mu$, two methods are proposed. The first method is to choose one of the failed samples with minimum L-2 norm and use its threshold value as the initial mean value. The second method is to find the minimum volume ellipsoid of the set of failure points and use its centroid as the initial mean value.

The ellipsoid can be represented by [BV]

$$\begin{aligned} &\varepsilon = \{v \,|\, \|Av + b\|_2 \leq 1\} \\ &where \quad A \in S^n \quad and \quad b \in R^n \end{aligned} \tag{10}$$

And problem formulation for the case above is

$$\begin{aligned} &\min \quad \log\det A^{-1} \\ &s.t. \quad \sup_{v\in C} \|Av + b\|_2 \leq 1 \end{aligned} \tag{11}$$

Where C is the set of failed points and v represents every single failed point. This formulation can be efficiently solved using convex programming. After solving for A and b, the centroid of the ellipsoid is $A^{-1}b$ which is used as a starting point.

We want to emphasize that neither the results of these methods are regarded as the optimal distribution, which can only serve as first-cut location of sampling distribution. In fact, a closed-form formulation starting from these results is needed to obtain the optimal solution.

## 4.3 Optimal Distribution Evaluation

The basic idea is to reduce the Figure of Merit (FOM) of probability estimation with much faster speed, which means the convergence rate is increased. Different from previous works, we tried to directly minimize the calculated FOM value for each iteration. Since the FOM for importance sampling is represented as

$$fom = \frac{\sqrt{\sigma^2}}{P_f} = \frac{\sqrt{\frac{1}{N^2}(\sum_{i=1}^{N} \omega(\xi)^2 I(\xi)^2 - NP_f^2)}}{P_f} \tag{12}$$

it is obvious that the minimal value of FOM is 0 without any numerical constraint which will give us an intuitive thought: for every iteration we could try to make the numerator to be 0 such that the simulation would have a faster convergence speed. Then the problem will become

$$\min \quad \frac{1}{N^2}(\sum_{i=1}^{N} \omega(\xi)^2 I(\xi)^2 - NP_f^2) \tag{13}$$

This representation for minimizing variance is not a convex programming in general, since the second derivative of variance representation might not always be nonnegative. In fact, a derivable function is convex if and only if its second derivative is nonnegative [BV]. However, from the definition of the variance this value will be nonnegative which means the minimum value for this problem is 0. In other words, the minimum value will still take place at extreme points which can be obtained by taking derivatives.

Therefore, we find the derivative of the numerator in two parts, the first part is to find the derivative of the first part in (13) and the second part is to find the derivative of the latter part in (13) whose value is actually the square of failure probability.

For example, to find the mean value, we first find the derivative of the first part of the numerator as shown below:

$$\mu_1 = \frac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi,\mu,\sigma)^2 \times \xi}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi,\mu,\sigma)^2} \tag{14}$$

By taking the derivative of the latter part in (13), we would have

$$\mu_2 = \frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi,\mu,\sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi,\mu,\sigma)} \tag{15}$$

Note that this representation is also the closed form formula for PCIS. In details, PCIS minimizes the Kullback-Leibler distance to find a distribution that could approximate the optimal. This formula shows that minimizing the failure probability is equivalent to finding the minimum Kulback-Leibler distance.

Based on the goal for minimizing figure of merit, we would have

$$\begin{aligned} &\min \quad \textit{fom with mean} \quad (a\mu_1, b\mu_2) \\ &s.t. \quad a+b=1 \end{aligned} \tag{16}$$

where $a, b$ are coefficient for mean value

Since this formulation is not a convex programming representation, it is not easy to solve it directly to obtain a closed form solution for the optimal distribution. Here, a heuristic method will be used to solve this issue.

Since we know the minimum value for the FOM is 0 in each iteration, we can set the numerator of the FOM formula to 0. Therefore, in every iteration we will calculate variables $a_{\mu}^{t}$ and $b_{\mu}^{t}$ as follows,

$$a_{\mu} = \frac{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}}{\dfrac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2 \times \xi}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2} + \dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}} \tag{17}$$

and thus

$$b_{\mu} = 1 - \frac{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}}{\dfrac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2 \times \xi}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2} + \dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}} \tag{18}$$

During each iteration, the closed form function for shifting the mean will be

$$\mu^{t} = a_{\mu}^{t} \mu_{1}{}^{t} + b_{\mu}^{t} \mu_{2}{}^{t} \tag{19}$$

The calculation for standard deviation is similar to the procedure above. The standard deviation after taking derivatives will be

$$\sigma_{1} = \sqrt{\frac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2 \times (\xi - \mu_{1})^2}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2}} \tag{20}$$

and

$$\sigma_2 = \sqrt{\frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times (\xi - \mu_2)^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}} \tag{21}$$

The coefficient for the standard deviation is

$$a_\sigma = \frac{\sqrt{\frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times (\xi - \mu_2)^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}}}{\sqrt{\frac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2 \times (\xi - \mu_1)^2}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2}} + \sqrt{\frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times (\xi - \mu_2)^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}}} \tag{22}$$

And

$$b_\sigma = 1 - \frac{\sqrt{\frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times (\xi - \mu_2)^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}}}{\sqrt{\frac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2 \times (\xi - \mu_1)^2}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma)^2}} + \sqrt{\frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma) \times (\xi - \mu_2)^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma)}}} \tag{23}$$

During each iteration, the closed form function for shifting the mean is

$$\sigma^t = a_\sigma^t \sigma_1{}^t + b_\sigma^t \sigma_2{}^t \tag{24}$$

The optimal distribution for estimating the SRAM rare failure event in importance sampling is obtained after the above method has been converged.

