Title
Stochastic Modeling and Analysis of Custom Integrated Circuits

Author
Gong, Fang

Publication Date
2012

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Stochastic Modeling and Analysis of Custom Integrated Circuits

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Fang Gong

2012

Abstract of the Dissertation

Stochastic Modeling and Analysis of Custom Integrated Circuits

by

Fang Gong

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2012

Professor Lei He, Chair

In the past few decades, the semiconductor industry kept shrinking the feature size of CMOS transistors with great efforts in order to pack more functional devices onto a smaller footprint, which follows the famous Moore's law. However, it becomes extremely difficult to ensure the correct functionalities of fabricated circuits in today's integrated circuit (IC) technology, because the increasing variations from the manufacturing have introduced inevitable and significant uncertainties in circuit performance. Moreover, the requirements of lower power consumption and higher operating frequency for today's mobile devices demand tighter performance constraints on fabricated circuits. Therefore, reliable and efficient statistical analysis methodologies are highly sought to enable IC designers to predict the stochastic behavior in fabricated circuits under random process variations before entering expensive manufacturing.

In this research, the impacts of process variations are studied in the contexts of failure analysis of memory circuits, stochastic behavioral modeling and variational capacitance extraction and novel solutions to these contexts are presented. In particular, memory circuits require an extremely small failure probability for one single cell due to their high replication count on a small footprint, thereby making it a great challenging task to provide accurate estimations. To this end,

an improved importance sampling algorithm is proposed to significantly expedite the convergence rate of failure probability estimation for memory circuits without compromising accuracy. For high dimensional problems, the conventional importance sampling schemes tend to lose accuracy and become very unreliable. To fix this issue, a novel and fast statistical analysis is presented to estimate the extremely small failure probability of memory circuits in high dimensions. In addition, an efficient statistical analysis is proposed to explore the stochastic behavior of circuit designs due to random process variations. This methodology enables IC designers to accurately predict the "arbitrary" probabilistic distribution of circuit performance considering the uncertainties from the manufacturing. Lastly, parasitic capacitance has more impact on circuit performance in today's sub-micron CMOS technology, which leads to unpredictable delay variations and severe timing errors. To address this issue, a novel and fast capacitance extraction algorithm is proposed to model the geometric variations of interconnect circuits and accurately calculate the variational parasitic capacitance. These stochastic modeling and analysis methodologies can be used to analyze custom circuits under process variations in the present nano-technology era and future generations of IC technology.

The dissertation of Fang Gong is approved.

Milos D. Ercegovac

Puneet Gupta

Alan Laub

Lei He, Committee Chair

University of California, Los Angeles

2012

To My Lovely Family and Friends.

Table of Contents

List of Figures

List of Tables

Vita

2001–2005 B.S., Department of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing, China.

2005–2008 M.S., Department of Computer Science, Tsinghua University, Beijing, China.

2008–present Ph.D. program, Department of Electrical Engineering, University of California, Los Angeles, California, USA.

- Teaching Assistant, Modeling of VLSI Circuits and Systems, Winter 2011, Winter 2012.
- Teaching Assistant, Circuit Measurement Laboratory, Winter 2010, Spring 2010.
- Graduate Student Researcher, UCLA Design Automation Lab, 2008–present.

Summer 2012 Research Intern, IBM Corp., Systems and Technology Group, Hopewell Junction, NY, USA. Worked on Timing and Noise analysis tool for VLSI chip design.

Publications

F. Gong, S. Basir-Kazeruni, H. Yu and L. He, "Stochastic Behavioral Modeling Analysis of Analog/Mixed-Signal Circuits", *IEEE Transactions on COMPUTER-AIDED DESIGN of Integrated Circuits and Systems (TCAD).*(to appear)

F. Gong, S. Basir-Kazeruni, L. Dolecek and L. He, "A Fast Estimation of SRAM Failure Rate Using Probability Collectives", *ACM International Symposium on Physical Design* (ISPD'12), Napa Valley, CA, March 25-28, 2012. Page(s):41-47.

F. Gong, H. Yu, L. Wang and L. He,"A Parallel and Incremental Extraction of Variational Capacitance with Stochastic Geometric Moments", *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, vol. 20, no. 9, Page(s): 1729–1737, Sept. 2012.

F. Gong, H. Yu and L. He, "Fast Non-Monte-Carlo Transient Noise Analysis for High-Precision Analog/RF Circuits by Stochastic Orthogonal Polynomials", *in Proceedings of the 48th IEEE Design Automation Conference* (DAC'11), San Diego, CA, June5-10, 2011. Page(s):298-303.

F. Gong, H. Yu and L. He, "Stochastic Analog Circuit Behavior Modeling by Point Estimation Method", *in Proceedings of the 2011 International Symposium on Physical Design 2011* (ISPD'11), Santa Barbara, California, March 27-30, 2011. Page(s): 175-182.

F. Gong, H. Yu, Y. Shi, D. Kim, J. Ren and L. He, "QuickYield: An Efficient Global-Search Based Parametric Yield Estimation With Performance Constraints", *in Proc. ACM/IEEE 47th Design Automation Conference* (DAC'10), Anaheim, California, June 13 - 18, 2010. Page(s):392-397.

F. Gong, H. Yu and L. He, "PiCAP: A Parallel and Incremental Capacitance Extraction Considering Stochastic Process Variation", *in Proc. ACM/IEEE 46th Annual Design Automation Conference* (DAC'09), San Francisco, California, July 26 - 31, 2009. Page(s): 764-769.

CHAPTER 1

Introduction

1.1 Background

The semiconductor industry has been migrating to the nanometer regime in order to pack more functional devices into a smaller footprint in the past few decades. In particular, the feature size of CMOS transistor has been shrinking to 45nm and below which follows Moore's law [Moo75]. As such, modern mobile devices (e.g., smart phones, tablets, laptop computers and etc.) are able to provide more functions, small sizes and high performance by integrating more transistors into smaller devices. However, it has become extremely challenging to guarantee high-precision and high-reliability in modern IC designs due to inevitable uncertainties and variations.

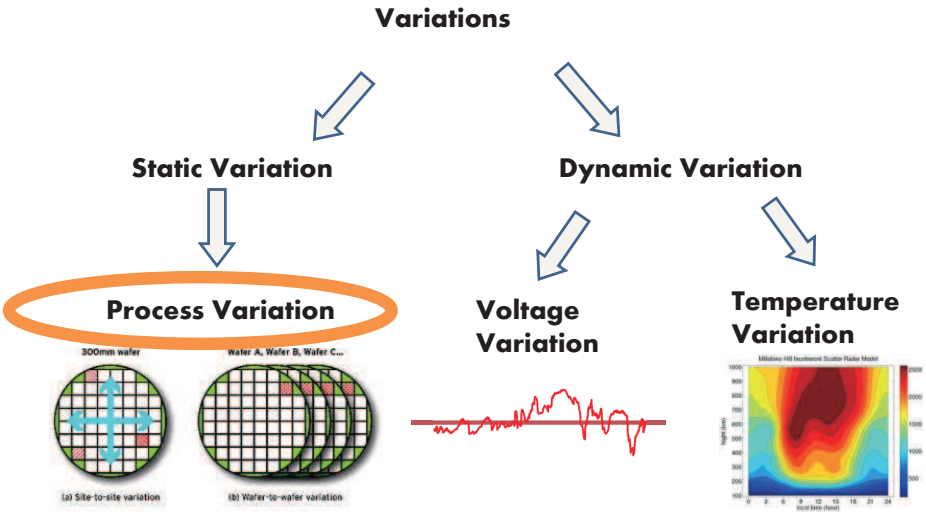


Figure 1.1: The categories of variations exist in custom integrated circuits.

In general, these variations can be categorized into two types: "static variations" denote the uncertainties during the manufacturing process, such as process variations (e.g., uncertainties of channel width, length and oxide thickness in CMOS transistors); "dynamic variations" include the uncertainties of fabricated circuits during operations that change over time, such as variations of power supply voltage and environmental temperature. Specifically, the research in this thesis focuses on process variations, which have been identified as the leading source that introduces unavoidable uncertainties in circuit behavior and leads to significant yield loss [BDM02, CCS04, EBS97].

Moreover, process variations have become larger as technology scales down to smaller feature sizes. As shown in Fig. 1.2 [Ass05], the variability of CMOS threshold voltage has increased in the past few years. Clearly, the large variability of threshold voltage can be translated into large amount of variations in circuit behavior (e.g., leakage power, timing delay, output swing, etc.), where circuit performance merits have shifted from deterministic to probabilistic and are more likely to fail the performance constraints. To compensate for the effects of process variations, stochastic analysis tools are urgently sought to accurately characterize the random process variations and efficiently predict their effects on circuit behavior.

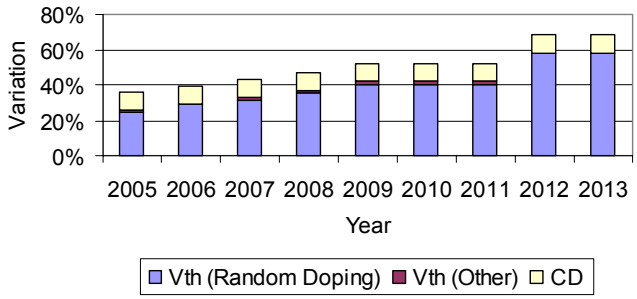


Figure 1.2: The trend of process variations on CMOS threshold voltage.

1.2 Motivations

Standard cells (e.g., SRAM bit-cell, flip-flop, I/O cells, etc.) are very important components in IC designs and need to be repeated millions of times to provide high integration density. As such, it is a must that each cell has an extremely small failure probability [HW04, BDG09] and tight layout footprint. In fact, the failure probability of an SRAM cell should be kept extremely small (e.g., 1e-5∼1e-8), making the failure event become a "rare event" [AN06]. In addition, SRAM cell design tends to adopt the most advanced process technology to achieve the minimum-sized cells and thus becomes more vulnerable to process variations. Therefore, accurate failure analysis of SRAM design considering process variations has become increasingly important and challenging to the modern VLSI industry.

To enable IC designers to predict the stochastic behavior of circuit designs due to process variations, many statistical methodologies have been developed in the past few years [Nas01,XK02,VWG06,PR90,LLG04,LL08]. Previous works usually assume small deviations on variable parameters and make use of *linearization* technique (e.g., [Nas01] models circuit performance as a first-order polynomial function of variable parameters). These methodologies fail to handle today's VLSI technology for two reasons:

- **Large Variation**: each variable parameter tends to have a probabilistic distribution with a greater standard deviation than ever before, which can be translated to be larger uncertainty in circuit behavior.
- **Strong Nonlinearity**: the behaviors of modern custom circuits typically have strongly *nonlinear* relationships with variable parameters. Therefore, it becomes extremely difficult to accurately predict the stochastic behavior of custom circuits with previous methodologies.

Consequently, novel stochastic methodologies are needed to accurately model the

circuit behavior in the presence of process variations.

Lastly, parasitic capacitance extraction considering process variations has recently regained popularity [ZW05, ZZC07, CCS08]. Due to process variations, fabricated interconnects and dielectrics show significant differences from the nominal shapes. As a result, the extracted parasitic capacitance can be off from the nominal value by a large margin, which may further lead to a significant variability for the delay calculation and timing analysis. For example, as shown in [LNP00] variations of interconnects can cause as much as 25% variations in the clock skew. Thus, it becomes an urgent need to accurately extract variational parasitic capacitance under random process variations.

1.3 Contributions

Importance sampling scheme has been investigated in past few years to estimate the failure probability of SRAM cells [KJN06,DQS08,QTD10,SR07,KHT10, DL11], because the failure of SRAM cells is a "rare event" and classic Monte Carlo (MC) method is extremely time-consuming (e.g., one million MC samples can only capture one single failure event). However, these importance sampling methodologies are plagued by slow convergence rates of probability estimation. In this thesis, an improved importance sampling algorithm is proposed to increase the convergence speed while providing high accuracy. The proposed approach has been applied to the failure analysis of SRAM cells and compared with other various methods. Extensive experiments show several orders of magnitude speedups over other existing techniques along with better accuracy.

The importance sampling schemes work sufficiently very well in low dimensional problems but become extremely inaccurate and unreliable in high dimensions. To fix this issue, a fast statistical failure analysis of memory circuits in high dimensions is proposed in this thesis, which has been successfully applied to failure

probability prediction of memory circuits with up to hundreds variables. To the best of our knowledge, this is the first work that successfully applies the importance sampling paradigm to high dimensional problems. Experimental results show the failure of existing importance sampling methods and demonstrate the validity of the proposed approach.

A novel moment-matching based algorithm is presented to extract the "arbitrary" probabilistic behavioral distributions of custom circuits. Note that "circuit behavior" denotes the performance merits (e.g., node voltage, period, bandwidth, etc.) and thus we use "circuit behavior" and "performance merits of circuit" interchangeably in this thesis. In particular, the proposed method can accurately recover the "arbitrary" probabilistic distributions of circuit performance and provide great computational complexity reduction. The proposed approach has been successfully applied to high dimensional problems with large variations and strongly nonlinear circuits. The extensive experiments demonstrate that the proposed method can provide significant speedup over Monte Carlo method while retaining the accuracy.

The existing parasitic capacitance extraction algorithms [NW91,SD97,SLK98] fail to characterize the geometric variations of interconnects and usually needs to solve a large-scale dense system with great computational efforts. To take variation effects into account and provide better efficiency, a parallel and variability-aware solver for stochastic capacitance extraction is developed in this thesis, which makes use of stochastic orthogonal polynomials to describe random geometric variations and solves the dense linear system in parallel for variational capacitance. The overall extraction flow is called *piCAP* and a number of experiments show that *piCAP* efficiently handles a large-scale on-chip capacitance extraction with variations.

1.4 Structure of Dissertation

The research presented in this dissertation mainly focuses on process variation modeling and analysis using numerical and statistical techniques, which studies four important issues: failure analysis of SRAM cells, failure analysis of memory circuits in high dimension, stochastic behavioral modeling and variational capacitance extraction.

The remainder of this dissertation is organized as follows:

- **Chapter 2: Fast Failure Probability Estimation of SRAM Cells**
  An improved importance sampling method is presented for the failure analysis of SRAM cells where circuit failure is a rare event.

- **Chapter 3: Fast Failure Analysis of Memory Circuits in High Dimensions**
  A fast statistical failure analysis of memory circuits in high dimensions is proposed in this chapter.

- **Chapter 4: Stochastic Behavioral Modeling and Analysis**
  The "arbitrary" probabilistic behavioral distributions of custom circuits are accurately recovered with affordable computational efforts using the proposed algorithm in this chapter.

- **Chapter 5: Parallel and Variability-Aware Capacitance Extraction**
  The parasitic capacitance extraction under process variations is studied and a novel solution is developed in this chapter.

- **Chapter 6: Conclusion**
  The conclusion and future works are discussed.

CHAPTER 2

Fast Failure Probability Estimation of SRAM Cells

2.1 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 one single cell could be catastrophic. Therefore, SRAM cells require extremely small failure probability [HW04, BDG09] and the failure has become a rare event [AN06] that can only be captured with millions of samples and with extremely long Monte Carlo (MC) simulations.

To avoid the expensive MC runs, importance sampling has been proposed based on the insight that only the "importance samples" of rare events can improve the estimation accuracy and further speed up the estimation convergence. This approach has been extensively used for rare event estimation problems [KJN06, DQS08, QTD10, SR07, KHT10, DL11]. However, one critical issue that affects the efficiency of importance sampling is how to build an "optimal sampling distribution" so that more "importance samples" of rare events can be chosen.

Many statistical methodologies have been developed to build the optimal sampling distribution for importance sampling and applied to failure rate estimation of SRAM cells in the past few decades [KJN06,DQS08,QTD10,SR07,KHT10,DL11]:

For example, [KJN06] approximates the optimal sampling distribution by mixing a uniform distribution, the given sampling distribution and a "shifted" distribution centering around the failure region. [DQS08, QTD10] simply shift the mean values and keep the shape of original sampling distributions, but they minimize the norm value of shift vectors to find the optimal sampling distribution. [SR07] makes use of a "classifier" to block the Monte Carlo samples that are likely to satisfy the given performance constraints and runs simulations on the remaining samples. In addition, a "particle filtering" based approach was proposed in [KHT10] which tilts more samples towards the failure region. Moreover, [DL11] was recently proposed to adapt "Gibbs Sampling" in order to draw more failed samples directly from the failure region, which demonstrated improved performance over previous works. However, all the above-mentioned approaches either require many complicated techniques and become highly difficult, if not impossible, to implement, or converge to sub-optimal sampling distributions that cannot provide high efficiency. Therefore, one efficient algorithm with less implementation efforts and improved performance is still urgently needed to accurately estimate the failure rate of SRAM cells.

In this chapter, we present a fast algorithm based on probability collectives method for failure rate estimation of SRAM cells. First, "Kullback-Leibler (KL) distance" from probability theory [RE00] and information theory [CT91] 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 can be analytically solved by minimizing the KL distance with probability collectives method using immediate sampling [RWK06, RKW07], which is as close to the optimal sampling distribution as possible. Therefore, the estimation convergence of importance sampling 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 is more than $40X$ faster than other state-of-the-art techniques (i.e. mixture importance sampling method [KJN06] and spherical sampling method [QTD10]).

Although probability collectives was initially developed in the statistics field [RWK06,RKW07], it remains unknown how to interface it to the importance sampling method for 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 unknown how to initialize the parameterized sampling distribution. Third, the minimization of 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 [DQS08, QTD10] to efficiently initialize them by shifting the given sampling distribution towards the failure region. Moreover, the immediate sampling based probability collectives method [RWK06, RKW07] can be used to analytically solve for the optimal parameterized sampling distributions for importance sampling. To the best of our knowledge, it is the first time to present the probability collectives based importance sampling method for failure probability estimation of SRAM cells.

The rest of this chapter is organized as follows. In Section 2.2, we provide necessary background on importance sampling, KL distance and probability collectives methods. Section 2.3 contains more details of the required techniques in the proposed method for SRAM failure analysis. The experiments and more discussions are provided in Section 2.4 to validate the accuracy and efficiency of proposed method. This chapter is concluded in Section 2.5.

2.2 Background

2.2.1 Importance Sampling

Let ξ_i $(i = 1, \cdots, m)$ be independent random variables with probability density function (PDF) as $p(\xi_i)$ for circuit parameters under process variations, such as the threshold voltage and effective channel length of transistors.

As such, one Monte Carlo sample $\xi^j = (\xi_1^j, \cdots, \xi_m^j)$ consists of one sample from each random variable distribution, and their joint PDF $p(\xi)$ can be expressed as follows due to independence property:

$$p(\xi) = \prod_{i=1}^{m} p(\xi_i). \tag{2.1}$$

In addition, $f(\xi)$ is the performance merit of interest, such as static noise margin of SRAM cell (shown in Fig.2.2), and typically needs to be evaluated with expensive transistor-level circuit simulation.

Without loss of generality, f_0 can be a given performance constraint so that the circuit failure $f(\xi) < f_0$ becomes unlikely to happen or a "rare event". Thereby, one indicator function $I(\xi)$ can be defined to identify pass/fail of $f(\xi)$ as:

$$I(\xi) = \begin{cases} 0 & f(\xi) \geq f_0 \ (pass) \\ 1 & f(\xi) < f_0 \ (fail) \end{cases} \tag{2.2}$$

Therefore, the probability of failure events can be estimated in (3.3) where the failed samples count and passed samples are omitted:

$$prob(fail) = \int I(\xi) \cdot p(\xi) d\xi. \tag{2.3}$$

In general, $p(\xi)$ is known as given sampling distributions for variable parame-

ters but $I(\xi)$ is unknown. In fact, the indicator function $I(\xi)$ cannot be evaluated explicitly and usually needs extremely long Monte Carlo simulations on millions samples of ξ because the failures are *rare events*.

To avoid massive Monte Carlos samples and simulations, the importance sampling has been proposed to sample from one "distorted" sampling distribution $g(\xi)$ that tilts towards the failure region where failures become more likely to happen or "less rare":

$$prob(fail) = \int I(\xi) \cdot \frac{p(\xi)}{g(\xi)} \cdot g(\xi) d\xi = \int w(\xi) \cdot I(\xi) \cdot g(\xi) d\xi. \tag{2.4}$$

where $w(\xi)$ is likelihood ratio or weight for each sample of ξ which can unbias the probability estimation from $g(\xi)$. Theoretically, the optimal sampling distribution $g^{opt}(\xi)$ [DL11], where only one sample is needed to provide the accurate estimation of failure probability, can be expressed as:

$$g^{opt}(\xi) = \frac{I(\xi) \cdot p(\xi)}{prob(fail)} \tag{2.5}$$

However, $g^{opt}(\xi)$ cannot be evaluated with (2.5) directly because $I(\xi)$ is unknown and $prob(fail)$ is the very desired failure rate. Instead, another sampling distribution $h(\xi)$ should be created to provide an approximation as close to $g^{opt}(\xi)$ as possible so that the similar estimation behavior can be expected. As a result, the Kullback-Leibler distance can be used to define the distance between $h(\xi)$ and $g^{opt}(\xi)$.

2.2.2 Kullback-Leibler Distance

The Kullback-Leibler (KL) distance was first proposed in probability theory [RE00] and information theory communities [CT91] to measure the *directional* distance from one distribution to the other. In other words, KL distance is defined between

any two distributions and measures how "close" they are.

For example, the KL distance from distribution $g^{opt}(\xi)$ in (2.5) to $h(\xi)$ can be expressed as:

$$\mathbb{D}_{KL}(g^{opt}(\xi), h(\xi)) = \mathbb{E}_{g^{opt}}\left[log(\frac{g^{opt}(\xi)}{h(\xi)})\right]. \tag{2.6}$$

Note that both distribution g^{opt} and h should be defined over the same random variable ξ. 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}.

One important question would be why KL distance shall be chosen? In fact, there exist several divergences in probability theory (e.g., KL distance, Hellinger distance, total variation distance and etc. [LV06]) which can measure the difference between two probability distributions. The reasons why KL distance is chosen in this work are following:

- First, KL distance is a "non-symmetric" or "directed" measure of the difference between two probability distributions [CT91]. Here, the "directed" measure implies that one distribution is the fixed prior reference distribution and typically represents the "best" or "true" distribution, while the other distribution is the "less good" or "approximate" distribution. For example, the KL distance of $h(\xi)$ from $g^{opt}(\xi)$ in (2.6) is a measure of the error when the distribution $h(\xi)$ is used to *approximate* the "true" distribution $g^{opt}(\xi)$. Clearly, the KL distance is very suitable for our problem where the distribution $h(\xi)$ shall be as close to the fixed optimal distribution $g^{opt}(\xi)$ as possible.

- Second, the minimization of KL distance can easily be turned into convex optimization problems when Gaussian distributions are investigated [CT91, MR02,RR07,Mel07,BKM05]. It is well known that for a convex optimization problem, a local minimum is also a global minimum, which is easy to solve numerically. In fact, an analytical solution is available for KL distance

minimization problem in this work which will be discussed in Section 2.3.3.

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

2.2.3 Probability Collectives

In general, probability collectives (PC) method is an efficient optimization framework under uncertainty [RWK06,RKW07], which can search the optimal probability distributions of variable parameters in order to optimize the objective function.