Note that it is obvious from the algorithm that calculating the standard deviation $\sigma^t$ and its coefficient $a_\sigma^t$ and $b_\sigma^t$ in every iteration is highly dependent on the calculated value of the mean $\mu^t$ in every iteration. In other words, error between calculated mean and mean in optimal distribution $g_{opt}(\xi)$ will be propagated to the calculation of $\sigma^t$ and might lead to the algorithm not converging or reaching a local optima that is far away from the optimal distribution. In order to let the algorithm converge and be close to the optimal distribution, a preferred way is to let the initial starting point be located close to the failure region.

## 4.4 Failure Probability Estimation

After the convergence of the previous step, the best approximation for optimal distribution $g_{opt}(\xi)$ denoted as $h(\xi,\hat{\mu},\hat{\sigma})$ is obtained. The weighting function in this case is

$$\omega = \frac{g_{opt}(\xi)}{h(\xi,\hat{\mu},\hat{\sigma})} \tag{25}$$

A conventional importance sampling method is implemented here which will sample based on the new distribution $h(\xi,\hat{\mu},\hat{\sigma})$ .The failure probability of the SRAM cell is calculated by (4) and its figure of merit is obtained using (8). When the FOM reaches the expected value, the estimation of failure probability can also be obtained.

## 4.5 Overall Algorithm Flow

The proposed algorithm for SRAM failure analysis is described in Algorithm 1 and is based on all the above techniques. It still has three main steps like PCIS estimation.

1) Distribution Initialization: This step initializes the sampling distribution and uses it as the initial point for searching for the optimal sampling distribution. This initialization will find the minimum L2-norm distance or the centroid of minimum ellipsoid of the current detected failure region to find an initial starting point and shift the distribution to the failure region where SRAM failure will most likely to happen.

2) Optimal Distribution Evaluation: This step starts from the distribution obtained from the previous step and then uses the heuristic algorithm to allow the calculated FOM to approach 0 in every iteration. This solves the optimization problem and allows convergence to the distribution $h(\xi,\hat{\mu},\hat{\sigma})$ which approximates the optimal distribution $g_{opt}(\xi)$ for importance sampling in the next step. Coefficient for each part in (24) will be also calculated iteration by iteration.

3) Failure Probability Estimation: This step performs the conventional importance sampling method based on the approximate optimal distribution $h(\xi,\hat{\mu},\hat{\sigma})$ obtained in step 2. Failure rate of SRAM cells is estimated by converting the optimal distribution back to the original

distribution shown in (4).

Algorithm 1 Overall Algorithm for SRAM Failure Analysis

```
Input: random variables with given Gaussian distribution and sample counts
Output: the estimation of failure probability

/*Step 1: Initial Distribution Selection*/
Draw uniformly distributed points generated from Quasi Monte Carlo method.
Run transistor level simulation on these samples.
Screen out failed samples with given constraints and either calculate minimum L-2norm or minimum volume ellipsoid and then use the value of this sample or centroid of the ellipsoid as the initial mean value.
Set the initial standard deviation to be the same as the given standard deviation.
/*Step 2: Optimal Distribution Calculation*/
Draw samples from the initial distribution and then set the iteration index t=2.
Repeat
    Evaluate Indicator function with these samples
    For i = 1 to M do
        Solve for mean and standard deviation for each part with
```

$$\mu_{1,i}^{t} = \frac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t-1}, \sigma)^2 \times \xi}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t-1}, \sigma)^2}$$

$$\mu_{2,i}^{t} = \frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t-1}, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t-1}, \sigma)}$$

$$\sigma_{1,i}^{t} = \sqrt{\frac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma_{1,i}^{t-1})^2 \times (\xi - \mu_1)^2}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu, \sigma_{1,i}^{t-1})^2}}$$

and

$$\sigma_{2,i}^{t} = \sqrt{\frac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma_{2,i}^{t-1}) \times (\xi - \mu_1)^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu, \sigma_{2,i}^{t-1})}}$$

12. Calculate the coefficient for each part of mean and standard deviation with

$$a_{\mu} = \frac{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma)}}{\dfrac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t}, \sigma)^2 \times \xi}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t}, \sigma)^2} + \dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma)}}$$

$$b_{\mu} = 1 - \frac{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma)}}{\dfrac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{2,i}^{t}, \sigma)^2 \times \xi}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{2,i}^{t}, \sigma)^2} + \dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{1,i}^{t}, \sigma) \times \xi}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{1,i}^{t}, \sigma)}}$$

$$a_{\sigma}^{t} = \frac{\sqrt{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t}) \times (\xi - \mu_{2,i}{}^{t})^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t})}}}{\sqrt{\dfrac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t}, \sigma_{1,i}^{t})^2 \times (\xi - \mu_{1,i}{}^{t})^2}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t}, \sigma_{1,i}^{t})^2}} + \sqrt{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t}) \times (\xi - \mu_{2,i}^{t})^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t})}}}$$

and

$$b_{\sigma}^{t} = 1 - \frac{\sqrt{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t}) \times (\xi - \mu_{2,i}^{t})^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t})}}}{\sqrt{\dfrac{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t}, \sigma_{1,i}^{t})^2 \times (\xi - \mu_{1,i}^{t})^2}{\sum_{i=1}^{N} I(\xi)^2 \times \omega(\xi, \mu_{1,i}^{t}, \sigma_{1,i}^{t})^2}} + \sqrt{\dfrac{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t}) \times (\xi - \mu_{2,i}^{t})^2}{\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \mu_{2,i}^{t}, \sigma_{2,i}^{t})}}}$$

13. Obtain mean and standard deviation

$$\mu^{t} = a_{\mu}^{t} \mu_1{}^{t} + b_{\mu}^{t} \mu_2{}^{t} \text{ and } \sigma^{t} = a_{\sigma}^{t} \sigma_1{}^{t} + b_{\sigma}^{t} \sigma_2{}^{t}.$$

14. End for.

15. Draw samples from updated distribution and set t=t+1.

```
Until converged.
Obtain optimal distribution parameters.
/*Step 3 Failure Probability Estimation*/
Draw samples from obtained optimal distribution.
Run transistor level simulation on these samples and evaluate the indicator function.
Calculate weight function.
    $w(\xi, \hat{\mu}, \hat{\sigma}) = \frac{\prod_{i=1}^{M} h(\xi_i)}{\prod_{i=1}^{M} h(\xi_i, \hat{\mu}, \hat{\sigma})}$
Solve the SRAM failure probability with conventional importance sampling equation
    $P = \frac{1}{N}\sum_{i=1}^{N} I(\xi) \times \omega(\xi, \hat{\mu}, \hat{\sigma})$
```

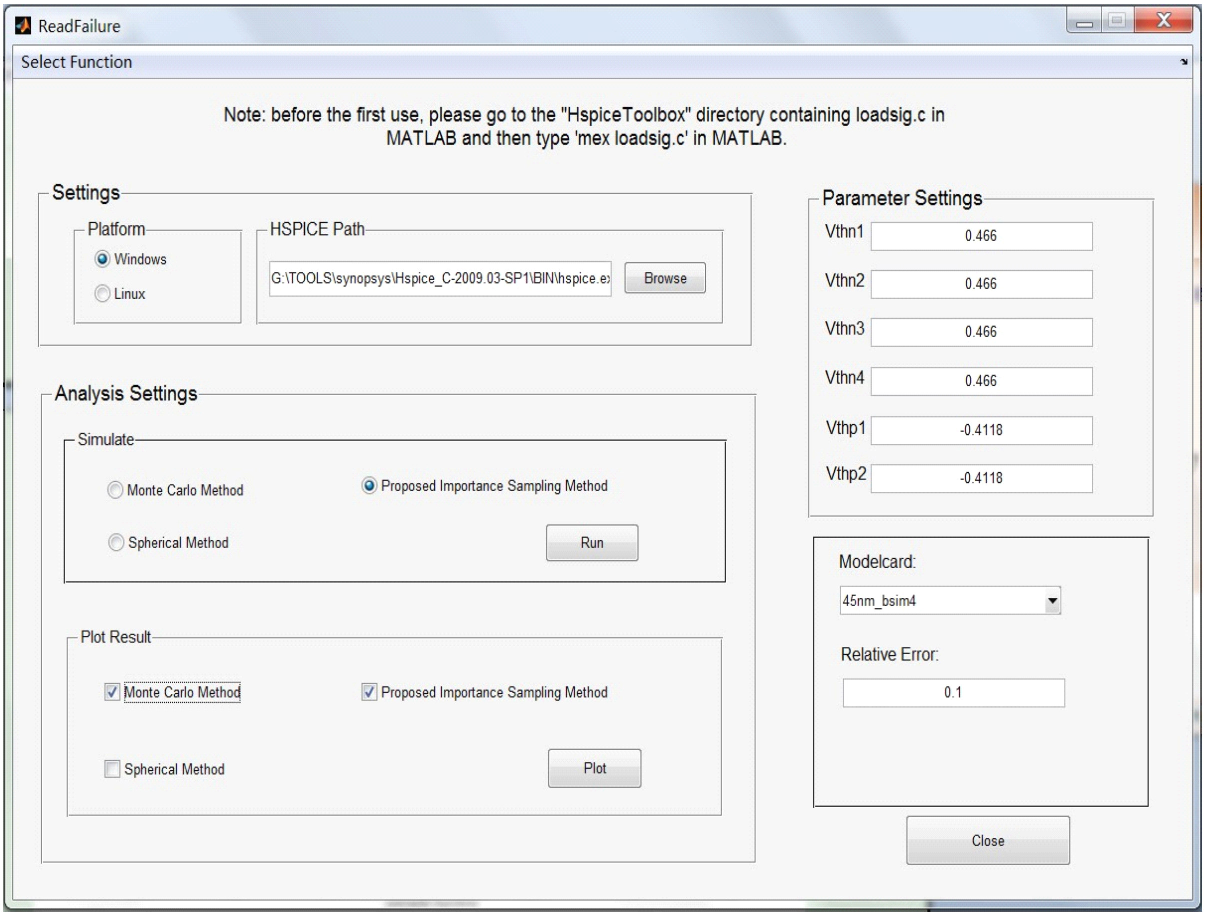
As shown in Chapter 5, the proposed method offers a 5x average speedup over PCIS without losing any accuracy.