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

$$\arg\min \mathbb{E}_{g^{opt}} \left[log(\frac{g^{opt}(\xi)}{h(\xi)}) \right]. \tag{2.7}$$

Clearly, the above minimization problem is equivalent to the following problem because $g^{opt}(\xi)$ is independent of $h(\xi)$ as shown in (2.5):

$$\arg\max \mathbb{E}_h \left[I(\xi) \cdot log(h(\xi)) \right]. \tag{2.8}$$

However, it is highly prohibitive to perform exhaustive search for $h(\xi)$ since the searching space is extremely large and contains arbitrary distributions. PC method simplifies the problem by utilizing a set of parameterized sampling distributions $h(\xi, \theta)$ with extra parameters $\theta = (\theta_1, \cdots, \theta_m)$. As such, the problem becomes:

$$\theta^* = \arg\max_{\theta} \mathbb{E}_h \left[I(\xi) \cdot log(h(\xi, \theta)) \right]. \tag{2.9}$$

where θ^* is the optimal parameter of distribution $h(\xi, \theta)$ which can lead to minimum KL distance in (2.7) and thereby $h(\xi, \theta^*)$ is the optimal approximation of

$g^{opt}(\xi)$.

Note that the expectation value $\mathbb{E}_h$ in (2.9) cannot be evaluated with analytical formula and thereby sampling techniques must be used. In fact, several sampling-based PC methods have been proposed in past few years [RWK06, RKW07], such as delay sampling based PC, immediate sampling based PC, etc.

In this chapter, we adapt the immediate sampling based PC method as summarized in Algorithm (3) and interested readers are referred to [RWK06, RKW07] for other PC methods.

Algorithm 1 Immediate Sampling based PC Algorithm

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

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

6: Draw random samples from the parameterized distribution $h(\xi, \theta^{(t)})$ and set $t = t + 1$.
7: **until** Converged (e.g. $\theta^{(t)}$ does not change for several subsequent iterations)
8: The optimum parameter θ^* can be obtained.
9: Sample final $h(\xi, \theta^*)$ to get solution(s).

Since the updated distribution $h(\xi, \theta^{(t)})$ at t-th iteration will be 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 and etc.), and it remains unclear how to choose $h(\xi, \theta)$ for SRAM failure analysis.
- It is important and nontrivial to find $\theta^{(1)}$ which provides a "starting point"

or "heuristic initial solution" for the solution of (2.9) and can significantly affect the speed of convergence in Algorithm (3).

- The optimization problem in (2.9) is very difficult to solve and one 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.

2.3 Proposed Method

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

2.3.1 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 masked [BDG09].

Moreover, V_{th} variations are typically modeled as independent random variables of Gaussian distributions [KJN06, DQS08, QTD10, SR07, DL11]. As such, it is a natural choice to deploy a family of Gaussian distributions parameterized by mean (μ) and stand deviation (σ). In fact, the choice of parameterized Gaussian distributions can help us to find one closed-form solution to the optimization problem in (2.9) 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 mean $\mu_i^{(0)}$ and standard deviation $\sigma_i^{(0)}$. To build

the parameterized Gaussian distribution for ξ_i, we introduce the extra parameters $\theta_i = (\mu_i^\theta, \sigma_i^\theta)$ by shifting the mean to μ_i^θ and reducing the standard deviation to σ_i^θ, which are motivated by the following insights:

- **Mean-shift** can lean the sampling distribution towards the infeasible region where the rare failures are more likely to happen, which is similar to the finding in [DQS08] and has been extensively used by many previous works such as [DQS08, QTD10, KHT10, MR02, RR07, Mel07, BKM05].
- **σ-change** can concentrate the samples around much smaller region where rare failures can happen with higher probability.

Therefore, the samples drawn from the parameterized Gaussian distribution $h(\xi_i, \mu_i^\theta, \sigma_i^\theta)$ are more likely to fail, and can thereby expedite the convergence of failure probability estimation in the importance sampling. However, it is still unknown how to find the optimal parameters θ_i^* efficiently, which will be investigated in following sections.

2.3.2 Initial Parameter Selection

As shown in the Algorithm (3), the first step is to initialize the parameter θ, which, in fact, provides a "starting point" or "heuristic initial solution" to search for the optimal parameter θ^*. As such, the initial parameter $\theta^{(1)}$ can significantly affect the efficiency of the iterative search in Algorithm (3) 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 [DQS08], 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 following $N(\mu_i^{(0)}, \sigma_i^{(0)})$ and the proposed initial

parameter selection can be summarized as following: first, a few hundred *uniform-distributed* samples of ξ_i can be generated using Quasi Monte Carlo method in order to evenly cover the entire parameter range, such as eight-sigma range from $(\mu_i^{(0)} - 4\sigma_i^{(0)})$ to $(\mu_i^{(0)} + 4\sigma_i^{(0)})$. Then, transistor-level simulations can be run on these samples and the failed samples can be identified with given performance constraints. We can further choose one failed sample with the *minimum* L_2-norm and use its value as the initial parameter for $\mu_i^{(1)}$. In addition, the initial sigma parameter $\sigma_i^{(1)}$ can be the same as given $\sigma_i^{(0)}$.

It is worthwhile to point out that above "norm minimization" based method can only be a heuristic for obtaining an *initial* parameterized Gaussian distribution but cannot provide the *optimal* sampling distribution $h(\xi, \theta^*)$ in (2.9) by any means. As a matter of fact, the optimization problem in (2.9) should be solved for $h(\xi, \theta^*)$ and one efficient closed-form approach is highly needed that will be discussed in next section.

2.3.3 Closed-Form Optimization Solution

Before we present the closed-form solution, it should be noted that the optimization in (2.9) must be revised as (2.10) because samples are generated from the parameterized distributions $h(\xi, \theta)$ rather than those given distributions $h(\xi)$:

$$\theta^* = \arg\max_{\theta} \mathbb{E}_h[I(\xi) \cdot w(\xi, \theta) \cdot log(h(\xi, \theta))]. \tag{2.10}$$

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

$$w(\xi, \theta) = \frac{h(\xi)}{h(\xi, \theta)}. \tag{2.11}$$

For illustration purpose, let us consider following example with a little abuse

of notation:

- $\xi = (\xi_1, \cdots, \xi_m)$: independent random Gaussian variables.
- $h(\xi) = (h(\xi_1), \cdots, h(\xi_m))$: the given Gaussian sampling distributions of ξ.
- $\theta_i = (\mu_i^\theta, \sigma_i^\theta)$: the parameters used to parameterize given PDFs $h(\xi_i)$.
- $h(\xi, \theta) = (h(\xi_1, \mu_1^\theta, \sigma_1^\theta), \cdots, h(\xi_m, \mu_m^\theta, \sigma_m^\theta))$: the chosen parameterized Gaussian distributions for ξ.
- $\xi_i^1, \cdots, \xi_i^j, \cdots, \xi_i^N$: the samples of ξ_i drawn from the parameterized Gaussian distribution $h(\xi_i, \mu_i^\theta, \sigma_i^\theta)$.

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

$$w(\xi^j, \theta) = \frac{h(\xi_1^j) \times \cdots \times h(\xi_m^j)}{h(\xi_1^j, \mu_1^\theta, \sigma_1^\theta) \times \cdots \times h(\xi_m^j, \mu_m^\theta, \sigma_m^\theta)}. \tag{2.12}$$

Moreover, the expectation value $\mathbb{E}_h$ in (2.10) cannot be evaluated directly 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 θ_i becomes the sampled form as:

$$\theta_i^* = \arg\max_\theta \frac{1}{N} \sum_{j=1}^{N} \big(I(\xi^j) \times w(\xi^j, \theta) \times \log(h(\xi_i^j, \theta_i)) \big). \tag{2.13}$$

As proposed in [RWK06], the above optimization problem is a convex optimization problem that can be solved with closed-form formula, because the parameterized distribution $h(\xi, \theta)$, following Gaussian distribution, is a log-concave distribution.

Specifically, the optimal parameters $\mu_i^{\theta,*}$ and $\sigma_i^{\theta,*}$ can be analytically solved

with closed-form formulae as [RWK06, RKW07]:

$$\mu_i^{\theta,*} = \frac{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \theta) \times \xi_i^j}{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \theta)}. \tag{2.14}$$

where $\mu_i^{\theta,*}$ can be asymptotically approached by iteratively updating the θ and re-evaluate the above formula. In practice, the iterative process can converge very fast within only a few iterations. Note that [CT91, MR02, RR07, Mel07, BKM05] have found the identical analytical formula to find the optimal parameter for mean shift.

Similarly, the closed-form formula can be derived to analytically compute $\sigma_i^{\theta,*}$ as:

$$\sigma_i^{\theta,*} = \sqrt{\frac{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \theta) \times (\xi_i^j - \mu_i^{\theta,*})^2}{\sum_{i=1}^{N} I(\xi^j) \times w(\xi^j, \theta)}}. \tag{2.15}$$

It is obvious that the calculation of $\sigma_i^{\theta,*}$ depends on the optimization result $\mu_i^{\theta,*}$ from (3.8). In other words, the potential error from the optimization of $\mu_i^{\theta,*}$ can be directly propagated into the computation of $\sigma_i^{\theta,*}$ 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 initial parameter selection presented in previous section.

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

2.3.4 Overall Algorithm Flow

The proposed algorithm for SRAM failure analysis is based on above-mentioned techniques. The overall algorithm flow has been described in Algorithm(2), which

mainly consists of three stages as summarized below:

(1) **Initial parameter selection**: The first stage aims to initialize the parameterized sampling distribution $h(\xi, \theta)$ as a "heuristic initial solution" to search for the optimal parameterized sampling distribution $h(\xi, \theta^*)$, which adopts the insight of "norm minimization" from [DQS08] and shifts the given sampling distribution towards the failure region where SRAM failures are more likely to happen.

(2) **Optimal parameter finding**: This stage starts with the initial parameterized sampling distribution and analytically solves the optimization problem as (3.8) and (2.15) to achieve the optimal parameterized sampling distribution $h(\xi, \theta^*)$.

(3) **Failure probability estimation**: The traditional importance sampling method can be performed with the obtained optimal sampling distribution $h(\xi, \theta^*)$ to estimate the failure rate of SRAM cells, where both faster convergence speed and improved accuracy can be expected.

As shown in Section 2.4, 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.

2.4 Experimental Results

We have implemented our proposed algorithm using MATLAB and Hspice with BSIM4 model. Also, Monte Carlo (MC), spherical sampling (SS) [QTD10] and mixture importance sampling (MixIS) [KJN06] are all implemented. As an illustration, the threshold voltages of all MOSFETs are considered as variation sources and static noise margin (SNM) failure is studied. Note that the same algorithm

Algorithm 2 Overall Algorithm for SRAM Failure Analysis

Input: random variables $\xi = (\xi_1, \cdots, \xi_M)$ with Gaussian distributions $h(\xi)$.
Output: the estimation of failure probability p_r.

/* Stage 1: Initial Parameter Selection */
Draw uniform-distributed samples from $h(\xi)$ and simulate these samples.
Identify those failed samples with given performance constraints and calculate the L_2-norm values.
Choose the failed sample with the minimum L_2 norm and use the value of this sample as the initial $\mu^{(1)}$.
Set the initial sigma $\sigma^{(1)}$ to be the same as given $\sigma^{(0)}$.

/* Stage 2: Optimal Parameter Finding */
Draw N_2 samples from the initial parameterized distribution $h(\xi, \mu^{(1)}, \sigma^{(1)})$ and set the iteration index number $t = 2$.
repeat
  Evaluate indicator function $I(\xi^j)$ in (3.8) and (2.15) with these samples.
  for $i = 1 \rightarrow M$ **do**
    Solve for $\mu_i^{(t)}$ and $\sigma_i^{(t)}$ with (3.8) and (2.15)

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

/* Stage 3: Failure Probability Estimation */
Draw N_3 samples from the obtained optimal sampling distribution $h(\xi, \mu^*, \sigma^*)$.
Simulate the samples ξ_j and evaluate the indicator function $I(\xi^j), (j = 1, \cdots, N_3)$.
Solve for the failure probability p_r with sampled form, where $w(\xi^j, \mu^*, \sigma^*)$ is the weight/liklihood-ratio for sample ξ^j.

$$p_r = \frac{1}{N_3} \sum_{i=1}^{N_3} I(\xi^j) \times w(\xi^j, \mu^*, \sigma^*).$$

can be applied to both other variation sources (i.e. L_{eff}, T_{ox}, etc.) and other rare failures (i.e. reading/writing failures) as well.

2.4.1 SRAM Cell and Static Noise Margin

The typical design of a 6-transistor SRAM cell has been shown in Fig. 3.5 and we introduce process variations to threshold voltage V_{th} of all MOSFETs which can be modeled as *independent* random variables of Gaussian distributions. Specifically, the nominal mean values of 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.

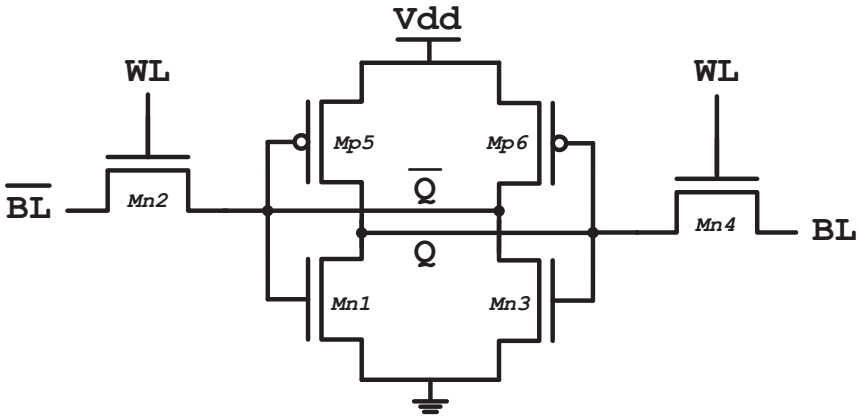


Figure 2.1: The schematic of the 6T SRAM cell.

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.

Moreover, the static noise margin (SNM) has been extensively used to measure 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 as shown in Fig. 2.2 which consist of the voltage transfer curve (VTC) of the two inverters in SRAM cell [MMR10]. As such, when SNM is less than zero, the butterfly curve is collapsed and the data

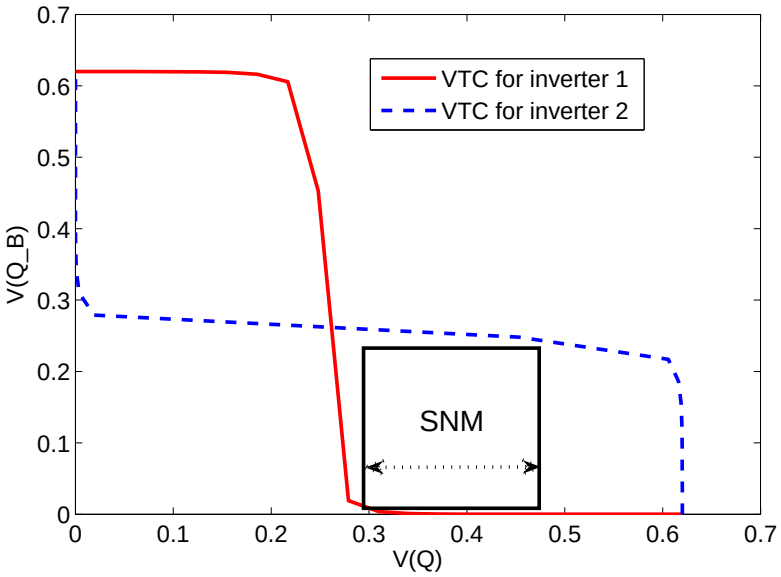


Figure 2.2: Illustration of SRAM static noise margin (SNM) butterfly curves.

retention failure happens.

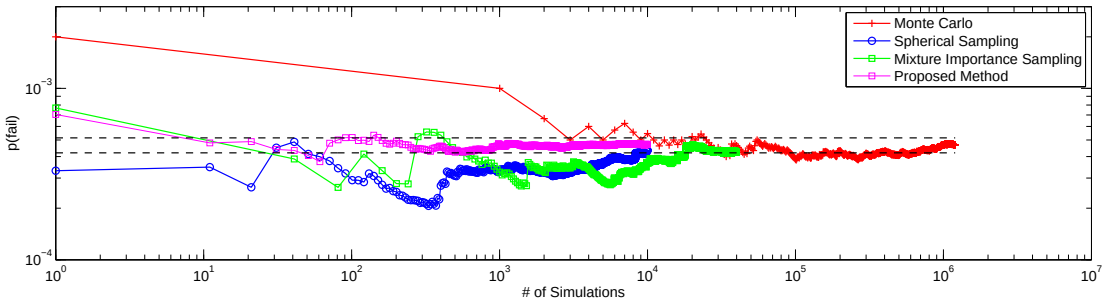
2.4.2 Accuracy Comparison

2.4.2.1 Comparison of Failure Rate Estimation

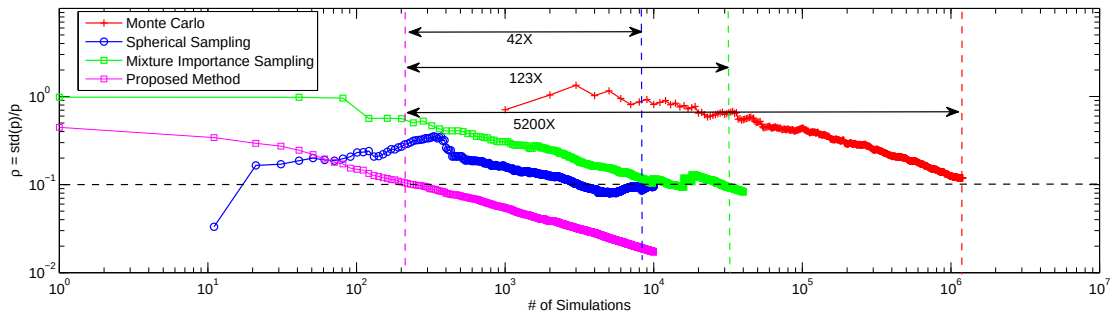
To validate the estimation accuracy of the proposed algorithm, we perform all different methods (e.g. Monte Carlo (MC), mixture importance sampling (Mix-IS) [KJN06], spherical sampling (SS) [QTD10] and proposed algorithm) on the same 6-T SRAM cell example in 45nm process to predict the probability of 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. 3.7(a), where following observations can be made:

- First, the failure rate estimations from different methods can closely match each other, which can validate the estimation accuracy of our proposed



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



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

Figure 2.3: Evolution comparison of the failure probability estimation and figure of merit for different methods.

method.

- Second, the proposed method in contrast to other methods starts with an estimation that is very close to the final accurate result, because only the proposed method can find the *optimal* sampling distribution using probability collectives method for importance sampling.

- 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 chapter to exploit the optimal sampling distribution.

2.4.2.2 Comparison of Figure-Of-Merit (FOM)

As stated in [DQS08, QTD10], Figure-Of-Merit (FOM), ρ, has been extensively used to quantify the accuracy of probability estimation, which is defined as:

$$\rho = \frac{\sqrt{\sigma^2_{prob(fail)}}}{prob(fail)}. \tag{2.16}$$

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

Similarly, we further calculate the evolution of FOM for different methods at $Vdd = 300mV$ which are plotted in Fig. 3.7(b). To clearly compare the accuracy of different methods, we plot a dashed line to indicate the 90% accuracy level with 90% confidence interval ($\rho = 0.1$) and can have two important observations as following:

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

- It is obvious that proposed method can significantly improve the accuracy in contrast to other methods when the same number of samples are available to all different methods. For instance, we compare their accuracy level in Table 2.1 with only $10,000$ samples for all different methods. 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.

Table 2.1: Results of all methods with $10,000$ samples.

	MC	MIS	SS	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

2.4.3 Efficiency Comparison

2.4.3.1 Comparison of Convergence Speed

Observing Fig. 3.7(b) again can help us to investigate the efficiency of proposed algorithm, which is shown to have the fastest speed of convergence between all the different methods illustrated. In this figure, the proposed method can choose more failed samples and increasingly improve the accuracy to an extremely high accuracy level due to the optimal sampling distribution.

The similar observation can also be found from Fig. 3.7(a): the proposed method starts with the estimation that is very close to the final accurate results and quickly converge to the 95% confidence interval of final Monte Carlo result (denoted by two dashed lines). Meanwhile, the estimations from other methods keep fluctuating and asymptotically approach 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 [KJN06] and $42X$ speedup over SS [QTD10].

Table 2.2: Accuracy and efficiency comparison for different methods.

Vdd		Monte Carlo (MC)	MixIS [KJN06]	Spherical Sampling (SS) [QTD10]	Proposed
275mV	*prob.*	1.82E-3	2.282E-3	2.678E-3	2.1E-3
	accuracy	90%	90%	90%	90%
	#runs	1.22E+05 (1X)	9.5E+4 (1.28X)	2.359E+3 (51.7X)	257 (474X)
290mV	*prob.*	7.823E-4	7.97E-4	8.163E-4	8.1E-4
	accuracy	90%	90%	90%	90%
	#runs	2.46E+5 (1X)	2.27E+4 (10.8X)	1.383E+3 (178X)	281 (875X)
300mV	*prob.*	4.675E-4	4.332E-4	4.208E-4	4.7E-4
	accuracy	88%	90%	90%	90%
	#runs	1.2E+6 (1X)	2.85E+4 (42X)	9.771E+3 (123X)	231 (5200X)

2.4.3.2 Comparison on Different VDD Levels

We also perform our proposed method to the same SRAM cell example with different VDD levels, such as $275mV$ and $290mV$, because the VDD level can significantly change the failure probability. The comparison is shown in Table 2.2 and reveals that the proposed method can provide faster convergence and improved accuracy for all VDD levels. For instance, when compared with Monte Carlo method, the proposed method can achieve $474X$ speedup for $VDD = 275mV$, $875X$ speedup for $VDD = 290mV$ and $5200X$ speedup for $VDD = 300mV$ in order to achieve the 90% accuracy. More importantly, the speedup ratio keeps increasing along with the increase of VDD value and the decrease of SRAM failure rate.

2.4.3.3 Other Efficiency Comparison

It should be noted that all importance sampling based methods require some "extra" samples to find the new sampling distribution, which are called "extra" because Monte Carlo method does not need these extra samples and 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.

Table 2.3: Comparison of total number of samples for similar accuracy.

	MC	MIS [KJN06]	SS [QTD10]	Proposed
prob. of failure	4.675E-4	4.332E-4	4.208E-4	4.7E-4
accuracy	88%	90%	90%	90%
#total samples	1.2E+6	3.15E+4	1.08E+4	2.23E+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. SS method needs 2000 samples which mainly aims to locate the failed samples with minimum L_2-norm in a spherical manner. The proposed method also needs 2000 samples to find the optimal sampling distribution. Table 2.3 shows the total number of required samples for all different methods. It is worthwhile to point out that these "extra" samples become negligible when compared with Monte Carlo method.

2.4.4 Discussion about Sigma-Change

One more question would be how the sigma-change can improve the convergence and accuracy? To answer this question, we simply compare two implementations of proposed algorithm: one only shifts the mean values of given sampling distributions similar to [DQS08, QTD10, MR02, RR07, Mel07, BKM05], and the other adopts both mean-shift and sigma-change to construct the optimal sampling distribution.

We apply these two implementations to the same SRAM cell problem at $VDD = 300mV$ and plot the evolutions of FOM in Fig. 2.4. In this figure, we similarly draw a dashed line to indicate the 90% accuracy level. So, it can be observed that proposed method with both mean-shift and sigma-change needs 231 samples, while the proposed method with only mean-shift requires 561 samples. Clearly, the sigma-change technique provides an extra $2.4X$ speedup.