# CHAPTER 5

# EXPERMIMENTAL RESULTS

The proposed algorithm for SRAM reading failure has been implemented in MATLAB interfaced with HSPICE and BSIM4 model. A Graphical User Interface that can choose the simulation method and set parameters is shown in figure 4. As it was mentioned before, SRAM reading failure estimation involves variations of the threshold voltages of all six MOSFETs. These variations have the most significant effect on the circuit compared to other parameters, thus other parameters may be ignored. Moreover, if other variation sources or other rare event estimation cases are selected, the proposed algorithm can still be applied by substituting the variation sources and deterministic calculation method after sampling.

We compared three different methods (i.e., the proposed method, PCIS and Monte Carlo simulation) to validate the accuracy of the proposed algorithm. We report the results of simulation for a figure of merit value of $\rho(P) = 0.1$ , which has already been illustrated to guarantee 90% accuracy with 90% confidence. For the first two methods, we use the L2-norm in the first step for comparison convenience and terminate the process if the number of samples is greater than 10,000. This value is selected because any more than 10,000 samples will be unusable for importance sampling.



Note: before the first use, please go to the "HspiceToolbox" directory containing loadsig.c in MATLAB and then type 'mex loadsig.c' in MATLAB.

Settings

Platform: ● Windows ○ Linux

HSPICE Path: G:\TOOLS\synopsys\Hspice_C-2009.03-SP1\BIN\hspice.e:

Analysis Settings

Simulate: ○ Monte Carlo Method ● Proposed Importance Sampling Method ○ Spherical Method

Plot Result: ☑ Monte Carlo Method ☑ Proposed Importance Sampling Method □ Spherical Method

Parameter Settings

Vthn1: 0.466
Vthn2: 0.466
Vthn3: 0.466
Vthn4: 0.466
Vthp1: -0.4118
Vthp2: -0.4118

Modelcard: 45nm_bsim4

Relative Error: 0.1

Figure 4 User interface for failure estimation

## 5.1 Parameter Settings

Two main aspects should be set: the distribution and specific distribution settings for the input variance source, and also the threshold value for recognizing SRAM reading failure.

The original distribution of threshold voltages for each MOSFET in SRAM is given as follows.

All the distributions are assumed to be normal distribution. The normal mean value for NMOS and PMOS threshold voltage are 0.466 $V$ and -0.4118 $V$ respectively. The standard deviation for all 6 threshold voltage is 10% of the absolute mean values, 0.0466 and 0.04118 respectively. The value of $V_{dd}$ is $1.3V$. The threshold value for bit line voltage difference is 0.001 $V$.

## 5.2 Comparison of Failure Estimation

The evolution of failure rate for different methods is plotted in figure 5. The black starred line is the evolution of failure rate using Monte Carlo simulation, the blue dotted line is the evolution of failure rate using PCIS and the purple circled line is the evolution of failure rate using the proposed method.

The final failure rate estimations for all three different methods are very close. Since Monte Carlo simulation would be recognized as a ground-truth value for failure rate, it is clear that the proposed method provide very high accuracy.

Apart from that, both the results from the proposed method and PCIS are close to the final accurate answer. This means both of the methods could find a good approximation distribution of the optimal sampling distribution.

In addition, by observing the two importance sampling based methods, we see that importance sampling is highly sensitive to the sample distribution and this affects both accuracy and efficiency of the estimation. This issue is the main reason we focus on searching for the best approximation distribution $h(\xi,\hat{\mu},\hat{\sigma})$ of the optimal sampling distribution $g_{opt}(\xi)$.