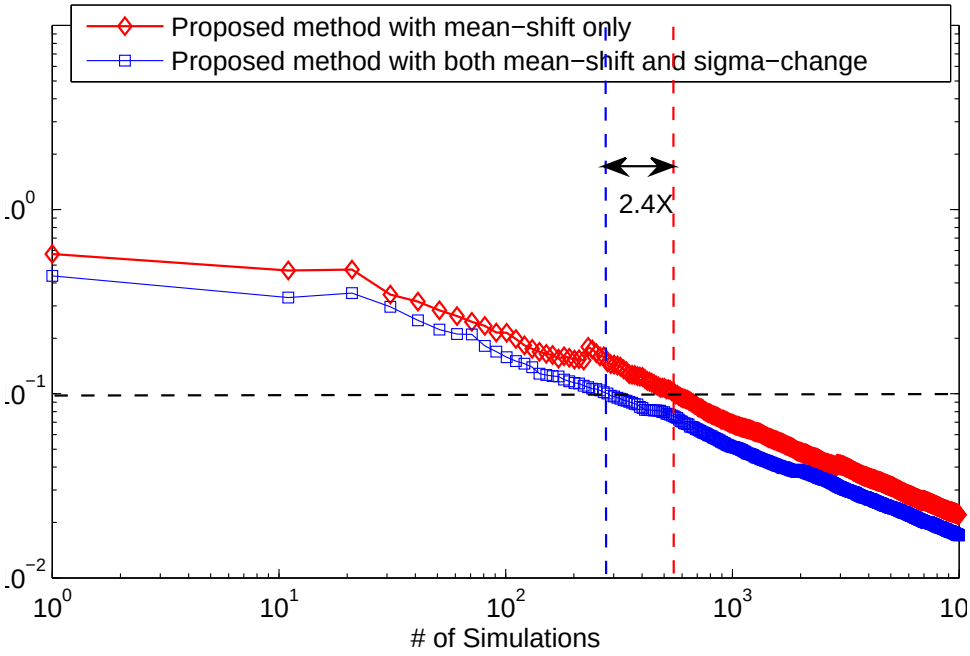


Figure 2.4: Comparison to validate the performance of sigma-change.

Moreover, we compare and show the given sampling distribution and the optimal sampling distribution for one PMOS in Fig. 2.5 as an illustration. This figure demonstrates the optimal sampling distribution not only shifts the mean value from -0.4118 to -0.5185, but also reduces the sigma from 0.0412 to 0.0302. Note that the sigma-change can also provide the potential sensitivity information which can be used for design optimization purpose, since the proposed method tends to find tightly peaked sampling distributions for those critical parameters in terms of the rare failures.

2.5 Conclusion

In this chapter, we presented an improved importance sampling method based on the probability collectives method to efficiently estimate the rare event failures of SRAM cells. This method adopts the "Kullback-Leibler (KL) distance" to

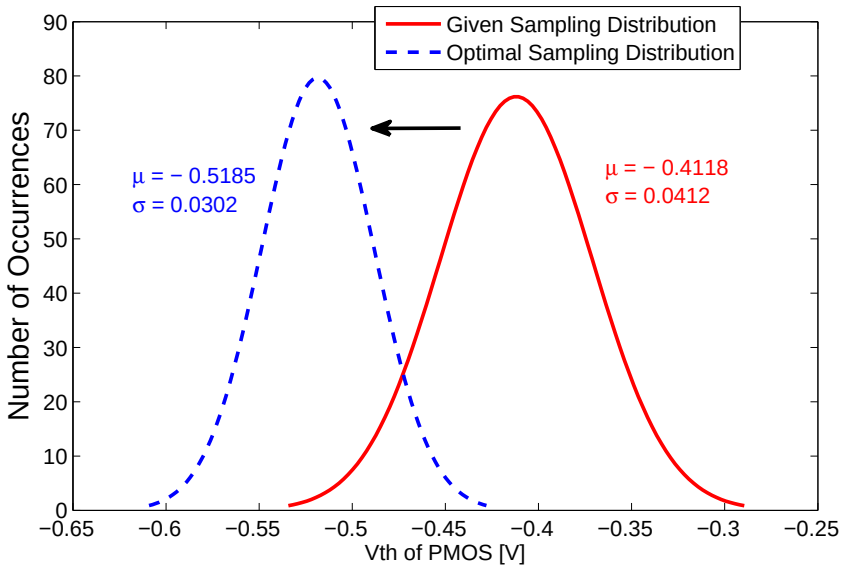


Figure 2.5: Comparison of given distribution and the optimal sampling distribution.

represent the distance between the optimal sampling distribution and a given sampling distribution. Then, the KL distance is further analytically minimized using immediate sampling based probability collectives method and parameterized Gaussian distributions are obtained as the optimal sampling distribution. The experiments demonstrate that the 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 [KJN06] and spherical sampling [QTD10]) with the same accuracy.

CHAPTER 3

Fast Failure Analysis of Memory Circuits in High Dimensions

3.1 Introduction

The reliability of memory circuits has become an increasingly challenging issue due to inevitable process variations in the manufacturing. For example, the critical components of memory circuits (e.g., SRAM bit-cell, delay chain, sense amplifier, etc.) need to be replicated millions or even billions times for very high capacity, thereby, making the circuit failure a "rare-event" with extremely small probability [AN06].

The analysis of rare events is usually analytical intractable due to high complexity of memory circuits. Sampling based methods must be used. The most straightforward approach is the Monte Carlo (MC) method, which repeatedly draws samples and evaluates circuit performance with transistor-level SPICE simulation. However, MC is extremely time-consuming for rare-event estimation because millions or even billions of samples are needed to capture one single failure.

To mitigate the complexity issue of the MC method, many statistical methodologies have been developed to predict the probability of rare failure events for SRAM cells in the past few years [SR09, KJN06, DQS08, QTD10, KHT10, GBD12, DL11]. These methods can be categorized into three groups:
(1) **Classification**: the approach in [SR09] makes use of a "classifier" to "block" those Monte Carlo samples that are unlikely to cause failures and simulates the

remaining samples. However, this method has two limitations. First, a perfectly accurate classifier is usually unavailable. A safety margin is used in [SR09] to prevent the classifier error. Second, the imperfect classifier can easily incur large error beyond the safety margin for circuits with irregular failure region and strongly nonlinear behavior, which typically cannot be detected by the approach in [SR09].

(2) **Importance Sampling**: several approaches in [KJN06,DQS08,QTD10,KHT10, GBD12] had been developed to construct a new "proposed" sampling distribution under which a "rare event" becomes "less rare" so that more failures can be easily captured. The critical issue is how to build an optimal proposed sampling distribution. Previous works investigated different approaches. For example, [KJN06] mixes a uniform distribution, the original sampling distribution and a "shifted" distribution centering around the failure region. The approaches in [DQS08, QTD10] simply shift the sampling distribution towards the point of failure region with a minimum L_2-norm. The work in [KHT10] uses "particle filtering" to tilt more samples towards the failure region. The approach in [GBD12] approximates the optimal sampling distribution with a parameterized sampling distribution by minimizing the Kullback-Leibler (KL) distance between them. These importance sampling based methods are plagued by the curse of high dimensionality [AB03,BB05,RG09]. In general, they can only be used for low-dimensional problems (e.g., those with a scope of 6-12 variables) but become very untrustworthy for high-dimensional problems.

(3) **Markov Chain Monte Carlo**: the approach in [DL11] uses a set of sample "chains" to explore the failure region with the aid of the Markov Chain Monte Carlo (MCMC) method. However, it is very difficult to cover the entire failure region with several chains of MCMC samples, particularly when tens or hundreds of random variables are considered.

Clearly, most of these existing approaches can successfully be applied to rel-

atively simple problems with a few random variables but, in general, perform poorly in high dimensions. Therefore, an effective and low-complexity approach is still urgently needed for rare-event analysis in high dimensions.

In this chapter, we propose a novel statistical algorithm to efficiently estimate the probability of rare events in high dimension, called "*High Dimensional Importance Sampling* (HDIS)". We also successfully apply HDIS to failure probability estimation of memory circuits with tens or hundreds of random variables. The proposed algorithm constructs a new subset of the sampling space that dominates the failure region and can be efficiently estimated with a few samples. Then, the probabilities of rare failure events can be evaluated using the product rule for conditional probability and an importance sampling-based method. More importantly, it is proved that the estimation of the proposed algorithm is always bounded, while the estimations of existing IS methods become unbounded in high dimensions. The experiments show that the proposed approach can achieve up to 708X speedup over the MC method on a 108-dimensional problem without compromising any accuracy. Also, the proposed method is 17X faster than a classification based method (e.g., Statistical Blockade [SR09]) while existing importance sampling methods (i.e., Spherical Sampling [DQS08, QTD10]) completely fail to provide reasonable accuracy.

In general, this work provides a fast and reliable rare-event analysis in high dimensions which can be applied to multiple application domains. In particular, this work enables further studies that were prohibitive before due to high dimensionality, such as the analysis of large-scale circuits, the variation analysis with more accurate models, fast system-level analysis with a large number of components and etc.

The rest of this chapter is organized as follows. Section 3.2 provides the necessary background on importance sampling and revisits the reasons for its failure in high dimensions. Section 3.3 describes the techniques underpinning the

proposed algorithm in detail. Section 3.4 provides experiments to validate the accuracy and efficiency of proposed method. Section 3.5 concludes this chapter.

3.2 Background

3.2.1 Formulation of Probability Estimation

Let $f(X)$ be a probability density function (PDF) for a random variable X (e.g., any process or electronic variable parameters) which is the input of a measurement process as shown in (4.1); the output Y is an observation (e.g., voltage, amplitude, period, etc.) with input X:

$$\underbrace{X}_{variable} \Rightarrow \boxed{Measurement,\ SPICE,\ etc.} \Rightarrow \underbrace{Y}_{observation} \tag{3.1}$$

Usually, it is of great interest to estimate the probability of Y from a small subset $\mathcal{S}$ of the entire sampling space. For example, a small subset is the "failure region" for SRAM design and includes all failed samples where performance constraints cannot be satisfied. Therefore, the probability $p(Y \in \mathcal{S})$ can be estimated as:

$$p(Y \in \mathcal{S}) = \int I(X) \cdot f(X) dX. \tag{3.2}$$

$$I(X) = \begin{cases} 0 & if\ Y \notin \mathcal{S} \\ 1 & if\ Y \in \mathcal{S} \end{cases}$$

where Y is the observation/performance with the input variable X and the indicator function $I(\cdot)$ identifies whether $Y \in \mathcal{S}$ or not. Note that the integral in equation (3.3) is intractable because the analytical formula of $I(X)$ is unavailable. Therefore, sampling based method must be used. For example, the MC method enumerates as many samples of X as possible according to $f(X)$ (e.g., $x_1, \cdots, x_n$)

and evaluates their indicator function values to estimate $p(Y \in \mathcal{S})$ as:

$$\tilde{p}(Y \in \mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} I(x_i) \xrightarrow{a.s.}{}_{n\to+\infty} p(Y \in \mathcal{S}). \tag{3.3}$$

Here $\tilde{p}(X \in \mathcal{S})$ is an unbiased estimate and can be very close to $p(X \in \mathcal{S})$ with a large number of samples.

3.2.2 Importance Sampling (IS)

When $Y \in \mathcal{S}$ is a *rare event*, the MC method becomes extremely inefficient because most $I(x_i)$ are zeros. Millions or billions samples of X are needed to capture only one failed sample from the failure region $\mathcal{S}$.

To deal with this issue, the *importance sampling* (IS) has been introduced to sample from a "proposed" sampling distribution $g(X)$ that tilts towards $\mathcal{S}$ where a rare-event becomes more likely to happen:

$$\begin{aligned} p_{IS}(Y \in \mathcal{S}) &= \int I(X) \cdot \frac{f(X)}{g(X)} \cdot g(X) dX \\ &= \int I(X) \cdot w(X) \cdot g(X) dX. \end{aligned} \tag{3.4}$$

Here, $w(X)$ is the "likelihood ratio" or the weight for each sample of X. $w(X)$ compensates for the discrepancy between $f(X)$ and $g(X)$ and unbiases the probability estimation under $g(X)$. Sampling based methods can be used to evaluate above integral as:

$$\tilde{p}_{IS}(Y \in \mathcal{S}) = \frac{1}{n}\sum_{j=1}^{n} w(\tilde{x}_j) \cdot I(x_j) \xrightarrow{a.s.}{}_{n\to+\infty} p(Y \in \mathcal{S}). \tag{3.5}$$

It is worthwhile to point out that $\tilde{x}_j$ $(j = 1, \cdots, n)$ follows the "proposed" sampling distribution $g(X)$ rather than the original distribution $f(X)$. As such,

it becomes much easier to obtain rare-event samples from the subset $\mathcal{S}$ by sampling from $g(X)$.

Theoretically, $\tilde{p}_{IS}(Y \in \mathcal{S})$ is consistent with $p(Y \in \mathcal{S})$ in (3.3) if $supp(g(X)) \supset supp(I(X) \cdot f(X))$, where $supp(\cdot)$ denotes the support of a probabilistic distribution.

3.2.3 Failure Analysis of Importance Sampling

While importance sampling is, in principle, mathematically correct, the *degeneration* or *collapse* of the likelihood ratios leads to the failure of importance sampling in high dimensions as discussed in [BB05, RG09].

Let us consider a classical case, as shown in Fig. 3.1, where $f(X)$ is the original sampling distribution and $g(X)$ is the proposed sampling distribution. The small circles with the same size following $g(X)$ are samples drawn from $g(X)$. In the bottom of Fig. 3.1, a few circles with different sizes represent the illustrative scales of the likelihood ratios corresponding to the samples on top of them. Clearly, if $g(X)$ has thinner tails than $f(X)$, the likelihood ratios $w(X) = f(X)/g(X)$ approach infinity in the tails of $g(X)$. The likelihood ratios thus vary dramatically, have extremely large variance and lead to unstable probability estimate.

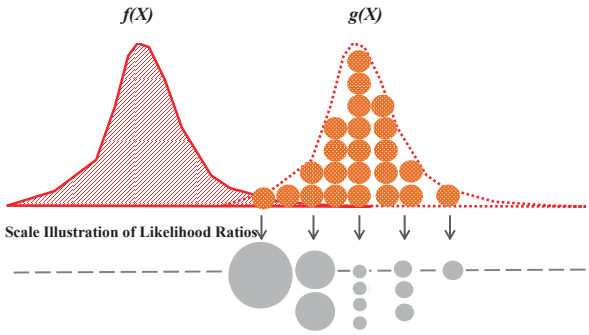


Figure 3.1: The scale illustration of likelihood ratios in importance sampling.

Moreover, the reason for the collapse of likelihood ratio can be explained from

another perspective: when importance sampling shifts $g(X)$ towards the rare-event region that is typically in the tails of $f(X)$, $f(X)$ and $g(X)$ become mutually singular and have "disjoint" support [BB05]. Therefore, IS fails to retain its accuracy.

This collapse issue of likelihood ratios in importance sampling becomes much more severe in high dimensions because $w(X)$ is a product of probabilities for multiple parameters and consequently approaches infinity more quickly.

3.3 Proposed Method

3.3.1 Algorithm Overview

We consider a small subset $\mathcal{S}$ as the failure region in SRAM design under the given performance constraint (e.g., the performance of SRAM circuit Y should be greater than certain performance threshold t_c). As such, the subset $\mathcal{S} = \{Y|Y \geq t_c\}$ contains all failed samples which should be "rare events".

The basic idea of the proposed algorithm is to construct a new subset $\mathcal{T}$ with certain threshold t (e.g., $t = 0.99$-quantile point so that $\mathrm{P}(Y \in \mathcal{T}) = 0.99$, which includes "non-rare" events and dominates the subset $\mathcal{S}$ containing rare events(e.g., $supp(\mathcal{T}) \supset supp(\mathcal{S})$).

In this way, the desired failure probability of SRAM design can be estimated by a product rule from the probability theory [PP01]:

$$P(Y \geq t_c) = \quad P(Y \geq t) \cdot P(Y \geq t_c | Y \geq t). \tag{3.6}$$

The proposed algorithm can be illustrated with Fig. 3.2 which consists of two stages:

1) Initial Sampling with MC: This step aims to evaluate the probability $P(Y \in$

$\mathcal{T}) = P(Y \geq t)$ where t is the threshold, such as $t =$ 0.99-quantile point shown in the left of Fig. 3.2. Since the samples in $\mathcal{T}$ are "non-rare" events, this evaluation needs only a few samples using standard MC method.

2) Conditional Probability Estimation: The most difficult task of the proposed approach is to efficiently calculate the conditional probability $P(Y \geq t_c|Y \geq t)$ with high accuracy. For this purpose, we propose an importance sampling-based method which takes two steps to construct the "proposed" sampling distribution $g(X)$: first, the original sampling distribution $f(X)$ is shifted towards a "non-rare" subset $\mathcal{T} = \{Y|Y \geq t\}$; second, a larger standard deviation is properly chosen for the shifted sampling distribution to reach the failure region $\mathcal{S} = \{Y|Y \geq t_c\}$ (shown in the right of Fig. 3.2). We present more details in the following sections.

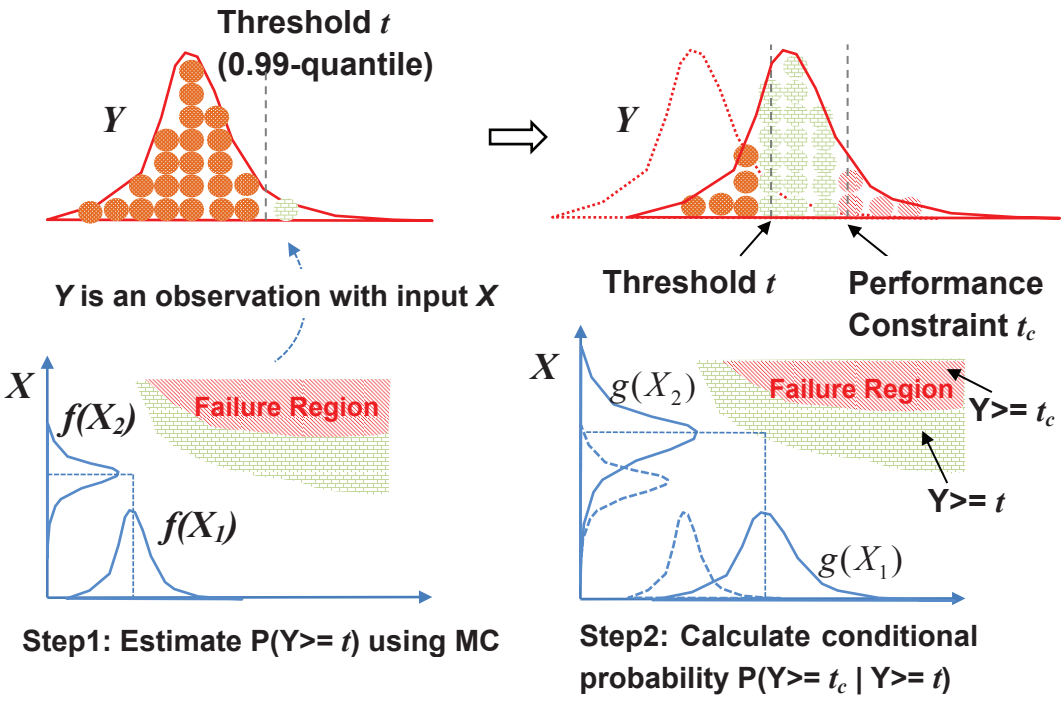


Figure 3.2: Overall flow in proposed algorithm.
(Noted that $\mathcal{T} = \{Y|Y \geq t\}$ contains $\mathcal{S} = \{Y|Y \geq t_c\}$).

Remarks: For better understanding purpose, we discuss the two-fold motivations intuitively behind the construction of $g(X)$: first, $f(X)$ is shifted towards a

"non-rare" subset $\mathcal{T}$ which is typically around the mean of $f(X)$. Thus the shifted distribution share almost the same support with $f(X)$ to avoid the "disjoint support" issue. Second, the "proposed" sampling distribution should dominate or completely cover the "rare-event" region $\mathcal{S}$. Hence, a larger standard deviation is chosen to easily draw samples from $\mathcal{S}$ which is typically in the tails of $f(X)$. In this way, a "proposed" sampling distribution $g(X)$ can be obtained.

The overall algorithm flow is described in Algorithm(3). Section 3.4 shows that the proposed algorithm can handle more than one hundred random variables by providing MC accuracy and up to 708X speedup. Also, it is 17X faster than the classification based method such as statistical blockade [SR09].

Algorithm 3 Overall Algorithm

Input: random variables X with sampling distributions $f(X)$ and performance constraints $Y \geq t_c$.
Output: the estimation of failure probability $p_{IS}(Y \geq t_c)$.

1: **/* 1: Initial Sampling with MC */**
2: Use few MC samples to find the threshold value t of performance (e.g., t = 0.99-quantile point).
3: Run standard Monte Carlo method to calculate $P_{MC}(Y \geq t)$ with certain accuracy level.
4:
5: **/* 2: Conditional Probability Calculation */**
6: Shift the sampling distribution $f(X)$ towards a "non-rare" subset $\mathcal{T} = \{Y|Y \geq t\}$.
7: Choose the standard deviation of sampling distribution to construct $g(X)$.
8: Generate samples from $g(X)$ and evaluate $P(Y \geq t_c|Y \geq t)$ using importance sampling-based method.
9:
10: **/* 3: Failure Probability Estimation */**
11: Solve for the failure probability $p_{IS}(Y \geq t)$ as

$$P_{IS}(Y \geq t_c) = P_{MC}(Y \geq t) \cdot P(Y \geq t_c|Y \geq t).$$

The key problem is to accurately calculate conditional probability with as few samples as possible. There exist several issues that need to be resolved:

(1) The original sampling distribution $f(X)$ needs to be shifted towards $\mathcal{T}$ but it is, at the moment, unclear how to find the shift vector for $f(X)$.

(2) It is important and nontrivial to find the value of the standard deviation for the proposed sampling distribution $g(X)$ since the standard deviation can significantly affect both accuracy and convergence speed in Algorithm(3).

(3) With the properly-chosen $g(X)$, an importance sampling-based method is needed to calculate the conditional probability where conventional IS cannot be applied.

(4) It is desired to investigate the robustness of proposed algorithm. For example, it is of great interest to study whether the estimations of proposed algorithm is always bounded or not.

The following sections discuss how we solve these issues.

3.3.2 Calculation of Conditional Probability

3.3.2.1 Mean-Shift Vector Selection

The first issue is to find the shift vector for sampling distribution $f(X)$. This is a classical problem in previous rare-event estimation works [KJN06, DQS08, QTD10, KHT10, GBD12]. Even though these works deploy different techniques, they indeed share the same basic idea: shift the sampling distribution towards the point where the failed samples are most likely to happen. In this work, we adopt the insights from [GBD12] which provides a close-to-optimal sampling distribution.

For illustration purpose, we consider a 1-D example but similar technique can be easily applied to high dimension as well. The algorithm in [GBD12] starts with an initial parameterized distribution $\hat{f}(X, \hat{\mu})$ and tries to update the mean value in order to achieve a close-to-optimal sampling distribution $f^*(X, \mu^*)$ by an

analytic formula:

$$\mu^* = \frac{\sum_{i=1}^{N} I(x_i) \cdot w(x_i) \cdot x_i}{\sum_{i=1}^{N} I(x_i) \cdot w(x_i)}. \tag{3.7}$$

Here x_i $(i = 1, \cdots, N)$ are samples drawn from $\hat{f}(X, \hat{\mu})$ and $w(x_i)$ are their likelihood ratios as $w(x_i) = f(x_i)/\hat{f}(x_i, \hat{\mu})$.

Intuitively, the updated mean value μ^* can be viewed as the coordinates of the *centroid point* in the failure region where the failed samples are most likely to happen. This interesting finding becomes more obvious if $\hat{f}(X, \hat{\mu})$ equals $f(X)$ and all likelihood ratios take on value 1. Hence, μ^* is:

$$\mu^* = \frac{\sum_{i=1}^{N} I(x_i) \cdot x_i}{\sum_{i=1}^{N} I(x_i)}. \tag{3.8}$$

Therefore, our proposed algorithm shifts the sampling distribution towards the "centroid point" of the subset $\mathcal{T} = \{Y|Y \geq t\}$, which can be calculated with available MC samples from the first step in Algorithm (3) and requires no extra sampling/simulation.

3.3.2.2 Standard Deviation Selection

The second issue is to choose the standard deviation for the proposed sampling distribution $g(X)$. As an illustration, let us consider a 2-D problem in Fig. 3.3. The same method can be applied to high dimensional problems as well.

Note that $f(X)$ has been shifted to the centroid point of subset $\mathcal{T} = \{Y|Y \geq t\}$ (as marked in Fig. 3.3). The problem now becomes how to choose the standard deviation of the proposed sampling distribution $g(X)$ to obtain the samples in $\mathcal{S} = \{Y|Y \geq t_c\}$.

The proposed algorithm first approximates the centroid point of $\mathcal{S} = \{Y|Y \geq t_c\}$ using uniformly-distributed samples and then calculates the distance between these two centroid points along each parameter axis (e.g., d_{X_1} and d_{X_2} shown in

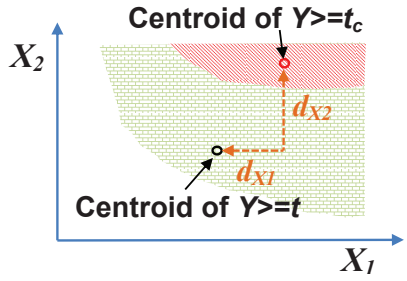


Figure 3.3: The distance between centroid points of two subsets along each parameter axis.

Fig.3.3). Then, we choose $max(d_{X_i}, \sigma_{(0,X_i)})$ as the standard deviation of $g(X_i)$ for the variable X_i, where $\sigma_{(0,X_i)}$ is the original standard deviation of $f(X_i)$. This choice can be intuitively explained as following:

- $d_{X_i} > \sigma_{(0,X_i)}$**:** the failure region $\mathcal{S}$ is very far away from the subset $\mathcal{T}$, therefore, the larger value d_{X_i} is used to extend the range of $g(X_i)$ and obtain the rare-event samples in the failure region. In the meantime, $g(X_i)$ has almost the same supports with $f(X_i)$ because its mean position locates at the centroid point of $\mathcal{T}$ and is not far away from $f(X_i)$.

- $d_{X_i} < \sigma_{(0,X_i)}$**:** Suppose the smaller one, d_{X_i}, is chosen as the standard deviation of $g(X_i)$, the proposed sampling distribution $g(X)$ will have much smaller sampling space, thereby, making it fail to keep the same supports with $f(X_i)$ and suffer from "disjoint supports" issue. The proposed algorithm chooses $\sigma_{(0,X_i)}$ as the standard deviation of $g(X_i)$ in this case.

3.3.2.3 Importance Sampling-based Method

With the proposed sampling distribution $g(X)$, an importance sampling-based method has been proposed in this work to estimate the conditional probability in

Algorithm(3). We can start with the product rule in the probability theory [PP01]:

$$P(Y \geq t_c | Y \geq t) = \frac{P(Y \geqslant t_c, Y \geqslant t)}{P(Y \geqslant t)}. \tag{3.9}$$

In addition, the subset $\mathcal{T} = \{Y | Y \geq t\}$ dominates the failure region $\mathcal{S} = \{Y | Y \geq t_c\}$ (e.g., $\mathcal{T} \supset \mathcal{S}$), which implies following two properties [PP01]:

$$\begin{aligned} P(Y \geqslant t_c, Y \geqslant t) &= P(Y \geqslant t_c), \\ P(Y \geqslant t) &\geqslant P(Y \geqslant t_c). \end{aligned} \tag{3.10}$$

Moreover, when samples x_i $(i = 1, \cdots, N)$ are generated from $g(X)$, both $P(Y \geqslant t_c)$ and $P(Y \geqslant t)$ can be estimated by conventional importance sampling method. Thus, the equation (3.9) becomes:

$$\begin{aligned} P_{MIS}(Y \geq t_c | Y \geq t) &= \frac{P(Y \geqslant t_c)}{P(Y \geqslant t)} \\ &= \frac{\frac{1}{N} \sum_{i=1}^{N} w(x_i) \cdot I_{\{Y \geqslant t_c\}}(x_i)}{\frac{1}{N} \sum_{i=1}^{N} w(x_i) \cdot I_{\{Y \geqslant t\}}(x_i)}. \end{aligned} \tag{3.11}$$

where $I_{\{Y \geqslant t_c\}}(\cdot)$ and $I_{\{Y \geqslant t\}}(\cdot)$ are indicator functions for subsets $Y \geq t_c$ and $Y \geq t$, respectively. $w(x_i)$ are likelihood ratios for these samples. In this way, the conditional probability can be efficiently evaluated under proposed sampling distribution $g(X)$.

3.3.3 Analysis of Boundedness

3.3.3.1 Importance Sampling

Let us first investigate the existing importance sampling and assume samples x_j $(j = 1, \cdots, M)$ are generated from the proposed sampling distribution $g(X)$.

We find the upper bound of probability estimate from the conventional importance sampling according to Boole's inequality (also known as the union bound from probability theory [PP01]) as:

$$\begin{aligned} P(Y \geqslant t_c) &= P_f(\sum_{j=1}^{M} I_{\{Y \geqslant t_c\}}(x_j)) \leqslant \sum_{j=1}^{M} P_f(x_j) \cdot I_{\{Y \geqslant t_c\}}(x_j) \\ &= \sum_{j=1}^{M} w(x_j) \cdot I_{\{Y \geqslant t_c\}}(x_j). \end{aligned} \tag{3.12}$$

where P_f stands for the probability estimation under sampling distribution $f(X)$. As discussed in [BB05, RG09], the likelihood ratios $w(x_j)$ can vary dramatically in high dimension and be any random quantities. Therefore, the union bound of the estimation $P(Y \geqslant t_c)$ in (3.12) approaches infinity and importance sampling becomes unreliable and untrustworthy.

3.3.3.2 Proposed Algorithm

The proposed algorithm constructs a subset $\mathcal{T} = \{Y|Y \geq t\}$ that *dominates* the failure region $\mathcal{S} = \{Y|Y \geq t_c\}$ (i.e., $\mathcal{T} \supset \mathcal{S}$). Therefore, the upper bound of conditional probability can be derived using the properties in (3.10) as:

$$\begin{aligned} P(Y \geqslant t_c|Y \geqslant t) &= \frac{P(Y \geqslant t_c)}{P(Y \geqslant t)} \\ &= \frac{\sum_{j=1}^{N} w(x_j) \cdot I_{\{Y \geqslant t_c\}}(x_j)}{\sum_{j=1}^{N} w(x_j) \cdot I_{\{Y \geqslant t\}}(x_j)} \leqslant 1. \end{aligned} \tag{3.13}$$

Clearly, no matter how likelihood ratios $w(x_j)$ vary, the conditional probability estimation of proposed algorithm is always bounded by the upper bound 1 if and only if the calculations of both $P(Y \geqslant t_c)$ and $P(Y \geqslant t)$ utilize the *same* set of samples x_j $(j = 1, \cdots, M)$ drawn from $g(X)$. Therefore, the rare-event estimation

of proposed algorithm can reliably provide bounded estimation results.

3.4 Experimental Results

In general, the proposed HDIS algorithm is intrinsically application-independent and can be applied to broad disciplines. As an example, we investigate its performance for a failure analysis on memory circuits (e.g., SRAM bit-cell and delay chain) in this section. All experiments are performed using MATLAB and H-spice with BSIM4 transistor model. In addition, Monte Carlo (MC), statistical blockade (SB) [SR09], and spherical sampling (SS) [DQS08, QTD10] have been implemented for comparison purpose.

3.4.1 SRAM Circuit and Variation Modeling

A functional diagram of SRAM circuit with one bit-cell column is shown in Fig. 3.4, which consists of a decoder, bit-cells, a sense amplifier and a delay chain [PS08]. Let us consider a reading operation to illustrate the functionalities of these components: the bit-cells store the data in forms of '0' or '1'; the decoder generates an address of a specific bit-cell and releases a read enable signal. As such, the chosen bit-cell starts to discharge the bit-lines (i.e., the lines that connect to all bit-cells) to produce a voltage difference between two bit-lines. The delay chain serves as a timing control unit and aims to activate the sense amplifier which reads out the stored data by capturing the voltage difference on bit-lines before the read operation is complete.

The process variations are introduced into each transistor of SRAM circuit, which are modeled by 9 process parameters shown in Table 4.1. The parameters are physically independent [DM03] and can be considered to be Gaussian random variables. Note that the threshold voltage V_{th} is not a process parameter and depends on V_{fb}, t_{ox}, ΔL and ΔW through related effects [DM03].

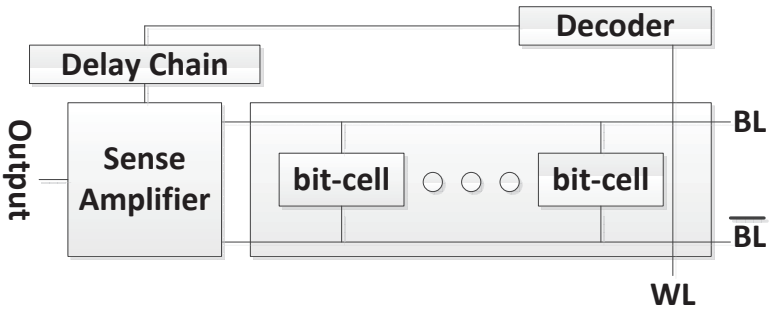


Figure 3.4: Functional diagram of an SRAM circuit.

Table 3.1: Process Parameters of MOSFETs.

Variable Name	σ/μ	unit
Flat-band Voltage (V_{fb})	0.1	V
Gate Oxide Thickness (t_{ox})	0.05	m
Mobility (μ_0)	0.1	m^2/Vs
Doping concentration at depletion (N_{dep})	0.1	cm^{-3}
Channel-length offset (ΔL)	0.05	m
Channel-width offset (ΔW)	0.05	m
Source/drain sheet resistance (R_{sh})	0.1	Ohm/mm^2
Source-gate overlap unit capacitance (C_{gso})	0.1	F/m
Drain-gate overlap unit capacitance (C_{gdo})	0.1	F/m

Table 3.2: Comparison for SRAM bit-cell with 90% target accuracy and confidence level.

	MC	SS [QTD10]	SB [SR09]	Proposed (HDIS)
failure probability	2.413E-05	2.8415E-05	2.7248e-05	2.4949E-05
relative error	(0%)	(+17.7%)	(+12.9%)	(+3.39%)
#sim. runs	4.6e+6	2e+4	8.16e+5	4e+3
speedup	(1150X)	(5X)	(204X)	(1X)

3.4.2 SRAM Cell with Reading Failure

A typical 6-transistor SRAM bit-cell is shown in Fig. 3.5: $Mn2$ and $Mn4$ control the accessing of the cell; the remaining four transistors form two inverters and use two stable states (either '0' or '1') to store the data in this memory cell. The *reading access failure* happens when the voltage difference between $\bar{BL}$ and BL is too small to be sensed by the sense amplifier at the end of reading operation [AN06].

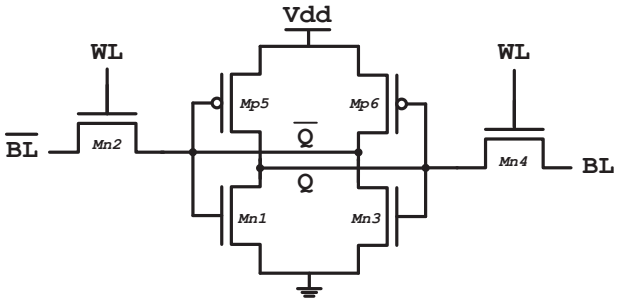


Figure 3.5: The schematic of the 6T SRAM cell.

We perform different methods (MC, SS [QTD10], SB [SR09], HDIS) on this 6-T SRAM bit-cell example to predict the reading failure probability under process variations and the comparison results are shown in Table 3.2.

3.4.2.1 Accuracy Comparison

At a first glance, we would be very surprised to find that SS [QTD10] method based on conventional importance sampling framework can provide accurate failure rate predictions in this 54-dim problem!

However, this comparison cannot allow us to reach that conclusion, because this SRAM bit-cell example is a "pseudo" high-dimensional problem for two-fold reasons: (1) during the reading operation, not all transistors are active. In fact, both $Mp5$ and $Mn3$ are shut off, therefore, the process variations on these two

transistors have no effect on discharge behavior of bit-lines at all; (2) without loss of generality, assuming $\bar{BL}$ = '0' and BL='1', the discharge current flows from $\bar{BL}$ to the ground through $Mn2$ and $Mn1$ so that to pull down the voltage of $\bar{BL}$. As such, the process variations in $Mn2$ and $Mn1$ have more significant effects on the discharge behavior of bit-lines and can potentially mask the variation effects in $Mp6$ and $Mn4$. In this way, there are only 18 "effective" variable parameters, which suggests that this example is a problem with modest dimension.

When compared with MC results, the proposed HDIS method provides the most accurate failure probability estimation with only 3.39% relative error, while the estimations from SS [QTD10] and SB [SR09] have more than 10% relative error.

3.4.2.2 Efficiency Comparison

From Table 3.2 we also compare the efficiency of these methods: MC is very time-consuming and requires nearly 4.6 millions transistor-level SPICE simulations; SB [SR09] can provide 6X complexity reduction by screening out and simulating those "most-likely-to-fail"samples; SS [QTD10] method is made more efficient (230X speedup over MC) by better choosing failed samples using importance sampling algorithm; the proposed HDIS algorithm achieves the best convergence rate (1150X faster than MC) by efficiently spreading more samples into the failure region using a sampling distribution with a large-standard-deviation in high dimensions.

3.4.3 Delay Chain for Target Delay

Next, we consider a delay chain example which includes 6-stage inverters as shown in Fig. 3.6.

In general, the delay chain generates timing intervals as the control signals for

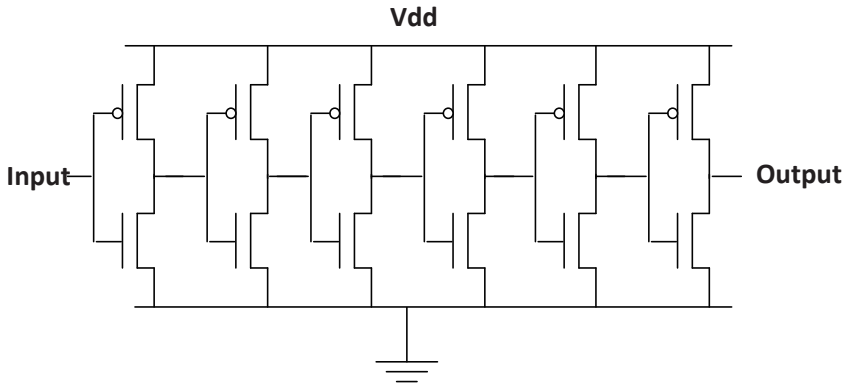


Figure 3.6: The schematic of a delay chain circuit.

the read/write operation, which should match with the delay of the discharge on bit-lines. Due to the process variations, the timing interval from the delay chain could become very small and activate the sense amplifier too early in the reading operation when the voltage difference on bit-lines is not large enough to be sensed. Therefore, a timing failure happens.

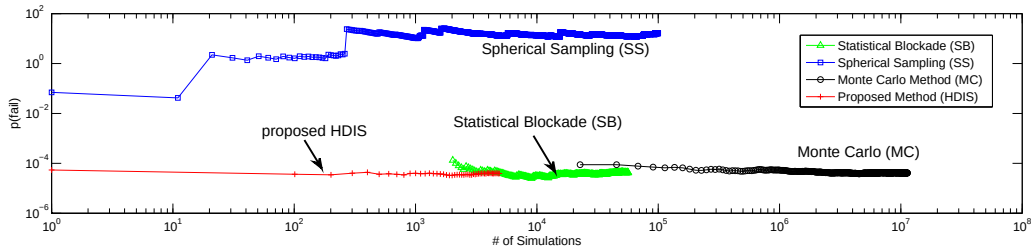
With the variation modeling summarized in Table 4.1, the delay chain example has 108 random variables in total. More importantly, all of these variable parameters are "effective" because all transistors are active and process variations on each transistor can significantly change the delay interval, which is a truly high-dimensional problem.

3.4.3.1 Accuracy Comparison

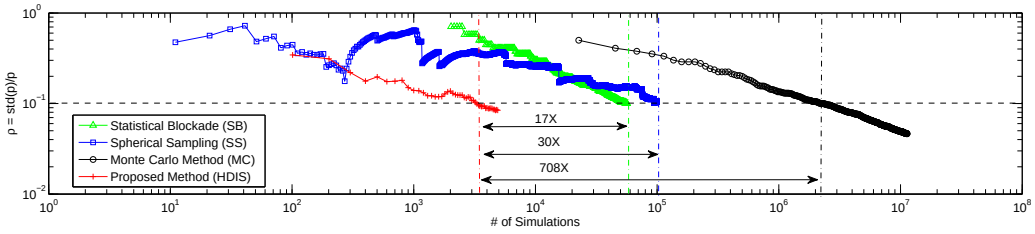
To validate the accuracy of the proposed algorithm, we apply different methods (MC, SB [SR09], SS [QTD10] and HDIS) on this 108-dim delay chain problem to predict the timing failure probability. Here, MC serves as the "gold standard".

The evolution of the probability estimation in different methods are plotted in Fig. 3.7(a). Several observations can be made:

- First, this figure shows the failure of conventional importance sampling (i.e.,



(a) failure probability



(b) figure of merit

Figure 3.7: Evolution comparison of the failure probability estimation and figure of merit for different methods.

SS [QTD10]). In fact, due to the degeneration or collapse of likelihood ratios, SS [DQS08, QTD10] method converges to a random quantity which is obviously wrong and far away from the MC result. Moreover, SS [QTD10] does not have a mechanism for improving accuracy even though more samples are added.

- SB [SR09] filters those "most-likely-to-fail" samples using machine learning technique without using likelihood ratios. Therefore, it can provide accurate failure probability. However, it captures no failure in the first thousand or so samples at all because it manipulates MC samples directly and cannot draw failed samples from the failure region more efficiently.

- The proposed HDIS method uses the likelihood ratios. But it calculates the conditional probability using importance sampling-based method with an

Table 3.3: Comparison for delay chain analysis with 90% target accuracy and confidence level.

Target failure probability		MC	SS [QTD10]	SB [SR09]	Proposed (HDIS)
8e-3	prob.(failure)	0.0088125 (0%)	0.89646	0.0077333 (-12.2%)	0.0088552 (+0.48%)
	#sim. runs	3.2e+4 (16X)	5e+3	5.9e+3 (6X)	1e+3 (1X)
6e-4	prob.(failure)	0.00065714 (0%)	2.6305	0.00068667 (+4.5%)	0.00066885 (+1.7%)
	#sim. runs	7e+4 (54X)	1.5e+5	1.4e+4 (10X)	1.3e+3 (1X)
4e-5	prob.(failure)	3.897e-05 (0%)	16.2873	3.7244e-05 (-4.4%)	3.832e-05 (-1.6%)
	#sim. runs	2.338e+6 (708X)	1e+5	5.6e+4 (17X)	3.3e+3 (1X)

effective proposed sampling distribution, which can tolerate the degeneration of likelihood ratios and is theoretically bounded. Therefore, HDIS can reliably estimate the failure probability that matches with MC results.

3.4.3.2 Efficiency Comparison

Even though the Fig. 3.7(a) provides a rough comparison of efficiency, the detailed comparison can be shown in Fig. 3.7(b), where different methods try to achieve the "comparable" accuracy. Note that circuit simulation is the most time-consuming part and the runtime cost of the remaining computation becomes negligible. As such, the required number of circuit simulations for the same accuracy and confidence level serves as a measurement of the efficiency.

First, the Figure-Of-Merit (FOM) is used to quantify the accuracy of probability estimation as [DQS08, QTD10]:

$$\rho = \frac{\sqrt{\sigma^2_{p(fail)}}}{p(fail)}. \tag{3.14}$$

where $p(fail)$ is the failure probability and $\sigma_{p(fail)}$ is the standard deviation of $p(fail)$. In fact, the FOM can be viewed as a *relative error* so that lower FOM means higher accuracy of probability estimation.

We compare the evolutions of FOM for different methods in Fig. 3.7(b) and draw a dash line to indicate the 90% accuracy with 90% confidence ($\rho = 0.1$).

And we can have following observations:

- SS [QTD10] has reached $\rho = 0.1$ but its estimation is completely wrong. Clearly, it cannot detect the failure at all. The same observation is applied to other existing importance sampling methods due to the boundedness analysis in Section 3.3.3.

- SB [SR09] simulates "most-likely-to-fail" samples and captures more failed samples with fewer simulations. However, it has to manipulate an extremely huge number of MC samples and suffer from the undetectable error due to the imperfect classifier.

- The proposed HDIS algorithm can provide the accurate estimation of failure probability with only a few thousands samples, which dramatically relieves the requirements of computing and storage efforts. As shown in this figure, the proposed method can achieve $708X$ speedup over Monte Carlo and be $17X$ faster than statistical blockade method [SR09].

3.4.3.3 Comparison for Different Failure Probabilities

We study various methods on the delay chain example with three different failure probabilities summarized in Table 3.3. It is obvious that SS [QTD10] method fails to achieve any reasonable accuracy in all these cases. This demonstrates the failure of conventional importance sampling method. On the contrary, the estimates from SB [SR09] and the proposed HDIS method match the MC result.

In addition, the table reveals that the proposed HDIS method provides the fastest convergence speed in all these cases and, more importantly, offers substantial complexity reduction as the failure probability becomes smaller. This property makes HDIS suitable for industrial problems where exist "rare events" with extremely small probability.

3.4.3.4 Discussion on Statistical Blockade

When compare the performance of SB [SR09] on these two examples, we may observe better convergence rate for the delay chain example in high dimension. However, it is not safe to conclude SB [SR09] provides better efficiency in high dimension due to below reasons: first, SB [SR09] adopts "linear classifier" to predict the circuit performance, thereby, making it unsuitable for strongly nonlinear circuits. Second, a safety margin defined in [SR09] is used to compensate the error of "classifier" which can significantly affect both the accuracy and efficiency. In fact, the SRAM bit-cell has strongly nonlinearity in the discharge behavior, and a relax safety margin (e.g., 0.95-quantile point) is used to prevent the error of classifier. Thus, more "likely-to-fail" samples are screened out and actually 8e+5 samples are simulated. On the contrary, weakly-nonlinear delay chain example can have tight safety margin (i.e., 0.99-quantile point) without significant accuracy loss. Hence, SB [SR09] screens out only $\sim 6e + 4$ samples to be simulated.