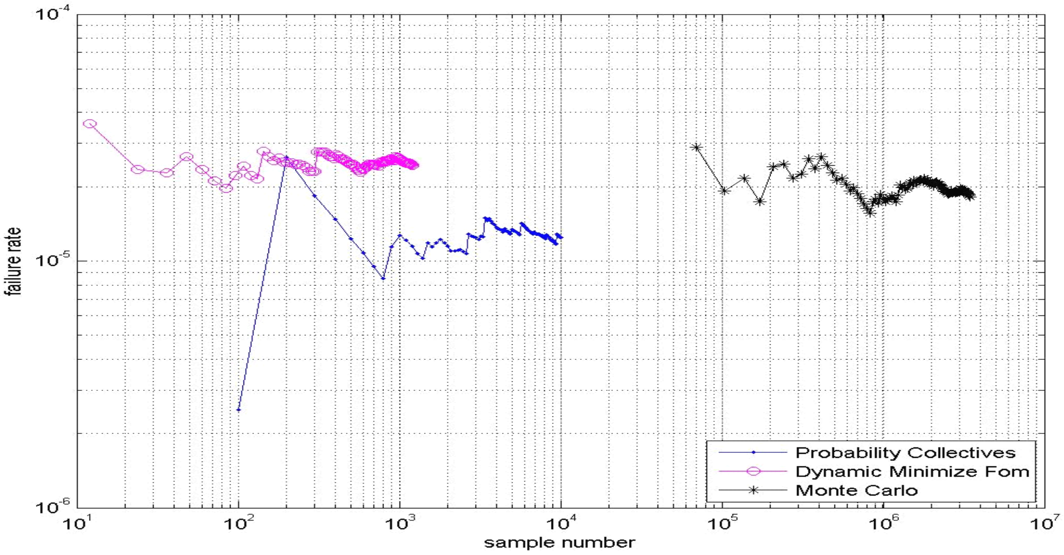


Figure 5 Comparison of evolution of failure rate among Monte Carlo, Probability Collectives based Importance Sampling and Proposed Method

## 5.3 Comparison of Figure of Merit

As mentioned above, Figure of Merit (FOM) is a standard criterion that is used to decide when to stop a simulation. Using this method, we can achieve $(1-\varepsilon)100\%$ accuracy with

$(1-\delta)100\%$ confidence. For importance sampling, FOM is given by

$$fom=\frac{\sqrt{\sigma^2}}{P_f}=\frac{\sqrt{\frac{1}{N^2}(\sum_{i=1}^{N}\omega(\hat{X}_i)^2 I(\hat{X}_i)^2-N_{P_f}^2)}}{P_f} \tag{26}$$

Where $P_f$ is the failure probability, $\omega$ is the weight function and $I$ represents the indicator function which has a value of 1 if a sample point is a failed sample, and 0 otherwise. As such, figure of merit can also be treated as a relative error.

Similar to the comparison of failure probability, the evolution of figure of merit for three different methods are plotted in figure 6. The black starred line is the evolution of figure of merit using Monte Carlo simulation, the blue dotted line is the evolution of figure of merit using PCIS and the purple circled line is the evolution of figure of merit using the proposed method. We could see that the proposed method and PCIS reach the expected figure of merit level much faster than Monte Carlo method. This is because these two importance sampling based methods can pick more samples in the failure region whereas Monte Carlo simulation performs completely random sampling across the whole region even though we want points in the failure region more than we want points outside of it.

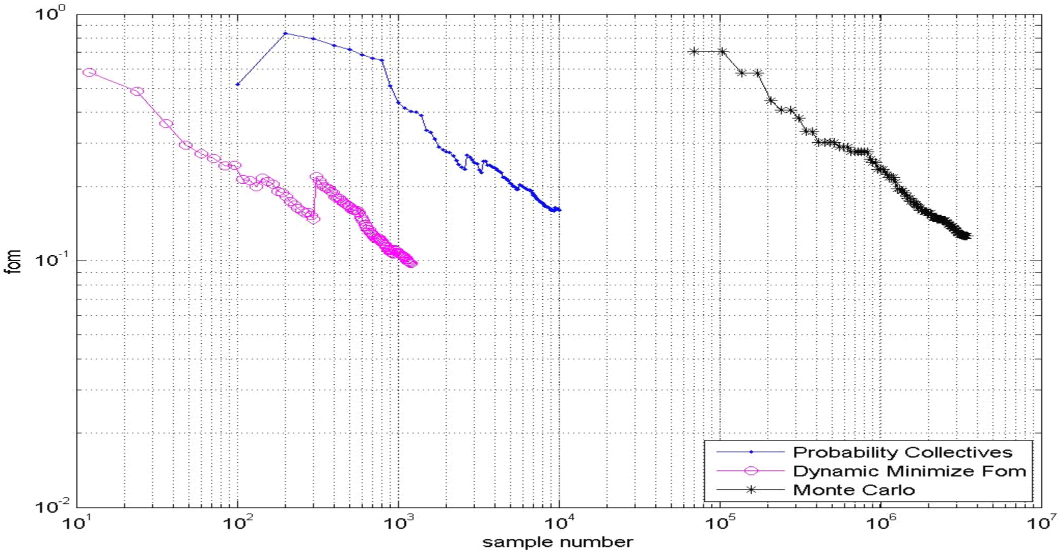


Figure 6 Comparison of evolution of Figure of Merit among Monte Carlo, Probability Collectives based Importance Sampling and Proposed Method

## 5.4 Comparison of Efficiency

### 5.4.1 Convergence Rate

Figure 6 also shows the convergence rate of the proposed algorithm. The proposed method is the fastest among all three methods. Compared to the other two methods, the proposed method samples more points in the failure region and increasingly improves accuracy to a high level.