3.5 Conclusion

In this chapter, a fast failure analysis method for memory circuits (e.g., SRAM bit-cell, delay chain) in high dimensions is proposed which has proved to be bounded. Experiments show that the proposed method can provide 708X speedup over MC with the same accuracy for a 108-dimensional problem. Also, the proposed approach is 17X faster than the Statistical Blockade method [SR09] and trumps existing importance sampling methods that completely fail to provide any reasonable accuracy.

CHAPTER 4

Stochastic Behavioral Modeling and Analysis

4.1 Introduction

Large-scale process variations have become inevitable in the nano-technology era [Nas01, LZP08] and significantly change the behavior of custom integrated circuits (e.g. voltage swing, timing delay, clock frequency, etc.) [BDM02, CCS04, EBS97, YLW10, GYH09, GYS10, VWG06]. Therefore, it is urgently sought to accurately extract the probabilistic behavioral distribution of custom circuits under process variations.

In general, there are two types of process variation sources: systematic global variation and local random variation. In this chapter, we focus on the local variation which is purely random and more difficult to model. The most straightforward approach is crude Monte Carlo (MC) method [GS96], which utilizes massive samples and expensive SPICE simulations to evaluate the probabilistic distributions (e.g., probability density function (PDF) and cumulative distribution function (CDF)) of circuit behavior. MC method can be easily applied to any variable parameter and circuit behavior with arbitrary distributions. However, it is too time-consuming and not affordable.

Instead, many statistical methods have been developed in past few years: the linear regression method [Nas01] models the circuit behavior as a linear function of a number of normally distributed process variables and thus becomes inaccurate for strongly nonlinear circuits. The work in [XK02, VWG06] can estimate the

unknown distribution of circuit behavior with stochastic orthogonal polynomials (SoPs) but requires prior knowledge of the distribution type which is unavailable in practice.

In addition, asymptotic waveform evaluation (AWE) [PR90] approximates the "arbitrary" circuit behavioral distribution with the impulse response of a linear time-invariant (LTI) system by matching a few high-order moments. This approach requires no prior knowledge of the circuit behavioral distribution but needs expensive computational efforts to evaluate the high-order moments. Note that our work in this chapter is based on the AWE framework but proposes a novel method to calculate high-order moments efficiently with high accuracy.

To resolve this issue of AWE [PR90], response-surface-method (RSM) based methods [LLG04, LL08] have been proposed to model the circuit behavior as a polynomial function of all variable process parameters and further evaluate the high-order moments. For example, asymptotic probability extraction (APEX) [LLG04] evaluates the RSM model using ordinary least-square (OLS) regression method so that the number of needed SPICE simulations equals the total number of unknown coefficients in the polynomial function of the RSM model. Moreover, a novel approach has been proposed [LL08] to extract all unknown coefficients of RSM model from a small number of samples with regularization based regression method. In fact, this approach finds a unique sparse solution of an under-determined equation system using L_0-norm regularization [Li10].

However, these RSM based methods have been plagued by following issues: first, fully nonlinear custom circuits tend to need *higher* order RSM models (e.g. strongly nonlinear functions of random process variables) where the number of unknown coefficients and required SPICE simulations in OLS based RSM method can increase exponentially, thereby, making the OLS based RSM method infeasible; second, the regularization based regression method [LL08] suffers from bias-variance tradeoff [HF08] which can potentially degrade the accuracy and robust-

ness of the extracted RSM models; third, when a large number of process variables are considered, the RSM model becomes highly complicated and requires more computational efforts. Therefore, an efficient and accurate method to evaluate high order moments and further extract the stochastic behavior of custom circuits is, still, urgently sought.

In this chapter, we propose a novel and efficient algorithm to accurately predict the arbitrary probabilistic distributions of circuit behavior based on asymptotic waveform evaluation [PR90]. This approach first utilizes Latin Hypercube Sampling (LHS) method along with correlation control technique to generate a few samples (e.g. sample-size is in linear with respect to the number of variable parameters) and further evaluates the high-order moments accurately with analytical formulae. Then, the PDF/CDF of stochastic circuit behavior can be recovered using conventional moment-matching method in AWE. In addition, a normalized PDF function is introduced to enhance the accuracy by eliminating the potential numerical errors. The experiments demonstrate that the proposed method provides very high accuracy along with up to 1666X speed-up when compared with MC.

It is worth noting the benefits that the proposed work can offer:

- This method does not need RSM models to estimate the moments and, therefore, avoids the aforementioned exponential complexity and bias-variance tradeoff.
- The proposed method can handle strongly nonlinear custom circuits and high dimensional problems with a large number of random process variables.
- This approach can achieve nearly linear complexity while providing high accuracy of behavioral distributions.

The rest of this chapter is organized as follows. Section 4.2 presents the necessary background knowledge and Section 4.3 describes the high order moments

estimation. The PDF/CDF calculation is presented in Section 4.4. Section 4.5 summarizes the overall algorithm. The experiments are provided in Section 4.6. This chapter is concluded in Section 4.7.

4.2 Background

4.2.1 Mathematical Formulation

Assume $\vec{x} = [x_1, x_2, \cdots]$ is a vector of random variables (e.g.,threshold voltage, channel length, gate oxide capacitance, etc.) and can be characterized by a sequence of probabilistic distributions $[pdf(x_1), pdf(x_2), \cdots]$ where $pdf(x_i)$ is the PDF function associated with the element x_i of $\vec{x}$. These random variables can be fed into SPICE simulator engines and the output is the circuit behavior y (e.g. voltage, bandwidth, power, etc.) as shown below:

$$\underbrace{\vec{x}}_{variable} \Rightarrow \boxed{SPICE\ simulators} \Rightarrow \underbrace{y}_{circuit\ behavior} . \tag{4.1}$$

Clearly, there exist two spaces: *"parameter space"* contains all possible values of $\vec{x}$ and *"performance space"* has all possible values of y. In fact, there is a *mapping* from the parameter space to the performance space so that each sample of $\vec{x}$ has its corresponding y. Mathematically, the mapping can be viewed as an implicit function $y = f(\vec{x})$ which, unfortunately, has no analytical formula. Therefore, our aim is to determine the *unknown* probabilistic distribution of y that results from the uncertainties in $\vec{x}$.

To this purpose, the high-order moments of y need to be evaluated and then the probabilistic distributions of y can be recovered by AWE method as proposed in [PR90]. According to probability theory [PP01,DS11], the p-th order probabilistic

moments of y can be defined as:

$$m_y^p = E(y^p) = \int_{-\infty}^{+\infty} (y^p \cdot pdf(y))dy. \tag{4.2}$$

where $pdf(y)$ is the PDF function of y and m_y^p is the p-th probabilistic moment of y.

For illustration purpose, we introduce a significant observations of AWE method [PR90] as below:

Property 1. *The low order moments are more important to achieve high accuracy when moments m_y^p $(p = 1, \cdots, +\infty)$ are used to recover $pdf(y)$.*

Proof. Let $\Phi(\omega)$ be the Fourier transform of $pdf(y)$ as (detailed derivation can be referred to [GYH11]):

$$\Phi(\omega) \;=\; \sum_{p=0}^{+\infty} \frac{(-j\omega)^p}{p!} \cdot m_y^p. \tag{4.3}$$

Equation (4.3) is equivalent to the Taylor expansion of $\Phi(\omega)$ at the expansion point $\omega = 0$ [LLG04], and the high order moments are related to the coefficients of the Taylor expansion. It is well-known that Taylor expansion linearizes the function around the expansion point and thus equation (4.3) provides high accuracy around $\omega = 0$. As such, the low order moments (around $\omega = 0$) are more important for the accuracy of both the expansion in (4.3) and approximated $pdf(y)$.

From another point of view [LLG04], the magnitude of moments (coefficients) reaches its maximum at $\omega = 0$ as $\Phi(0) = 1$ and decays as ω increases, which behaves as a low-pass filter. Therefore, the low frequency band is much more important for approximation accuracy which is mainly determined by low order moments (ω around 0). □

4.2.2 Moment Matching for PDF Calculation

We will briefly review the method to extract $pdf(y)$ with probabilistic moments $\{m_y^p\}$ [PR90, LLG04]. First, *"time moments"* for y can be defined as:

$$\widehat{m}_y^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} y^k \cdot pdf(y)dy. \tag{4.4}$$

It is clear that $\widehat{m}_y^k$ is different from m_y^k in (4.2) due to a scaling factor $(-1)^k/k!$. In the mean time, consider a linear time-invariant (LTI) system H whose time moments defined as [PR90]:

$$\widehat{m}_t^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} t^k \cdot h(t)dt. \tag{4.5}$$

where t is the time variable and $h(t)$ is the impulse response of the LTI system H.

One important observation is that impulse response $h(t)$ in (4.5) can be an *optimal approximation* to $pdf(y)$ in (4.4) if we treat t in (4.5) as y in (4.4) and make $\widehat{m}_t^k$ equal to $\widehat{m}_y^k$ (i.e. moment-matching technique).

Moreover, according to probability theory [PR90, PP01, DS11], the time moments in (4.5) can be further expressed as:

$$\widehat{m}_t^k = -\sum_{r=1}^{M} \frac{a_r}{b_r^{k+1}}. \tag{4.6}$$

where a_r and b_r $(r = 1, \cdots, M)$ are the residues and poles of this LTI system, respectively. As such, the impulse response of the LTI system can be evaluated as:

$$h(t) = \begin{cases} \sum_{r=1}^{M} a_r \cdot e^{b_r \cdot t} & (t \geq 0) \\ 0 & (t < 0) \end{cases} \tag{4.7}$$

The overall algorithm that calculates $h(t)$ as an optimal approximation to $pdf(y)$ can be summarized as follows:

Algorithm 4 Overall Algorithm for PDF Calculation

1: Input probabilistic distributions of variable parameters.
2: /* Step 1: Moment Calculation */
3: Calculate the time moments of observations as $\widehat{m}_y^k$ with (4.4) in the performance space.
4:
5: /* Step 2: Moment Matching */
6: Make $\widehat{m}_y^k$ equal to $\widehat{m}_t^k$ by matching first several moments.
7: Solve the resulting nonlinear equation system in (4.6) for residues a_r and poles b_r.
8:
9: /* Step 3: PDF Calculation */
10: Compute impulse response $h(t)$ in (4.7) with residues a_r and poles b_r.
11: Use $h(t)$ as the optimal approximation of $pdf(y)$.

The most challenging step is to evaluate high-order moments m_y^k and time moments $\widehat{m}_y^k$ in the performance space. In this chapter, we proposed a novel and efficient algorithm to evaluate these high-order moments without response surface model but with high accuracy.

4.3 High Order Moment Estimation

4.3.1 Moments via Point Estimation

Usually it is impractical to compute m_y^k as (4.2) because $pdf(y)$ is unknown. Instead, *"Point Estimation Method"* proposed in [Ros75, ZO00] approximates m_y^k by a weighted sum of several sampling values of y. For example, assume x is the only variable and x_j $(j = 1, \cdots, p)$ are estimating points of x with weights P_j, the

k-th order probabilistic moment of y can be approximated as:

$$m_y^k = \int_{-\infty}^{+\infty} y^k \cdot pdf(y)dy \approx \sum_{j=1}^{p} P_j \cdot y_j^k = \sum_{j=1}^{p} P_j \cdot f(x_j)^k. \tag{4.8}$$

The works in [Ros75] and [ZO00] only provide empirical analytical formulae of x_j and P_j for first four moments (e.g. the mean, the variance, the skewness, the kurtosis) and thus cannot satisfy the requirement of AWE where *higher* order moments (e.g. $k \gg 4$) are needed. To this end, an systematical approach has been established in [GYH11] to efficiently calculate the estimating points x_j and weights P_j for arbitrary order moments.

However, all these approaches [GYH11, Ros75, ZO00] can only handle simply low-dimensional problems. Therefore, it is significant but remains unknown how to evaluate high order moments for high-dimensional problems (e.g. tens or hundreds of variables) which is the motivation behind this work.

4.3.2 Basic Idea of Moments via Sampling Method

The integral of m_y^k in (4.2) is very difficult to compute because an analytical evaluation is not available. Therefore, it is inevitable to utilize a sampling method. In fact, *"Point Estimation Method"* is a sampling-like method, which picks a few "representative" samples y_j in (4.8) and weights them by P_j to approximate the integral value.

Inspired by this observation, our proposed method tries to choose a few samples as "good representatives" of the entire sampling space so that a huge number of samples can be saved. For example, Fig. 4.1 shows a probability density map consisting of two normal distributed variables. Note that only partial of the probability density map is plotted in order to show the interior "representative" sampling points.

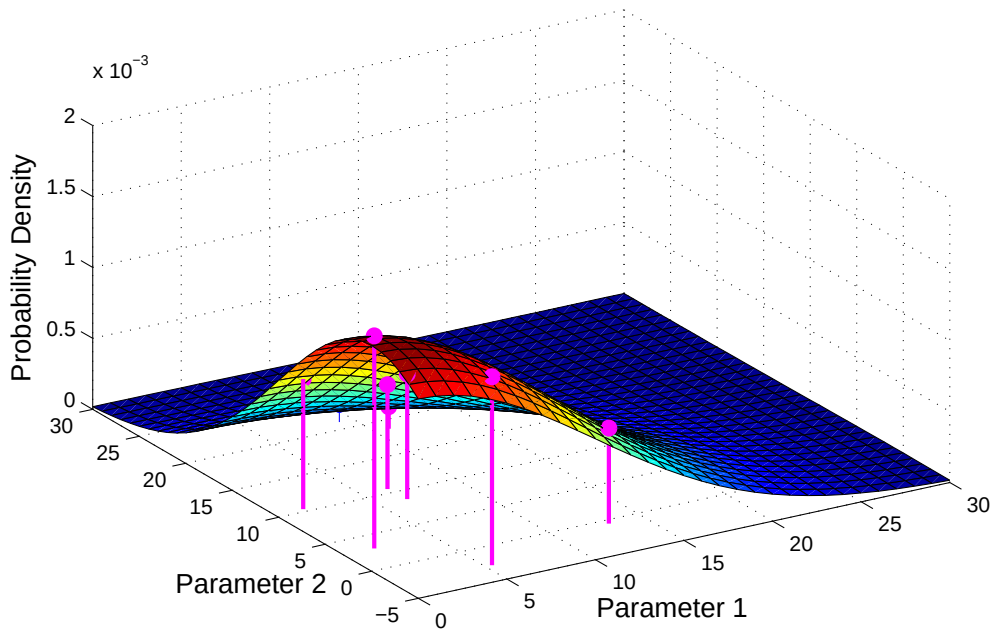


Figure 4.1: The probability density map and "representative" sampling points.

Since the integral of m_y^k in (4.2) is defined over the entire space shown in Fig. 4.1, the most straightforward sampling method is to generate as many samples as possible spreading over the "entire" sampling space which is infeasible due to unaffordable computational efforts.

Instead, the proposed method chooses a few "representative" samples $\vec{x}_j = [x_{1,j}, x_{2,j}, \cdots]$ shown as marked stems in Fig. 4.1 , which should satisfy below conditions:

- The samples of each element of $\vec{x}$ (e.g., x_i) should follow its known marginal distribution (e.g. $pdf(x_i)$).
- Various correlations and other relationships between the elements of $\vec{x}$ should remain intact.
- These chosen samples should fully cover the entire sampling space to provide closer approximation.

- These samples should be incoherent so that the number of required samples can be kept to be the minimum.

To meet above requirements, we propose to leverage well-established Latin Hypercube Sampling (LHS) method [Ste87] along with correlation control technique [IC82] as discussed in the following section.

4.3.3 Latin Hypercube Sampling and Correlation Control

4.3.3.1 Latin Hypercube Sampling

The Latin Hypercube Sampling (LHS) method [Ste87] is a widely used variant of Monte Carlo method which can "efficiently" generate samples. LHS method first divides the cumulative distribution of each random variable into several intervals with equal probability and picks one sample from each interval randomly. Then, LHS transforms these samples into the desired probabilistic distribution using inverse cumulative distribution function. As such, the samples for each variable can be paired randomly to generate LHS samples. Note that LHS is "efficient" because each random variable will be sampled only once from each of its intervals. Thereby, LHS method can use a small number of samples to ensure a full coverage of the sampling space.

For example, we plot some LHS samples in the probability density map consisting of two standard-normal distributed variables in Fig. 4.2 where all samples are dispersed over the entire parameter space and there is no duplicate/overlapped samples. In addition, all samples have been projected into a 1-dimension space in Fig. 4.3 which clearly demonstrates all samples follow the known marginal distribution $N(0, 1)$ and fully cover the entire sampling space. Moreover, it can be observed from Fig. 4.3 that there is no duplicate samples, which implies that any two different LHS samples have different values for the same random variable so that these two LHS samples are incoherent.

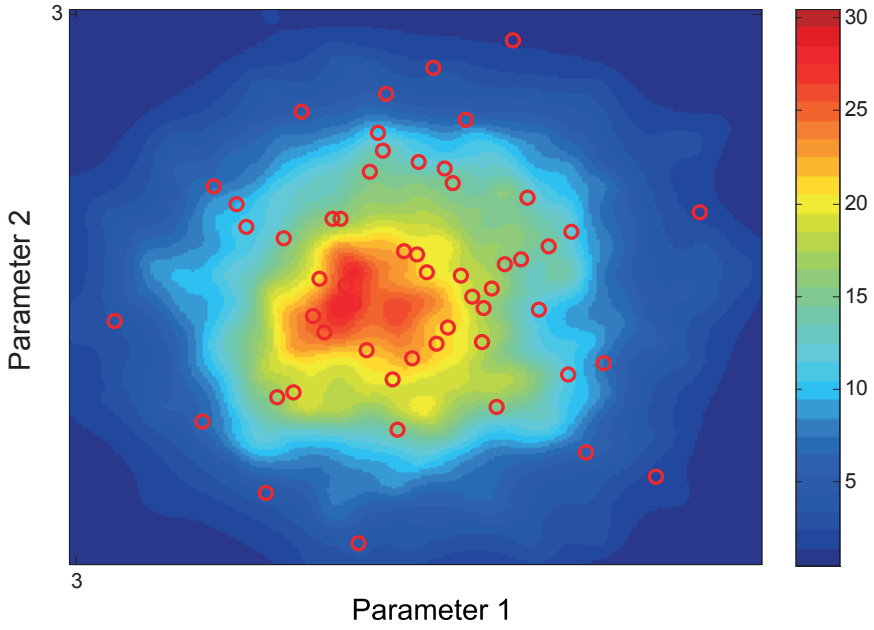


Figure 4.2: Probability Density Map and LHS Samples.

Therefore, LHS samples can meet all requirements except for the condition of correlation and thus correlation control technique [IC82] is needed.

4.3.3.2 Correlation Control Technique

The distributions of individual elements in $\vec{x}$ can be correlated which can be characterized by a correlation matrix C (e.g. entry $C_{ij} \in [-1, 1]$ is the correlation coefficient between x_i and x_j variables). As an illustrative example, we consider random variables in $\vec{x}$ to be *independent* where the target correlation matrix becomes an identity matrix.

The conventional sampling scheme is to generate samples for individual variable x_i independently and then pair them randomly (as combinations) to produce samples of $\vec{x}$. In particular, we are more interested in the case when sample-size N is small (e.g. N is in linear with the number of variables) and there is an important observation in this case:

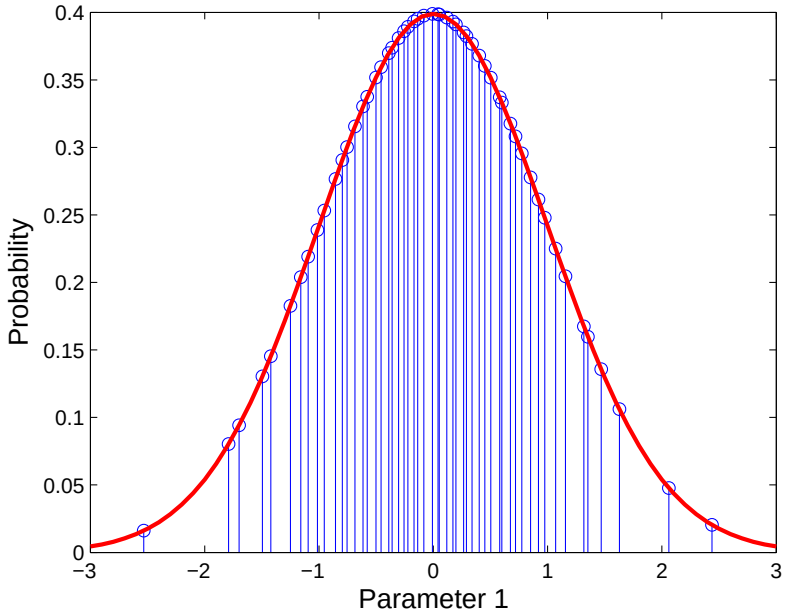


Figure 4.3: LHS samples in 1-Dimension.

Property 2. *The conventional sampling scheme can provide samples with desired correlation only if the sample-size is large enough. When sample-size is small, the generated samples have WRONG correlation relationship.*

Proof. The reason behind this observation is when sample-size is small, arbitrary correlations between individual elements of $\vec{x}$ can be introduced during the pairing/combination stage. However, this phenomena disappears as the sample-size increases, since large sample-size can ensure a close approximation to purely random combination.

As an example, we can consider two "independent" random variables which follows standard normal distribution and ideally the correlation matrix should be identity. We compare the correlation matrices with different number of samples as:

$$\underbrace{\begin{pmatrix} 1 & -0.0082 \\ -0.0082 & 1 \end{pmatrix}}_{100\ samples} \qquad \underbrace{\begin{pmatrix} 1 & -0.73 \\ -0.73 & 1 \end{pmatrix}}_{4\ samples}$$

Clearly, the correlation matrix extracted from 100 samples is closer to an identity matrix so that samples of two independent variables are obtained, while 4 samples introduce a strong correlation between these two independent random variables. □

Is there any way to fix the correlation issue of cases with small sample-size? The short answer is positive and we will now discuss the reasoning for this answer.

Since the incorrect correlation is introduced in the pairing/combination stage, it is a natural choice to "re-pair or re-combine" samples for different random variables to achieve the correct correlation and thus the correlation control technique developed by Iman and Conover [IC82] can be used. The theoretical basis can be briefly described as following:

Let us consider m random variables and the desired correlation matrix is C which is positive-definite and symmetric. First, n random samples can be generated for each variable and we can build a $n \times m$ matrix denoted as X, where X_{ij} is the i-the sample for the j-th variable. We assume R is the correlation matrix extracted from these n random samples, which would be different from C. Note that any positive-definite and symmetric matrix has Cholesky decompositions such as $C = PP^T$ and $R = QQ^T$ where P and Q are lower triangular matrices.

In principle, the procedure in [IC82] re-pairs the samples in X in order to obtain $\hat{X}$ that has the closest correlation matrix to C. To do so, it builds another $n \times m$ matrix K where each column has a random permutation of m van der Waerden scores (the inverse of the standard normal distribution [Con80]). In this way, $\hat{K} = K(Q^{-1})^T P^T$ has the closest correlation matrix to C. Then, the desired matrix $\hat{X}$ can be obtained by simply re-pair the samples in X in the same order as the samples in $\hat{K}$. Therefore, $\hat{X}$ has the same correlation matrix as $\hat{K}$ which is a close approximation to C.

For example, we apply the correlation control technique to the 4 samples for

two independent variables in above example and the correlation matrix becomes quite close to an identity matrix:

$$\underbrace{\begin{pmatrix} 1 & -0.73 \\ -0.73 & 1 \end{pmatrix}}_{before\ correlation\ control} \Rightarrow \underbrace{\begin{pmatrix} 1 & -0.098 \\ -0.098 & 1 \end{pmatrix}}_{after\ correlation\ control}$$

Therefore, the condition of correlation can also be satisfied.

Note we assume independent variables in this chapter for illustration purpose, however, the random process variables are spatially correlated in practice. Therefore, a correlation matrix extracted from the measurements is typically needed to generate the correlated samples.

In addition, it is worthwhile to point out that the correlation control technique [IC82] can only be applied to joint Normal distributions, since only second-order statistics is needed as the input (i.e., the covariance matrix).

4.3.4 Moments via Sampling Methods

Next, we need to approximately evaluate the integral in (4.2) with a small number of "representative" LHS samples. In particular, the k-th order probabilistic moment m_y^k can be estimated as:

$$m_y^k = E(y^k) = \int y^k \cdot pdf(y)dy \approx \frac{1}{N} \cdot \sum_{j=1}^{N} f(\vec{x}_j)^k. \tag{4.9}$$

where $\vec{x}_j (j = 1, \cdots, N)$ are the j-th samples of $\vec{x}$ using LHS method and correlation control technique. $f(\vec{x}_j)$ is the performance merit of the circuit with input $\vec{x}_j$. This approach is actually the sampled-form of the expectation value $E(y^k)$ and only utilizes these representative samples $f(\vec{x}_j)$.

4.3.5 Discussion of Proposed Methods

The proposed method has following positive features: (i) proposed method needs no response-surface-model, therefore, it avoids the exponential complexity and bias-variance tradeoff in the existing RSM models. (ii) LHS method is used to generate samples which is a variant of Monte Carlo method and intrinsically capable of handling high dimension problems efficiently.

In the meantime, proposed method has a major drawback: these methods pick a few samples as "representatives" of the entire sampling space, which implicitly implies that the neighborhood around each sampling points $\vec{x}_j$ has the similar circuit behavior $f(\vec{x}_j)$. This is a linearized assumption, thereby, more samples could be needed to accurately describe the strongly nonlinear circuit behavior in strongly nonlinear problems.

4.4 PDF/CDF Calculation with Moments

4.4.1 Normalized PDF for Error Prevention

The next step is to compute the residues a_r and the poles b_r in (4.6) with high order moments so that the impulse response $h(t)$ in (4.7) can be evaluated to approximate $pdf(y)$. Note that probabilistic moments m_y^k should be multiplied by a scaling factor to compute time moments in (4.4). As such, the equation (4.6) results in a nonlinear equation system as:

$$-\begin{bmatrix} \frac{a_1}{b_1}+\frac{a_2}{b_2}+\cdots\frac{a_M}{b_M} \\ \frac{a_1}{b_1^2}+\frac{a_2}{b_2^2}+\cdots\frac{a_M}{b_M^2} \\ \frac{a_1}{b_1^3}+\frac{a_2}{b_2^3}+\cdots\frac{a_M}{b_M^3} \\ \vdots \\ \frac{a_1}{b_1^{2M}}+\frac{a_2}{b_2^{2M}}+\cdots\frac{a_M}{b_M^{2M}} \end{bmatrix} = \begin{bmatrix} \widehat{m}_y^{\,0} \\ \widehat{m}_y^{\,1} \\ \widehat{m}_y^{\,2} \\ \vdots \\ \widehat{m}_y^{\,M} \end{bmatrix}. \tag{4.10}$$

This nonlinear system can be efficiently solved with the numerical solution presented in [PR90]. However, the calculated residues a_r and the poles b_r may suffer from numerical noises such as roundoff error. Therefore, we propose to normalize the PDF calculated from (4.7) to cancel out the roundoff error.

Let us denote $\widehat{m}_y^k$ as the *exact* value of k-th order time moment in (4.4), and $\tilde{m}_y^k$ as the *estimated* value of k-th order time moment. Also, we assume $\tilde{m}_y^k = const \cdot \widehat{m}_y^k$ due to roundoff error, where *const* is a scaling constant. As such, the right hand side of (4.10) should be substituted by $\tilde{m}_y^k$, which leads to $\tilde{a}_j = const \cdot a_j$ and a_j are exact values of the residues.

In order to eliminate the scaling constant in $\tilde{a}_j$, we propose to normalize $pdf(y)$ as follows: first, y can be discretized into several discrete points y_p, $(p = 1, \cdots, K)$. Then, the PDF value on p-th discrete point can be divided with the sum of PDF values for all discrete points as:

$$pdf_{norm}(y_p) = \frac{\sum_{r=1}^{M} \cancel{const} \cdot a_r \cdot e^{\widehat{b}_r^{k+1} \cdot y_p}}{\sum_{p=1}^{K} \sum_{r=1}^{M} \cancel{const} \cdot a_r \cdot e^{\widehat{b}_r^{k+1} \cdot y_p}}. \tag{4.11}$$

In this way, the scaling constant can be canceled out and thus the normalization procedure improves the numerical stability of proposed algorithm.

4.4.2 Error Estimation

It is significant to estimate the the approximation error of $pdf(y)$ using AWE method [PR90] but *exact* PDF is usually not available. Instead, we consider the approximation with first $q + 1$ order moments as the exact value and use the relative error of $\Phi(\omega)$ (i.e. the Fourier transform of $pdf(y)$ in (4.3)) to measure the accuracy of PDF approximation using first q order moments.

$$
\begin{aligned}
Error &= \left| \frac{\Phi^{q+1}(\omega) - \Phi^{q}(\omega)}{\Phi^{q+1}(\omega)} \right| \\
&= \left| \frac{(-j\omega)^{q+1}}{(q+1)!} \cdot \left(\sum_{p=0}^{q+1} \frac{(-j\omega)^{p}}{p!} \cdot \frac{m_y^p}{m_y^{q+1}} \right)^{-1} \right|.
\end{aligned} \tag{4.12}
$$

When $|m_y^p| \geq |m_y^{q+1}|$ ($p \leq q+1$), above estimation error has upper bound:

$$
Error \leq \left| \frac{(-j\omega)^{q+1}}{(q+1)!} \cdot \left(\sum_{p=0}^{q+1} \frac{(-j\omega)^{p}}{p!} \right)^{-1} \right|. \tag{4.13}
$$

Above error analysis results in an important observation as:

Property 3. *Assume high order moments $m_y^k (k = 1, \cdots, N)$ are used to predict pdf(y). When the moments m_y^k decay as the moment order increases, the approximation of pdf(y) has upper error bound.*

In this work, we consider a much *stronger* condition $y \in [0, 1]$ with the help of linear transformations of y (e.g. scaling, shifting, etc.). As such, the approximation of PDF/CDF can provide high accuracy with high order moments.

4.5 Overall Algorithm

4.5.1 Algorithm Flow

The overall algorithm flow for PDF/CDF approximation has been summarized in Algorithm (5).

4.5.2 Implementation Details

We briefly discuss several implementation issues as below:

- **Linear Transformation**: To ensure an upper error bound exists, we pro-

Algorithm 5 Overall Proposed Algorithm

Input: random variables $\vec{x} = (x_1, \cdots, x_M)$ with known probabilistic distributions and correlation matrix C.
Output: the estimated PDF/CDF of circuit behavior y.

```
/* Step 1: High Order Moments Calculation */
Use Latin Hypercube Sampling method to generate N samples x_j (j = 1, ..., N).
Re-pair these samples with correlation control technique to achieve the correlation matrix C.
Run SPICE simulations on these samples for corresponding circuit behavior y_j (j = 1, ..., N).
Compute moments m_y^k with y_j as (4.9).

/* Step 2: Moment Matching */
Calculate the time moments m̂_y^k with m_y^k as (4.4).
Make m̂_y^k equal to m̂_t^k by matching first several moments.
Solve the resulting nonlinear equation system in (4.10) for residues a_r and poles b_r.

/* Step 3: PDF Calculation */
Compute normalized PDF/CDF of y with a_r and b_r.
```

pose to transform y into the interval $[0, 1]$ so that moments can decay as the moment order increases. In general, the transformations include scaling, shifting, flipping and other linear operations. Note that the extracted PDF/CDF should be converted back to be the real results.

- **Numerical Instability:** In principle, more high order moments can improve the approximation accuracy of PDF/CDF. Unfortunately, it is not true because the moments can decay dramatically and be close to zero when the moment order increases. Therefore, the inevitable numerical noise (e.g., ill-conditioned moment matrix) prevents further accuracy improvement. This is an intrinsic drawback of the moment-matching method [PR90].

- **PDF/CDF Shifting:** The approximated PDF/CDF of y can be far from the real location and display a large delay in the time domain [LLG04].

Therefore, the PDF/CDF shifting technique using the modified Chebyshev inequality in [LLG04] can be used to ensure the PDF/CDF approximations match the MC results.

4.6 Experimental Results

The proposed algorithm has been implemented in MATLAB environment with HSPICE and BSIM4 transistor model. We use a two-stage operational amplifier and a SRAM bit-cell to demonstrate the accuracy and efficiency of proposed algorithm on both AC and transient performance merits by comparing against Monte Carlo method. In order to introduce process variations to these circuits, we consider 9 process parameters for each transistor shown in Table 4.1 which are physically independent parameters [DM03] and modeled as independent Gaussian random variables.

Table 4.1: Process Parameters of MOSFETs.

Variable Name	σ/μ	unit
Flat-band Voltage (V_{fb})	0.1	V
Gate Oxide Thickness (t_{ox})	0.05	m
Mobility (μ_0)	0.1	m^2/Vs
Doping concentration at depletion (N_{dep})	0.1	cm^{-3}
Channel-length offset (ΔL)	0.05	m
Channel-width offset (ΔW)	0.05	m
Source/drain sheet resistance (R_{sh})	0.1	Ohm/mm^2
Source-gate overlap unit capacitance (C_{gso})	0.1	F/m
Drain-gate overlap unit capacitance (C_{gdo})	0.1	F/m

For comparison purpose, all experiments involve three different approaches as following:

(1) **Monte Carlo method**: Calculate the probabilistic distributions (PDF and CDF) from a huge number of Monte Carlo samples. This is the "gold standard" for the comparison in this section.

(2) **MC+Moment Matching method**: The probabilistic distributions are

computed with moment matching method [PR90, LLG04] where the moments are evaluated using Monte Carlo samples.

(3) **Proposed Method**: The moments are calculated with (4.9) using a few samples from the LHS method coupled with correlation control technique. Also, the probabilistic distributions of circuit performance are approximated using the moment-matching method in [PR90, LLG04].

4.6.1 6-T SRAM Bit-Cell

Let us first study a typical design of 6-transistor SRAM bit-cell as shown in Fig. 4.4 [WYL09], which stores one memory bit and consists of six transistors: the four transistors $Mn1$, $Mn3$, $Mp5$ and $Mp6$ forms two inverters to keep either a logic '0' or '1'. Two additional access transistors $Mn2$ and $Mn4$ controls the access to the bit-cell during read and write operations. The word line WL is used to determine whether the bit-cell should be accessed (connected to bit lines) and the bit lines (BL and $\bar{BL}$) are used to read/write the actual data from/to the cell.

To model the process variations, we introduce random variations to 9 process parameters shown in Table 4.1 of each transistor, which implies totally 54 independent random variables in this example.

As an illustration, we investigate the discharge behavior on $\bar{BL}$ during the reading operation when Q node stores 1. In details, both $\bar{BL}$ and BL are first pre-charged to Vdd and then $\bar{BL}$ starts to discharge when the word line WL becomes high. When the voltage difference Δv between $\bar{BL}$ and BL becomes large, Δv can be sensed by the sense amplifier connected to both $\bar{BL}$ and BL.

However, the process variations can significantly change the discharge behavior and, particularly, a reading failure can happen when Δv is too small to be sensed by the sense amplifier at the end of reading operation. Therefore, it is of great interests to study the probabilistic distribution of node voltage $\bar{BL}$ at the end time

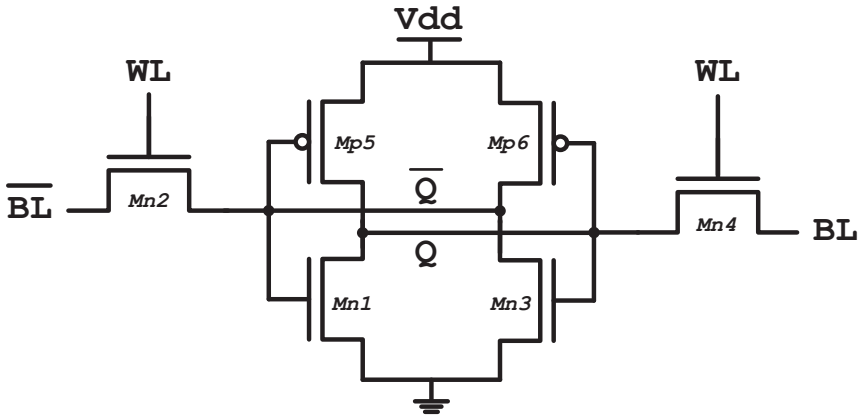


Figure 4.4: Schematic of a 6-T SRAM bit-cell.

step of reading operation considering the process variations. Note that the reading failure of SRAM bit-cell is a "rare event" with extremely small probability [Den01] that is *not* in the scope of this work, while the overall stochastic discharge behavior in this SRAM cell will be studied.

4.6.1.1 Comparison of Moment Calculation

Before we move forward to the extracted probabilistic distributions of the circuit behavior, let us study the accuracy of moments evaluations which significantly affects the accuracy of probabilistic distributions. In details, we use two different methods (e.g., MC and proposed method) to calculate the first ten moments and show the results in Table 4.2. Here, the moments from MC with 1E+5 samples serves as the "exact" moment values.

We have some observations from this table: first, the proposed method provides accurate evaluations of high order moments (i.e., $\leq 6\%$ relative error); second, the proposed method, in general, incurs increasingly large error in moments calculation as the order increases, however, low order moments are more important to the accuracy of extracted probabilistic distributions due to Property (1). Therefore, the proposed method achieves high accuracy in circuit behavior distribution with

Table 4.2: Comparison of First Ten Probabilistic Moments

Moment Order	Monte Carlo (1E+5 samples)	Proposed (54 samples)
0	1.000E+00 (0%)	1.000E+00 (0.00%)
1	1.611E-01 (0%)	1.623E-01 (+0.74%)
2	2.626E-02 (0%)	2.710E-02 (+3.19%)
3	4.331E-03 (0%)	4.436E-03 (+2.42%)
4	7.230E-04 (0%)	7.015E-04 (-2.97%)
5	1.221E-04 (0%)	1.262E-04 (+3.35%)
6	2.089E-05 (0%)	2.127E-05 (+1.81%)
7	3.620E-06 (0%)	3.596E-06 (-0.66%)
8	6.352E-07 (0%)	6.500E-07 (+2.32%)
9	1.129E-07 (0%)	1.198E-07 (-6.00%)

these moments as demonstrated in the following.

4.6.1.2 Comparison of PDF/CDF Approximation

With the moments available, the moment matching method is used to predict the probabilistic distributions (PDF and CDF) of circuit behavior. We have applied all different methods (i.e., MC, MC+Moment Matching, Proposed method) to this SRAM bit-cell example and plotted their approximations of PDFs and CDFs with first 20 moments in Fig. 4.5 and Fig. 4.6, respectively. Note that we use kernel density estimation method [BA97] to estimate the PDF using 1E+5 MC samples and then analytically integrate the PDF to get CDF.

For comparison purpose, we plotted PDF and CDF from proposed method using first ten moments in the same figures, which have significant accuracy loss and only provide *rough* approximations. When the approximation order increases to 20, the PDF and CDF estimations are clearly improved and closely match with MC results except for the tail regions.

In addition, the PDF from the proposed method (the curve with circle marks) contains numerical oscillations in the tails. In fact, the similar oscillations are demonstrated in the PDF from the MC+Moment Matching method (the

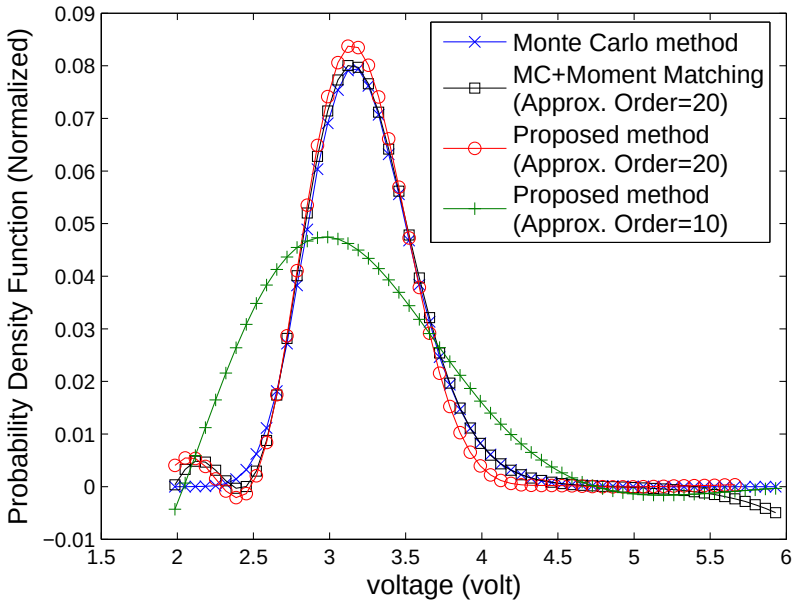


Figure 4.5: PDF approximation from proposed method for SRAM bit-cell example.

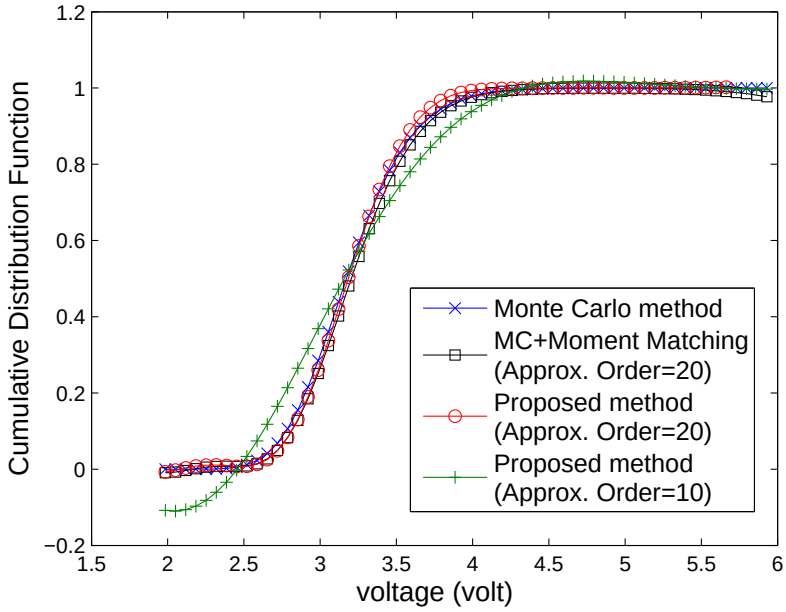


Figure 4.6: CDF approximation from proposed method for SRAM bit-cell example.

curve with square marks) where the "exact" moment values from MC samples are used. In principle, the approximated PDF should eliminate these oscillations and asymptotically approach the exact PDF when the approximation order increases. However, the moment matrix which is used to calculate the residues a_r and poles b_r in (4.6) becomes severely ill-conditioned and leads to inaccurate and unstable results of a_r and b_r. The numerical instability issue of moment-matching method [PR90] prevents further accuracy improvement with more probabilistic moments and remains a challenging issue.

Moreover, the approximation accuracy can be quantitatively characterized by the average error over several specific points of the CDF. The proposed method achieves 5.03% relative error on average when compared with the CDF from MC samples.

4.6.1.3 Comparison of Efficiency

We study the efficiency of different methods in Table 4.3 where the CDF from MC method with 1E+5 samples serves as the exact CDF. In addition, the efficiency is measured by the number of required samples which equals to the number of SPICE simulations, since the transistor-level simulations are the most time-consuming calculations. Note that the CDF from MC with 1E+5 samples serves as the exact CDF. Also, the accuracy of CDF approximation is measured by the average accuracy over several specific points of the CDF.

In order to provide fair comparison, we incrementally add MC samples to provide the same accuracy as the proposed method. In fact, the MC method with 8.6E+3 samples offers 95% accuracy on average, therefore, the proposed method is 159X faster than MC method for the same accuracy.

Table 4.3: Efficiency Comparison of CDF Approximations.

	MC method	MC method	Proposed method
accuracy	100%	95%	95%
#samples	1E+5	8.6E+3 (159X)	54 (1X)

4.6.2 Operational Amplifier

We have validated the proposed method on SRAM bit-cell example which involves *transient* performance merits (e.g., voltage discharge in time domain) and Gaussian-like probabilistic distribution. Next, an operational amplifier (OPAMP), shown in Fig. 4.7, is used to study the proposed method on AC performance of merits (e.g. bandwidth) and non-Gaussian behavioral distributions.

In the OPAMP, vb_n, vb_p are bias voltages for NMOS and PMOS devices, respectively. vfb is the feedback voltage set separately by common mode feedback block. The small triangular devices denote the gain boosting components. We introduce the variations to process parameters in the Table 4.1 for all transistors except transistors associated with vb_n, vb_p and vfb. As such, there exist totally 90 random variables to model the local random variations, which is a typical large-scale problem in practice.

The circuit behavior of the OPAMP is described by its *bandwidth*, therefore, we aim to extract the "arbitrary" unknown distributions (PDF and CDF) of bandwidth under process variations in this case. We applied all different methods on this example and compared their performance in the following.

4.6.2.1 Comparison of Moment Calculation

We first validate the moments evaluation of proposed method in Table 4.4. The moments from MC method with 5E+5 samples serves as the exact moment values for comparison purpose. From this table, the proposed method can estimate fist ten moments with high accuracy (i.e., $< 4\%$ relative error) using only 90 samples.

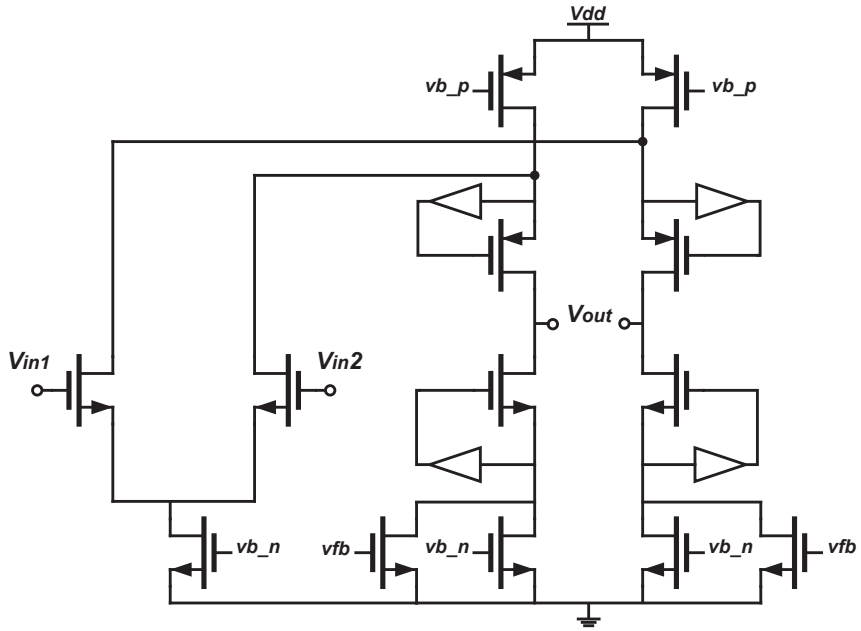


Figure 4.7: Simplified Schematic of Operational Amplifier

Similar to the observation in SRAM bit-cell example, the error of moment evaluations becomes increasingly larger as the order increases. However, the proposed method retains high accuracy in low order moments which are more important to the accuracy of PDF/CDF approximation, thereby, providing accurate PDF/CDF using these moments.