After 3.9million samples in Monte Carlo simulation, the value of FOM still does not reach 0.1 whereas the proposed method reaches 0.1 with only 1300 samples, which translates into 3000x speedup. This shows that under the same framework, the proposed method has a significant improvement for finding an approximate distribution for the optimal distribution

### 5.4.2 Other Efficiency

Like most of the importance sampling based methods, some extra samples are needed for finding an initial distribution and searching for the approximated optimal distribution. Monte Carlo simulation does not need any extra samples. However, in our experiment, the proposed method uses 1000 samples from the uniform distribution to find the initial starting distribution for iteration, and then uses approximately 1000~2600 samples to search for the approximate optimal distribution $h(\xi, \hat{\mu}, \hat{\sigma})$ . The same issue occurs in PCIS. This method also uses 1000 samples from the uniform distribution to generate an initial starting distribution for iteration, then uses around 600~3000 samples to get the approximate optimal distribution $h(\xi, \hat{\mu}, \hat{\sigma})$ . In conclusion, compared to the millions of samples used in Monte Carlo simulation, the 2000 extra samples used to generate an initial distribution are negligible.

## 5.5 Comparison of Multiple Trials of the Algorithm

To show that the above results are repeatable, we performed simulation for the proposed method and PCIS five times. Evolution of failure rate and figure of merit for these two methods and Monte Carlo simulation are plotted in figure 7 and figure 8. A few observations are made from the two figures.

First, we see that the failure probabilities from the proposed method in all trials are nearly the same. However, PCIS are not consistent. This shows that the approximate optimal distribution we generate is closer than the one generated by PCIS.

Second, the proposed method converges with less than 4000 samples in all simulations while only 2 simulations in PCIS converges within 10000 samples. This is because more samples would be chosen from the failure region in the proposed method than PCIS so that figure of merit will converge faster. On the other hand, in step 2 of the algorithm, the proposed method would incline to converge to the distribution which is closer to the optimal distribution. On the contrary, PCIS could converge to a distribution which might not be as close as the proposed method.

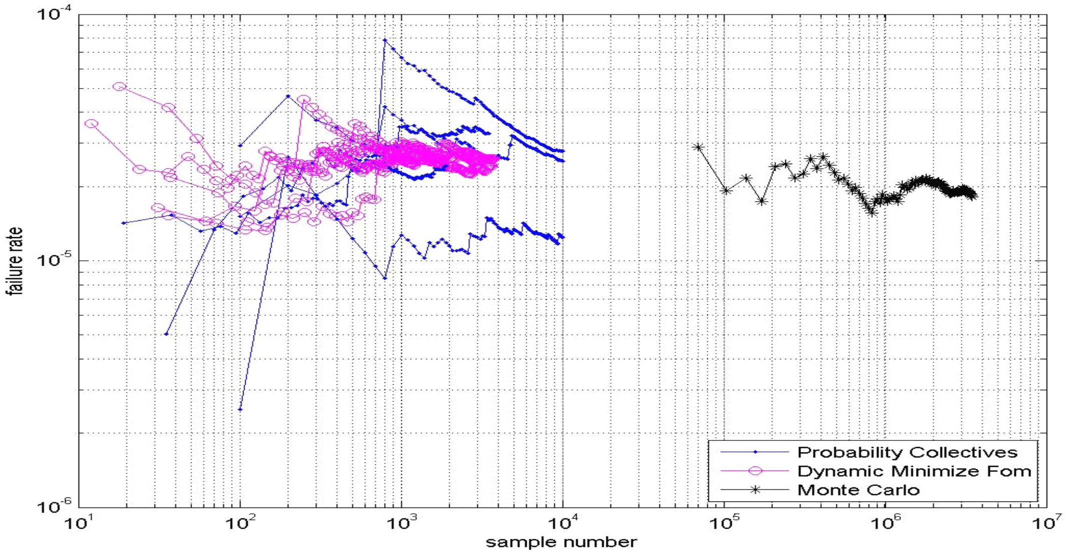


Figure 7 Multiple trials comparison of evolution of failure rate among Monte Carlo, Probability Collectives based Importance Sampling and Proposed Method

It needs more samples for the same accuracy. Judging from the two figures, the proposed method is more robust than PCIS and has an average 5x speed up over PCIS.