Table 4.4: Comparison of First Ten Probabilistic Moments

Moment Order	Monte Carlo (5E+5 samples)	Proposed method (90 samples)
0	1.000E+00 (0%)	1.000E+00 (0.00%)
1	1.913E-01 (0%)	1.936E-01 (+1.19%)
2	5.825E-02 (0%)	5.961E-02 (+2.28%)
3	2.266E-02 (0%)	2.308E-02 (+1.80%)
4	9.990E-03 (0%)	1.006E-03 (+0.77%)
5	4.739E-03 (0%)	4.737E-03 (-0.04%)
6	2.362E-03 (0%)	2.355E-03 (-0.28%)
7	1.220E-03 (0%)	1.222E-03 (+0.20%)
8	6.484E-04 (0%)	6.579E-04 (+1.43%)
9	3.524E-04 (0%)	3.646E-04 (+3.35%)

4.6.2.2 Comparison of PDF/CDF Approximation

We plot the approximated PDF and CDF from all different methods in Fig. 4.8 and Fig. 4.9, respectively. The PDF from MC method is estimated using kernel density estimation method [BA97] with 5E+5 samples and is the "exact" PDF of bandwidth. For comparison purpose, the PDF from proposed method with first 4 moments is plotted in the same figure, which clearly deviates from the exact PDF from MC method (the curve with cross marks) by a large amount.

We further increase the approximation order to 12 and the extracted PDF (the curve with circle marks) becomes much closer to the exact PDF from MC. However, there are some numerical oscillations in the PDFs from moment matching based methods (i.e., proposed method and the MC+Moment Matching method), which result from the numerical noise during moment matching.

The approximation accuracy is measured by the average error over several specific points of the CDF. In this OPAMP case, the proposed method achieves 1.65% relative error on average when compared with the exact CDF from MC method.

4.6.2.3 Comparison of Efficiency

We use the number of required samples to measure the efficiency between different methods in Table 4.5. The CDF from MC method with 5E+5 samples is treated as the *exact* CDF. In addition, the accuracy of CDF approximation is measured by the average accuracy over several specific points of the CDF. From this table, we can observe that the proposed method uses 90 samples to provide 98% accuracy while MC method needs 1.5E+5 samples for the same accuracy. It implies the proposed method offers nearly linear complexity and is $1666X$ faster than MC method.

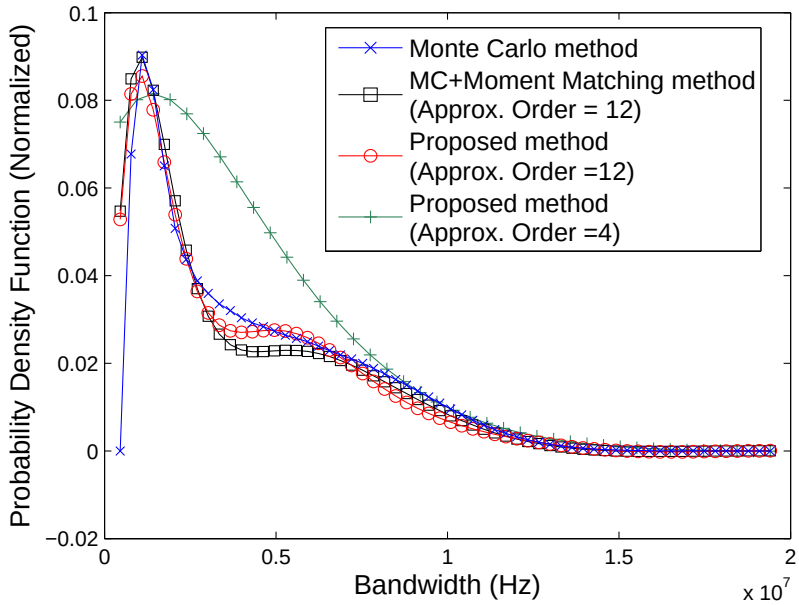


Figure 4.8: PDF approximation from proposed method for OPAMP example.

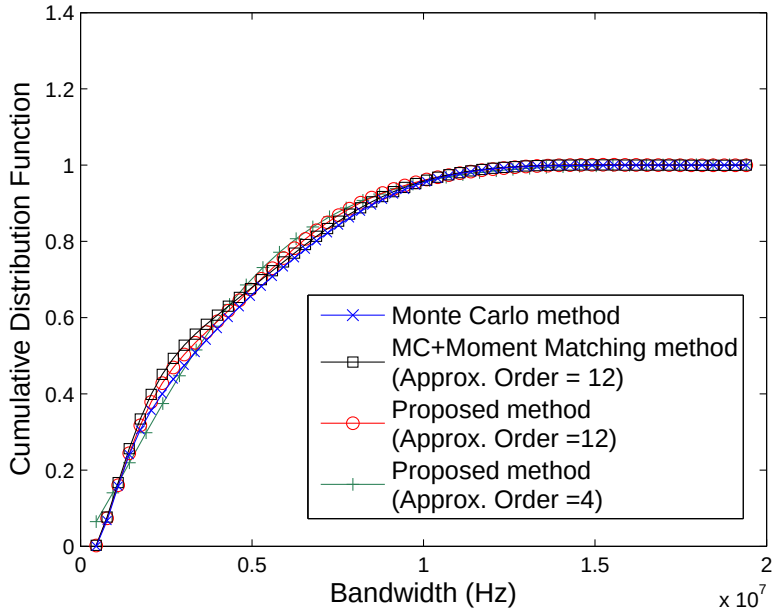


Figure 4.9: CDF approximation from proposed method for OPAMP example.

Table 4.5: Efficiency Comparison of CDF Approximations.

	MC method	MC method	Proposed method
accuracy	100%	98%	98%
#samples	5E+5	1.5E+5 (1666X)	90 (1X)

4.7 Conclusion

In this chapter, we have proposed an efficient moment-matching based algorithm to extract the arbitrary probabilistic distribution of stochastic circuit behavior. Our approach can perform an efficient evaluation of high-order moments of circuit behavior and thus circumvent the use of response surface models (RSM). Moreover, the proposed method has been successfully extended to deal with high dimension problems with nearly linear complexity. Experiments show that the proposed method can provide up to 1666X runtime speedup with the same accuracy when compared to the Monte Carlo method.

The work presented in this chapter has three-fold limitations: first, the deployed circuit examples are relatively small when compared to industrial designs, where 1000+ random variables can be easily involved. Second, most process variation sources are spatially correlated in practice but we assume independent random variables for illustration purpose. Third, the conventional moment matching method suffers from numerical noise and instability issues which remains a significant challenge. Our future study will investigate the correlated process variation sources, study large-scale industrial problems and deal with the numerical noise issues.

CHAPTER 5

Parallel and Variability-Aware Capacitance Extraction

5.1 Introduction

As IC designs are approaching processes below 45nm, there exist large uncertainties from chemical mechanical polishing (CMP), etching, and lithography [LNP00, CCS04, ZW05, ZZC07, CCS08]. As a result, the fabricated interconnect and dielectric can show a significant difference from the nominal shape. The value of an extracted capacitance can differ from the nominal value by a large margin, which may further lead to significant variability for timing analysis. For example, as shown in [LNP00], variation of interconnects can cause as much as 25% variation in the clock skew. Therefore, accurately extracting capacitance while considering the stochastic process variation becomes a necessity.

To avoid discretizing the entire space, the boundary element method (BEM) is used to evaluate capacitance by discretizing the surface into panels on the boundary of the conductor and the dielectric [NW91, SD97, SLK98, YLS07]. Though this results in a discretized system with a small dimension, the discretized system under BEM is dense. FastCap [NW91] solves such a dense system by a generalized minimal residual (GMRES) method. Instead of performing the expensive LU decomposition, GMRES iteratively reaches the solution with the use of the matrix-vector multiplication. The computational cost of the matrix-vector-product (MVP) can be reduced by either a fast-multipole-method (FMM) [NW91],

a low-rank approximation [SD97], and a hierarchical-tree decomposition [SLK98]. As a result, the complexity of the fast full-chip extractions generally comes from two parts: the evaluation of MVP and the preconditioned GMRES iteration.

A few recent works [ZW05,ZZC07,CCS08] discuss interconnect extraction considering process variation. The variation is represented by the stochastic orthogonal polynomial (SOP) [XK02,VWG06] when calculating a variational capacitance. Since the interconnect length and cross-area are at different scales, the variational capacitance extraction is quite different between the on-chip [ZZC07, CCS08] and the off-chip [ZW05]. The on-chip interconnect variation from the geometrical parameters, such as width and length of one panel and distance between two panels, is more dominant [ZZC07,CCS08] than the rough surface effect seen from the off-chip package trace. However, it is unknown how to leverage the stochastic process variation into the MVP by FMM [ZW05, ZZC07, CCS08]. Similar to the issue of stochastic analog mismatch for transistors [PDW89], a cost-efficient full-chip extraction needs to explore an explicit relation between the stochastic variation and the geometrical parameter such that the electrical property can show an explicit dependence on geometrical parameters. Moreover, the expansion by SOP with different collocation schemes [XK02,VWG06,ZZC07,CCS08] always results in an augmented and dense system equation. This significantly increases the complexity when dealing with a large-scale problem. The according GMRES thereby needs to be designed in an incremental fashion to consider the update from the process variation. As a result, a scalable extraction algorithm similar to [NW91,SD97,SLK98] is required to consider the process variation with the new MVP and GMRES developed accordingly as well.

To address the aforementioned challenges, this chapter contributes as follows. First, to reveal an explicit dependence on geometrical parameters, the potential interaction is represented by a number of geometrical moments. As such, the process variation can be further included by expanding the geometrical moments with

use of stochastic orthogonal polynomials, called *stochastic geometrical moments* in this chapter. Next, with the use of the stochastic geometric moment, the process variation can be incorporated into a modified FMM algorithm that evaluates the MVP in parallel. Finally, an incremental GMRES method is introduced to update the preconditioner with different variations. Such a parallel and incremental full chip capacitance extraction considering the stochastic variation is called *piCAP*. Parallel and incremental analysis are the two effective techniques in reducing computational cost. Experiments show that our method with stochastic polynomial expansion is hundreds of times faster than the Monte-Carlo based method while maintaining a similar accuracy. Moreover, the parallel MVP in our method is up to 3X faster than the serial method, and the incremental GMRES in our method is up to 15X faster than non-incremental GMRES methods.

The rest of the chapter is organized in the following manner. We first review the background of the capacitance extraction and fast multipole method (FMM) in Section 5.2. We introduce the concept of the stochastic geometrical moment in Section 5.3, and illustrate a parallel FMM method based on the stochastic geometrical moment in Section 5.4. We further propose a novel incremental GMRES method in Section 5.5 and present experimental results in Section 5.6. Finally, the chapter is concluded in Section 5.7.

5.2 Background

5.2.1 Boundary Element Method (BEM)

The boundary element method (BEM), used in most fast capacitance extractions [NW91, SD97, SLK98], starts with an integral equation

$$\phi(r) = \int_{r' \in a'} \frac{\rho(r')}{4\pi\epsilon_0 |r - r'|} da', \tag{5.1}$$

where $\phi(r)$ is the potential at the observer metal, $\rho(r')$ is the surface-charge density at the source metal, da' is an incremental area at the surface of the source metal, and the source r' is on da'.

By discretizing the metal surface into N panels sufficiently such that the charge-density is uniform at each panel, a linear system equation can be obtained by the *point-collocation* [NW91]:

$$Pq = b, \tag{5.2}$$

where P is an $N \times N$ matrix of potential coefficients (or potential interactions), q is an N vector of panel charges, and b is an N vector of panel potentials. By probing b iteratively with one volt at each panel in the form of [0,...,1,...,0], the solved vector q is one column of the capacitance matrix.

Note that each entry P_{ij} in the potential matrix P represents the potential observed at the *observer panel* a_j due to the charge at the *source panel* a_i:

$$P_{ij} = \frac{1}{a_i} \int_{r_i \in a_i} \frac{1}{4\pi\epsilon_0 |r_i - r_j|} da_i. \tag{5.3}$$

When panel i and panel j are well-separated by definition, P_{ij} can be well approximated by $\frac{1}{4\pi\epsilon_0 |r_i - r_j|}$ [NW91, SD97, SLK98, ZZC07, CCS08].

The resulting potential coefficient matrix P is usually dense in the BEM method. As such, directly solving (5.2) would be computationally expensive. FastCap [NW91] applies an iterative GMRES method [SS86] to solve (5.2). Instead of performing an expensive LU decomposition of the dense P, GMRES first forms a preconditioner W such that $W^{-1} \cdot P$ has a smaller condition number than P, which can accelerate the convergence of iterative solvers [Saa03]. Take the left

preconditioning as an example:

$$(W^{-1} \cdot P)q = W^{-1} \cdot b.$$

Then, using either multipole-expansion [NW91], low-rank approximation [S-D97] or the hierarchical-tree method [SLK98] to efficiently evaluate the matrix-vector-product (MVP) for $(W^{-1} \cdot P)q_i$ (q_i is the solution for i-th iteration), the GMRES method minimizes below residue error iteratively till converged.

$$min: \ ||W^{-1} \cdot b - (W^{-1} \cdot P)q_i||$$

Clearly, GMRES requires a well-designed preconditioner and a fast matrix-vector-product (MVP). In fact, fast multipole method (FMM) is able to accelerate the evaluation of MVP with $O(N)$ time complexity where N is the number of variables. We will introduce FMM first as what follows.

5.2.2 Fast Multipole Method (FMM)

The fast multipole method was initially proposed to speed up the evaluation of long-ranged particle forces in the N-body problem [WS93, Ran99]. It can also be applied to the iterative solvers by accelerating calculation of matrix-vector-product [NW91]. Let's take the capacitance extraction problem as an example to introduce the operations in the FMM. In general, the FMM discretizes the conductor surface into panels and forms a cube with a finite height containing a number of panels. Then, it builds a hierarchical oct-tree of cubes and evaluates the potential interaction P at different levels.

Specifically, the FMM first assigns all panels to leaf cells/cubes, and computes the multipole expansions for all panels in each leaf cell. Then, FMM calculates the multipole expansion of each parent cell using the expansions of its children

cells (called M2M operations in Upward Pass). Next, the local field expansions of the parent cells can be obtained by adding multipole expansions of well-separated parent cells at the same levels (called M2L operations). After that, FMM descends the tree structure to calculate the local field expansion of each panel based on the local expansion of its parent cell (called L2L in Downward Pass). All these operations are illustrated within Fig. 5.1.

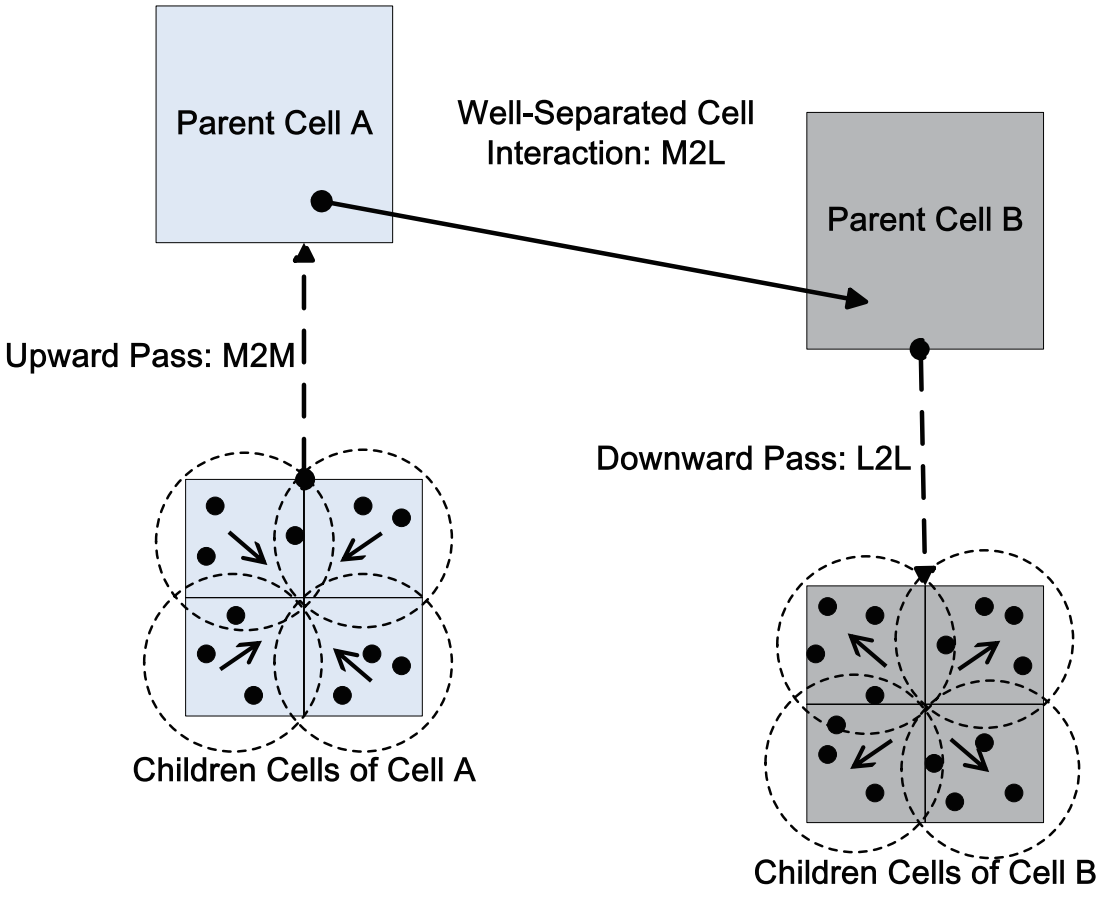


Figure 5.1: Multipole Operations Within the FMM Algorithm

In order to further speed up the evaluation of MVP, our stochastic extraction has a parallel evaluation Pq with variations, which is discussed in Section 5.4, and an incremental preconditioner, which is discussed in Section 5.5. Both of these features depend on how an explicit dependence between the stochastic process variation and the geometric parameters can be found, which will be discussed in

Section 5.3.

5.3 Stochastic Geometrical Moment (SGM)

With Fast Multipole Method, the complexity of MVP Pq evaluation can be reduced to $O(N)$ during the GMRES iteration. Since the spatial decomposition in FMM is geometrically dependent, it is helpful to express P using geometrical moments with an explicit geometry-dependence. As a result, this can lead to an efficient recursive update (M2M, M2L, L2L) of P on the oct-tree. The geometry-dependence is also one key property to preserve in presence of the stochastic variation. In this section, we first derive the geometrical moment and then expand it by stochastic orthogonal polynomials to calculate the potential interaction with variations.

5.3.1 Geometrical Moment

In this chapter, we focus on local random variations, or stochastic variations. Without loss of generality, two primary geometrical parameters with stochastic variation are considered for illustration purpose: panel-distance (d) and panel-width (h). Due to the local random variation, the width of the discretized panel, as well as the distance between panels, may show random deviations from the nominal value. With expansions in Cartesian coordinates, we can relate the potential interaction with the geometry parameter through *geometrical moments* (GMs) that can be extended to consider stochastic variations.

Let the center of an observer-cube be r_0 and the center of a source-cube be r_c. We assume that the distance between the i-th source-panel and r_c is a vector $\mathbf{r}$

$$\mathbf{r} = r_x \overrightarrow{x} + r_y \overrightarrow{y} + r_z \overrightarrow{z}$$

with $|\mathbf{r}| = r$, and the distance between r_0 and r_c is a vector $\mathbf{d}$

$$\mathbf{d} = d_x \overrightarrow{x} + d_y \overrightarrow{y} + d_z \overrightarrow{z}$$

with $|\mathbf{d}| = d$.

In Cartesian coordinates $(x - y - z)$, when the observer is outside the source region $(d > r)$, a *multipole expansion* (ME) [Jac75, Bra04] can be defined as

$$\begin{aligned} \frac{1}{|\mathbf{r} - \mathbf{d}|} &= \sum_{p=0} \frac{(-1)^p}{p!} (\underbrace{\mathbf{r} \cdots \mathbf{r}}_{p}) \underbrace{\times \cdots \times}_{p} (\underbrace{\nabla \cdots \nabla}_{p} \frac{1}{d}) \\ &= \sum_{p=0} M_p = \sum_{p=0} l_p(d) m_p(r) \end{aligned} \tag{5.4}$$

by expanding r around r_c, where

$$\begin{aligned} & l_0(d) = \frac{1}{d},\ m_0(r) = 1 \\ & l_1(d) = \frac{d_k}{d^3},\ m_1(r) = -r_k \\ & l_2(d) = \frac{3 d_k d_l}{d^5},\ m_2(r) = \frac{1}{6}(3 r_k r_l - \delta_{kl} r^2) \\ & \ldots \\ & l_p(d) = \underbrace{\nabla \cdots \nabla}_{p} \frac{1}{d},\ m_p(r) = \frac{(-1)^p}{p!} (\underbrace{\mathbf{r} \cdots \mathbf{r}}_{p}). \end{aligned} \tag{5.5}$$

Note that d_k, d_l are the coordinate components of vector $\mathbf{r}$ in Cartesian coordinates. The same is true for r_k and r_l. ∇ is the Laplace operator to take the spatial difference, δ_{kl} is the Kronecker delta function, and $(\mathbf{r} \cdots \mathbf{r})$ and $(\nabla \cdots \nabla \frac{1}{d})$ are rank-p tensors with $x^\alpha, y^\beta, z^\gamma$ $(\alpha + \beta + \gamma = p)$ components.

Assume that there is a spatial shift at the source-cubic center r_c for example, change one child's center to its parent's center by $\mathbf{h}$ $(|h| = c \cdot h)$, where c is a constant and h is the panel width. This leads to the following transformation for

m_p in (5.5)

$$\begin{aligned} m'_p &= \underbrace{((\mathbf{r}+\mathbf{h})\cdots(\mathbf{r}+\mathbf{h})}_{p}) \\ &= m_p + \sum_{q=0}^{p} \frac{p!}{q!(p-q)!} (\underbrace{\mathbf{h}\cdots\mathbf{h}}_{j}) m_{p-j}. \end{aligned} \tag{5.6}$$

Moreover, when the observer is inside the source region ($d < r$), a *local expansion* (LE) under Cartesian coordinates is simply achieved by exchanging d and h in (5.4)

$$\frac{1}{|\mathbf{r}-\mathbf{h}|} = \sum_{p=0} L_p = \sum_{p=0} m_p(h) l_p(r). \tag{5.7}$$

Also, when there is a spatial shift of the observer-cubic center r_0, the shift of moments $l_p(r)$ can be derived similarly to (5.6).