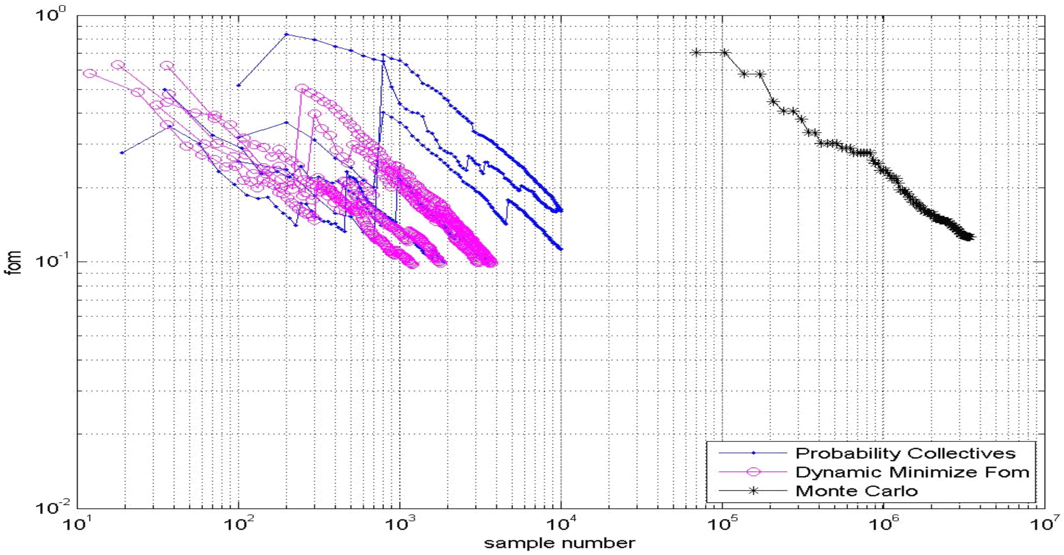


Figure 8 Multiple trials comparison of evolution of Figure of Merit among Monte Carlo, Probability Collectives based Importance Sampling and Proposed Method

Figure 9 and figure 10 show the failure rate evolution and figure of merit evolution and failure rate evolution for 10 trials of the proposed method. All 10 trials converge to the same failure rate and less than 4000 samples are needed among all these trails. This shows that in most cases the proposed algorithm converges to an accurately approximated distribution which is close to the actual optimal distribution and demonstrates its stability.

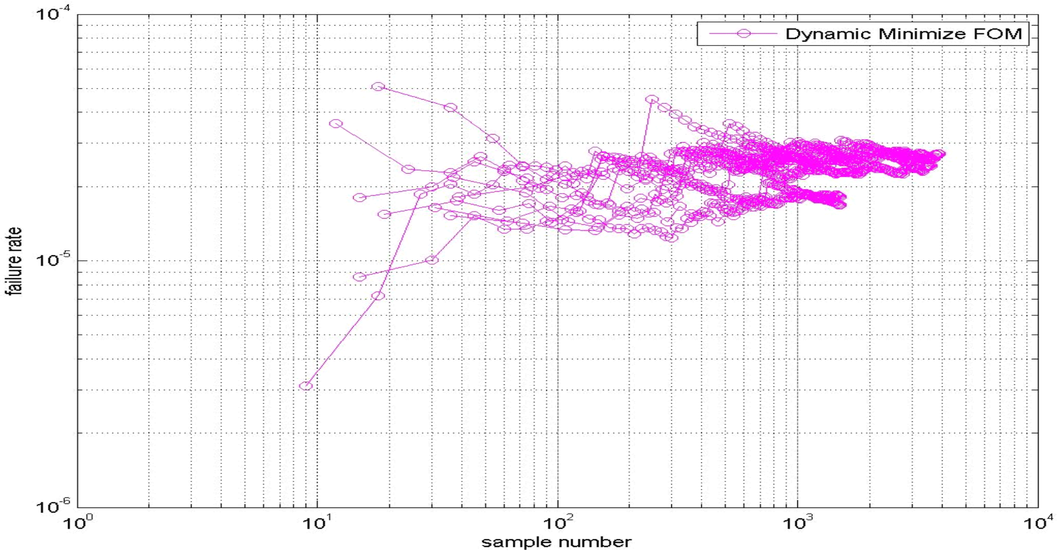


Figure 9 Ten trials of evolution of failure rate in Proposed Method

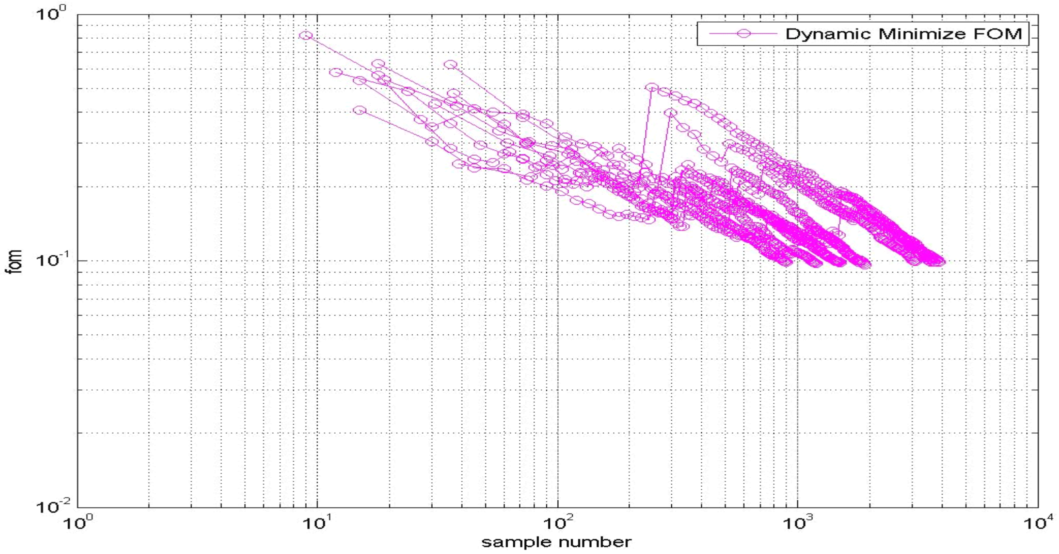


Figure 10 Ten trials of evolution of figure of merit in Proposed Method

## 5.6 Discussion on Coefficient of Function

Table 2 shows the coefficient in (19) for 5 iterations. We see that the coefficient $a_\mu$ is about 0.03 and coefficient $b_\mu$ (the coefficient for PCIS) is about 0.97. Since the closed form function in probability based importance sampling is linear in either numerator or denominator, we want to validate whether a high order representation with proper weighting process can better approximate the optimal distribution. From the table we see that the two coefficient values are almost a fixed number. We also see a dominant value of coefficient $b_\mu$ which is exactly the representation of PCIS. This demonstrates that proposed method experimentally proves that [GBD12] is a correct and practical approach for finding the approximation of optimal distribution $g_{opt}(\xi)$.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $a_\mu$ | 0.03184 | 0.03228 | 0.03156 | 0.03077 | 0.03421 |
| $b_\mu$ | 0.96815 | 0.96772 | 0.96843 | 0.96923 | 0.96579 |

Table 2 Coefficient of closed form function during five trails of the Proposed Method

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

[GBD12] have developed an excellent importance sampling algorithm based on probability collectives method. This method aims to minimize the Kullback-Leibler distance so as to find a distribution that could approximate the optimal distribution for importance sampling. As such, the method in [GBD12] can provide significant complexity reduction while retaining the accuracy. In this thesis, we improved this method by plugging in higher order components in closed form representation (19) and (24) to better approximate the optimal distribution. We minimize the calculated FOM value to better approximate the optimal distributions. Experiments show that the proposed method provides high accuracy and accelerates the convergence rate. More importantly, multiple trials of the proposed method show that the proposed method has better stability than other existing methods. For example, all 10 trials of the proposed method in the experiments need less than 4000 samples whereas 3 out of 5 trials of the PCIS need more than 10,000 samples to reach 90% accuracy.

Future work may include the following three topics. First, since the actual optimal distribution cannot be easily obtained, we would like to know if a larger number of high order components can be added to fit the optimal distribution more precisely. Second, since both the proposed method and PCIS were tested at low dimensions (6 threshold voltages as variation source), they should be validated in higher dimensions. Finally, since both the proposed method and PCIS are based on the assumption that the failure region is a single continuous region, some modifications should be considered to handle multiple failure regions.

# REFRENCES

[AN06] K. Agarwal and S. Nassif. "Statistical analysis of SRAM cell stability," Proceedings of the 43rd annual Design Automation Conference, DAC ’06, pp. 57–62, 2006.

[BV] S. Boyd and L. Vandenberghe."Convex Optimization," Camnridge University Press, 2004.

[CT91] T. M. Cover and J. A. Thomas, "Elements of information theory," John Wiley and Sons, 1991.

[DL11] C. Dong and X. Li, "Efficient SRAM failure ratepredictionvia Gibbs sampling," in Proceedings of the 43rd annual Design Automation Conference, DAC’11, 2011.

[DQS08] 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.

[IS] "Importance sampling." http://en.wikipedia.org/wiki/Importance_sampling Date Retrieved Nov 1, 2012.

[GBD09] 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.

[GBD12] Fang Gong, Sina Basir-Kazeruni, Lara Dolecek, Lei He, "A fast estimation of SRAM failure rate using probability collectives," 2012 International Symposium on Physical Design 2012.

[HW04] 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.

[KHT10] 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,

[KJN06] 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.

[MCM] "Monte carlo method." http://en.wikipedia.org/wiki/Monte_Carlo_method Date Retrieved Nov 1, 2012.

[RE00] V. K. Rohatgi and A. K. M. Ehsanes Saleh, "An introduction to probability and statistics" Wiley-Interscience, 2000.

[SR07] 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.

[SRA] "Static random-access memory." http://en.wikipedia.org/wiki/SRAM Date Retrieved Nov 1, 2012.

[YAL09] S.Yaldiz, U. Arslan, X. Li and L. Pileggi,"Efficient statistical analysis of read timing failures in SRAM circuits" in Proceeding of ISQED '09 Proceedings of the 2009 10th International Symposium on Quality of Electronic Design, pp 617-21, 2009.