Clearly, both M_p, L_p and their spatial shifts show an explicit dependence on the panel-width h and panel-distance d. For this reason, we call M_p and L_p *geometrical moments*. As such, we can also express the potential coefficient

$$4\pi\epsilon_0 \cdot P(h,d) \simeq \begin{cases} \sum_{p=0} M_p & if\ d > r \\ \sum_{p=0} L_p & otherwise \end{cases} \tag{5.8}$$

as a geometrical-dependence function $P(h,d)$ via geometrical moments.

Moreover, assuming that local random variations are described by two random variables. ξ_h for the panel-width h, and ξ_d for the panel-distance d, the stochastic forms of M_k and L_k become

$$\begin{aligned} \hat{M}_p(\xi_h,\xi_d) &= M_p(h_0 + h_1\xi_h, d_0 + d_1\xi_d) \\ \hat{L}_p(\xi_h,\xi_d) &= L_p(h_0 + h_1\xi_h, d_0 + d_1\xi_d) \end{aligned} \tag{5.9}$$

where h_0 and d_0 are the nominal values and h_1 as well as d_1 define the perturbation range (% of nominal). Similarly, the stochastic potential interaction becomes $\hat{P}(\xi_h, \xi_d)$.

5.3.2 Stochastic Orthogonal Polynomial (SOP) Expansion

By expanding the stochastic potential interaction $\hat{P}(\xi_h, \xi_d)$ with stochastic orthogonal polynomials (SOPs), we can further derive the *stochastic geometric moments* (SGMs) below.

Assuming that there is one random distribution ξ related to one stochastic geometric variation, its related stochastic orthogonal polynomial is $\Phi(\xi)$. For example, for a Gaussian random distribution, $\Phi_i(\xi)$ is a Hermite polynomial [XK02, VWG06]

$$\Phi(\xi) = [1,\ \xi,\ \ \xi^2 - 1, \cdots,]^T. \tag{5.10}$$

As such, we can get the n-th order expansion of a potential coefficient matrix with $n+1$ Hermite polynomials by

$$\begin{aligned} \hat{P}(\xi) &= P_0\Phi_0(\xi) + P_1\Phi_1(\xi) + \cdots + P_n\Phi_n(\xi) \\ &= \sum_{k=0}^{n} P_k\Phi_k(\xi). \end{aligned} \tag{5.11}$$

Accordingly, the charge-density $\hat{q}(\xi)$ becomes:

$$\hat{q}(\xi) = \sum_{j=0}^{n} q_j\Phi_k(\xi). \tag{5.12}$$

By applying an inner-product with $\Phi_k(\xi)(k = 0, 1, ..., n)$

$$\langle \Phi_k,\ \hat{P}(\xi)\hat{q}(\xi) - b\rangle = 0 \tag{5.13}$$

to minimize the residue, we can derive an augmented linear system equation

$$\mathcal{P} \times \mathcal{Q} = \mathcal{B}. \tag{5.14}$$

The augmented P is calculated by

$$\mathcal{P} = (W_0 \otimes P_0 + W_1 \otimes P_1 + \cdots + W_n \otimes P_n). \tag{5.15}$$

Note that $\otimes$ represents a tensor product, and

$$W_k = \begin{pmatrix} w_{k,0,0} & w_{k,0,1} & \cdots & w_{k,0,n} \\ w_{k,1,0} & w_{k,1,1} & \cdots & w_{k,1,n} \\ \vdots & \vdots & w_{k,i,j} & \vdots \\ w_{k,n,0} & w_{k,n,1} & \cdots & w_{k,n,n} \end{pmatrix},$$

where $w_{k,i,j} =< \Phi_k \Phi_i \Phi_j >$ is the inner product of Hermite polynomials Φ_k, Φ_i, and Φ_j.

In addition, the augmented $\mathcal{Q}$, $\mathcal{B}$ and b_i become

$$\mathcal{Q} = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix}, \ \mathcal{B} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}, b_i = \sum_{k=0}^{n} \sum_{j=0}^{n} w_{k,i,j} \times P_i \times q_j.$$

By further defining

$$\mathcal{P}_{i,j} = \sum_{k=0}^{n} w_{k,i,j} \cdot P_k,$$

The augmented system equation illustrated in Eq.(5.14) will have an explicit

block-structure as shown below

$$\begin{pmatrix} \mathcal{P}_{0,0} & \mathcal{P}_{0,1} & \cdots & \mathcal{P}_{0,n} \\ \mathcal{P}_{1,0} & \mathcal{P}_{1,1} & \cdots & \mathcal{P}_{1,n} \\ \vdots & \vdots & \mathcal{P}_{i,j} & \vdots \\ \mathcal{P}_{n,0} & \mathcal{P}_{n,1} & \cdots & \mathcal{P}_{n,n} \end{pmatrix} \times \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}. \tag{5.16}$$

We use $n = 1$ as an example to illustrate the above general expression. First, the potential coefficient matrix $\hat{P}$ can be expanded with the first two Hermite polynomials by

$$\hat{P}(\xi) = P_0\Phi_0(\xi) + P_1\Phi_1(\xi) = P_0 + P_1\xi.$$

Then, the W_k $(k = 0, 1)$ matrix becomes

$$W_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; W_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix},$$

and the newly augmented coefficient system can be written as

$$\begin{aligned} \mathcal{P} &= W_0 \otimes P_0 + W_1 \otimes P_1 \\ &= \begin{pmatrix} P_0 & 0 & 0 \\ 0 & P_0 & 0 \\ 0 & 0 & P_0 \end{pmatrix} + \begin{pmatrix} 0 & P_1 & 0 \\ P_1 & 0 & 2P_1 \\ 0 & 2P_1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} P_0 & P_1 & 0 \\ P_1 & P_0 & 2P_1 \\ 0 & 2P_1 & P_0 \end{pmatrix}. \end{aligned} \tag{5.17}$$

By solving q_0, q_1, $\cdots$ and q_n, the Hermite polynomial expansion of charge-density can be obtained. Especially, the mean and the variance can be obtained

from

$$E(q(\xi_d)) = q_0$$
$$Var(q(\xi_d)) = q_1^2 Var(\xi_d) + q_2^2 Var(\xi_d^2 - 1) = q_1^2 + 2q_2^2.$$

Considering that the dimension of $\hat{P}$ is further augmented, the complexity to solve the augmented system in Eq.(5.16) would be expensive. To mitigate this problem, we present a parallel FMM to reduce the cost of MVP evaluations in Section 5.4 and an incremental preconditioner to reduce the cost of GMRES evaluation in Section 5.5.

5.4 Parallel Fast Multipole Method with SGM

Although the parallel fast multipole method has been discussed before such as [YBZ03], the extension to deal with stochastic variation for capacitance extraction needs to be addressed in the content of stochastic geometric moments (SGMs). In the following, we illustrate the parallel FMM considering the process variation.

The first step of a parallel FMM evaluation is to hierarchically subdivide space in order to from the clusters of panels. This is accomplished by using a tree-structure to represent each subdivision. We assume that there are N panels at the finest (or bottom) level. Providing depth H, we build an oct-tree with $H = \lceil log_8 \frac{N}{n} \rceil$ by assigning n panels in one cube. In other words, there are 8^h cubes at the bottom level. A parallel FMM further distributes a number of cubes into different processors to evaluate $\mathcal{P}$. In the following steps, the stochastic $\mathcal{P} \times Q$ is evaluated in two passes: an upward pass for multipole-expansions (MEs) and a downward pass for local-expansions (LEs), both of which are further illustrated with details below.

5.4.1 Upward Pass

The upward-pass accumulates the multipole-expanded near-field interaction starting from the bottom level ($l = 0$). For each child cube (leaf) without variation (nominal contribution to P_0) at the bottom level, it first evaluates the stochastic geometrical moment with (5.4) for all panels in that cube. If each panel experiences a variation ξ_d or ξ_h, it calculates $P_i(\xi) \times q(i \neq 0, \xi = \xi_d, \xi_h)$ by adding perturbation $h_i\xi_h$ or $d_i\xi_d$ to consider different variation sources, and then evaluates the stochastic geometric moments with (5.9).

After building the MEs for each panel, it transverses to the upper level to consider the contribution from parents. The moment of a parent cube can be efficiently updated by summing the moments of its 8 children via a M2M operation. Based on Eq.(5.6), the M2M translates the children's $\hat{M}_p$ into their parents.

The M2M operations at different parents are performed in parallel since there is no data-dependence. Each processor builds its own panels' stochastic geometric moments while ignoring the existence of other processors.

5.4.2 Downward Pass

The potential evaluation for the observer is managed during a downward pass. At l-th level ($l > 0$), two cubes are said to be *adjacent* if they have at least one common vertex. Two cubes are said to be *well separated* if they are not adjacent at level l but their parent cubes are adjacent at level $l - 1$. Otherwise, they are said to be *far* from each other. The list of all the well-separated cubes from one cube at level l is called the *interaction list* of that cube.

From the top level $l = H-1$, interactions from the cubes on the interaction list to one cube are calculated by a M2L operation at one level (M2L operation at top level). Assuming that a source-parent center r_c is changed to an observer-parent's center r_0, this leads to a LE (5.7) using the ME (5.4) when exchanging the r and

d. As such, the M2L operation translates the source's $\hat{M}_p$ into the observer's $\hat{L}_p$ for a number of source-parents on the interaction list of one observer-parent at the same level. Due to the use of the interaction list, the M2L operations have the data-dependence that introduces overhead for a parallel evaluation.

After the M2L operation, interactions are further recursively distributed down to the children from their parents by a L2L operation (converse of the upward pass). Assume that the parent's center r_0 is changed to the child's center r_0' by a constant $\mathbf{h}$. Identical to the M2M update by (5.6), a L2L operation updates $\mathbf{r}$ by $\mathbf{r}' = \mathbf{r} + \mathbf{h}$ for all children's $\hat{L}_k$s. In this stage, all processors can perform the same M2L operation at the same time on different data. This perfectly employs the parallelism.

Finally, the FMM sums the L2L results for all leaves at the bottom level ($l = 0$) and tabulates the computed products $P_i \times q_j$ $(i, j = 0, 1, \cdots, n)$. By summing up the products in order, the FMM returns the product $\mathcal{P} \times Q^{(i)}$ in (5.16) for the next GMRES iteration.

5.4.3 Data Sharing and Communication

The total runtime complexity for the parallel FMM using stochastic geometrical moments can be estimated by $O(N/B) + O(log_8 B) + C(N, B)$, where N is the total number of panels and B is the number of used processors. The $C(N, B)$ implies communication or synchronization overhead. Therefore, it is desired to minimize the overhead of data sharing and communication.

We notice that data dependency mainly comes from the interaction list during M2L operations. In this operation, a local cube needs to know the ME moments from cubes in its interaction list. To design a task distribution with small latency between computation and communication, our implementation uses a complement interaction list and prefetch operation.

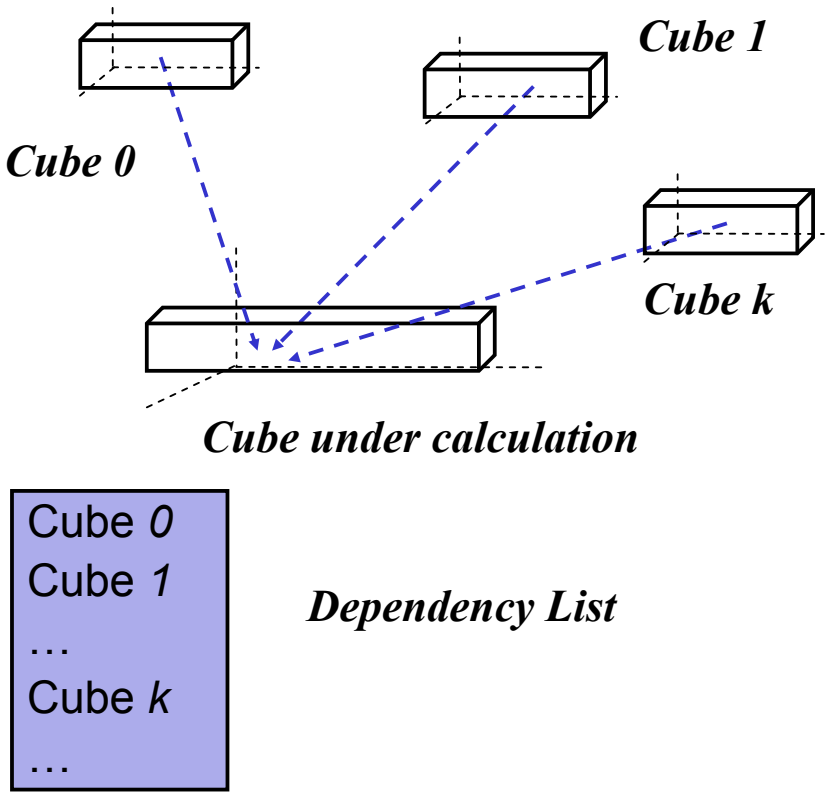


Figure 5.2: Prefetch operation in M2L.

As shown in Figure(5.2), the complement interaction list (or dependency list) for the cube under calculation records cubes that require its ME moments to be listed within the shaded area. As such, it first anticipates which ME moments will be needed by other dependent cubes (such as Cube 0, $\cdots$, Cube k shown in Figure(5.2)) and distributes the required ME moments prior to the computation. From the point of view of these dependent cubes, they can "prefetch" the required ME moments. Therefore, the communication overhead can be significantly reduced.

5.5 Incremental GMRES

The parallel FMM presented in Section 5.4 provides a fast matrix-vector-product for the fast GMRES iteration. As discussed in Section 5.2 and 5.3, another critical factor for a fast GMRES is the construction of a good preconditioner. In this section, to improve the convergence of GMRES iteration, we first present a deflated power iteration to improve convergence during the extraction. Then, we introduce an incremental precondition in the framework of the deflated power iteration.

5.5.1 Deflated Power Iteration

The convergence of GMRES can be slow in the presence of degenerated small eigen values of the potential matrix $\mathcal{P}$, such as the case for most extraction problems with fine meshes. Constructing a preconditioner W to shift the eigen value distribution (spectrum) of a preconditioned matrix $W \cdot \mathcal{P}$ can significantly improve the convergence [GGM07]. This is one of the so called *deflated GMRES* methods [SS07].

To avoid fully decomposing $\mathcal{P}$, an implicitly restarted Arnoldi method by ARPACK[1] can be applied to find its first K eigen values $[\lambda_1, ..., \lambda_K]$ and its Kth-

[1]http://www.caam.rice.edu/software/ARPACK/

order Krylov subspace composed by the first K eigen vectors $V_K = [v_1, ..., v_K]$, where

$$\mathcal{P}V_K = V_K D_K, \quad V_K^T V_K = I. \tag{5.18}$$

Note that D_K is a diagonal matrix composed of the first K eigen values

$$D_K = V_K^T A V_K = diag[\lambda_1, ..., \lambda_K]. \tag{5.19}$$

Then, an according spectrum preconditioner is formed

$$W = I + \sigma(V_K D_K^{-1} V_K^T), \tag{5.20}$$

which leads to a shifted eigen-spectrum using

$$(W \cdot \mathcal{P})v_i = (\sigma + \lambda_i)v_i \; i = 1, ..., K. \tag{5.21}$$

Note that σ is the shifting value that leads to a better convergence. This method is called *deflated power iteration*. Moreover, as discussed below, the spectral preconditioner W can be easily updated in an incremental fashion.

5.5.2 Incremental Precondition

The essence of the deflated GMRES is to form a preconditioner that shifts degenerated small eigen values. For a new $\mathcal{P}'$ with updated $\delta\mathcal{P}$, the distribution of the degenerated small eigen values change accordingly. Therefore, given a preconditioner W for the nominal system with the potential matrix $\mathcal{P}^{(0)}$, it would be expensive for another native Arnoldi iteration to form a new preconditioner W' for a new $\mathcal{P}'$ with updated $\delta\mathcal{P}$ from $\mathcal{P}^{(1)}, \cdots, \mathcal{P}^{(n)}$. Instead, we show that W can be incrementally updated as follows.

If there is a perturbation $\delta\mathcal{P}$ in $\mathcal{P}$, the perturbation δv_i of ith eigen vectors v_i

$(k = 1, ..., K)$ can be given by [Ste01]

$$\delta v_i = V_i B_i^{-1} V_i^T \delta \mathcal{P} v_i. \tag{5.22}$$

Note that V_i is the subspace composed of

$$[v_1, ..., v_j, ..., v_K]$$

and B_i is the perturbed spectrum

$$diag[\lambda_i - \lambda_1, ..., \lambda_i - \lambda_j, ..., \lambda_i - \lambda_K]$$

$(j \neq i,\ i, j = 1, ..., K)$. As a result, δV_K can be obtained similarly for K eigen vectors.

Assume that the perturbed preconditioner is W'

$$\begin{aligned} W' &= (I + \sigma V'_K (D'_K)^{-1} (V'_K)^T) \\ &= W + \delta W \end{aligned} \tag{5.23}$$

where

$$V'_K = V_K + \delta V_K, \quad D'_K = (V'_K)^T P V'_K. \tag{5.24}$$

After expanding V'_K by V_K and δV_K, the incremental change in the preconditioner W can be obtained by

$$\delta W = \sigma(E_K - V_K D_K^{-1} F_K D_K^{-1} V_K). \tag{5.25}$$

where

$$E_K = \delta V_K D_K^{-1} V_K^T + (\delta V_K D_K^{-1} V_K^T)^T. \tag{5.26}$$

and

$$F_K = \delta V_K^T V_K D_K + (\delta V_K^T V_K D_K)^T. \tag{5.27}$$

Note that all the above inverse operations only deal with the diagonal matrix D_K and hence the computational cost is low.

Since there is only one Arnoldi iteration to construct a nominal spectral preconditioner W, it can only be efficiently updated when $\delta\mathcal{P}$ changes. For example, $\delta\mathcal{P}$ is different when one alters the perturbation range h_1 of panel-width or changes the variation type from panel-width h to panel-distance d. We call this deflated GMRES method with the incremental precondition an *iGMRES method*.

For our problem in Eq.(5.16), we first analyze an augmented nominal system with

$$\begin{aligned}
\mathbf{W} &= diag[W,\ W, \cdots,\ W] \\
\mathbf{P} &= diag[\mathcal{P}^{(0)},\ \mathcal{P}^{(0)}, \cdots,\ \mathcal{P}^{(0)}] \\
\mathbf{D}_K &= diag[D_K,\ D_K, \cdots,\ D_K] \\
\mathbf{V}_K &= diag[V_K,\ V_K, \cdots,\ V_K],
\end{aligned}$$

which are all block diagonal with n blocks. Hence there is only one preconditioning cost from the nominal block $\mathcal{P}^{(0)}$. In addition, the variation contributes to the perturbation matrix by

$$\delta\mathcal{P} = \begin{pmatrix} 0 & \mathcal{P}_{0,1} & \cdots & \mathcal{P}_{0,n} \\ \mathcal{P}_{1,0} & 0 & \cdots & \mathcal{P}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{P}_{n,0} & \mathcal{P}_{n,1} & \cdots & 0 \end{pmatrix}. \tag{5.28}$$

5.6 Experimental Results

Based on the proposed algorithm, we have developed a program *piCap* using C++ on Linux network servers with Xeon processors (2.4GHz CPU and 2GB memory). In this section, we first validate the accuracy of stochastic geometrical moments by comparing them with the Monte-Carlo integral. Then, we study the parallel runtime scalability when evaluating the potential interaction using MVP with charge. In addition, the incremental GMRES preconditioner is verified when compared to its non-incremental counterpart with total runtime.

5.6.1 Accuracy Validation

To validate the accuracy of *SGM* by first-order and second-order expansions, we use two distant square panels. The nominal center-to-center distance d is d_0, and nominal panel width h is h_0.

5.6.1.1 Incremental Analysis

One possible concern is about the accuracy of incremental analysis, which considers independent variation sources separately and combines their contributions to get the total variable capacitance. In order to validate this, we first introduce panel width variation (Gaussian distribution with perturbation range h_1), and calculate the variable capacitance distribution. Then, panel distance variation d_1 is added and the same procedure is conducted. As such, according to incremental analysis, we can obtain the total capacitance as a superposition of nominal capacitance and both variation contributions. Moreover, we introduce the Monte Carlo simulations (10000 times) as the baseline, where both variations are introduced simultaneously. The comparison is shown in Table 5.1, and we can observe that the results from incremental analysis can achieve high accuracy.

Actually, it is ideal to consider all variations simultaneously, but the dimension

Table 5.1: Incremental Analysis vs. Monte Carlo Method

2 panels, $d_0 = 10\mu m, h_0 = 2\mu m$, $d_1 = 30\%d_0$, $h_1 = 30\%h_0$			
	Incremental Analysis (fF)	Monte Carlo (fF)	Error (%)
$\mu_{C_{ij}}$	-1.1115	-1.1137	0.19
$\sigma_{C_{ij}}$	0.11187	0.11211	0.21
2 panels, $d_0 = 25\mu m, h_0 = 5\mu m$, $d_1 = 20\%d_0$, $h_1 = 20\%h_0$			
	Incremental Analysis (fF)	Monte Carlo (fF)	Error (%)
$\mu_{C_{ij}}$	-2.7763	-2.7758	0.018
$\sigma_{C_{ij}}$	0.19477	0.194	0.39

of system can increase exponentially with the number of variations and thus the complexity is prohibited. As a result, when the variation sources are independent, it is possible and necessary to separate them by solving the problem with each variation individually.

5.6.1.2 Stochastic Geometrical Moments

Next, the accuracy of proposed method based on stochastic geometrical moments (SGM) is verified with the same two panel examples. To do so, we introduce a set of different random variation ranges with Gaussian distribution for their distance d and width h. For this example, Monte-Carlo method is used to validate the accuracy of stochastic geometrical moments.

First, Monte-Carlo method calculates their C_{ij} 3000 times and each time the variation with a normal distribution is introduced to distance d randomly.

Then, we introduce the same random variation to geometric moments in (5.9) with stochastic polynomial expansion. Because of an explicit dependence on geometrical parameters according to (5.4), we can efficiently calculate $\hat{C}_{ij}$. Table 5.2 shows the C_{ij} value and runtime using the aforementioned two approaches. The comparison in Table 5.2 shows that stochastic geometric moments can not only keep high accuracy, which yields an average error of 1.8%, but also are up to ~ 347 faster than the Monte-Carlo method.

Table 5.2: Accuracy and Runtime(s) Comparison between MC(3000), *piCap*.

2 panels, $d_0 = 7.07\mu m, h_0 = 1\mu m$, $d_1 = 20\%d_0$		
	MC	piCAP
$C_{ij}(fF)$	-0.3113	-0.3056
Runtime (s)	2.6965	0.008486
2 panels, $d_0 = 11.31\mu m, h_0 = 1\mu m$, $d_1 = 10\%d_0$		
	MC	piCAP
$C_{ij}(fF)$	-0.3861	-0.3824
Runtime (s)	2.694	0.007764
2 panels, $d = 4.24\mu m, h_0 = 1\mu m$, $d_1 = 20\%d_0, h_1 = 20\%$		
	MC	piCAP
$C_{ij}(fF)$	-0.2498	-0.2514
Runtime (s)	2.7929	0.008684

5.6.2 Speed Validation

In this part, we study the runtime scalability using a few large examples to show both the advantage of the parallel FMM for MVP and the advantage of the deflated GMRES with incremental preconditions.

5.6.2.1 Parallel Fast Multipole Method

The four large examples are comprised of $20, 40, 80$ and 160 conductors, respectively. For the two-layer example with 20 conductors, each conductor is of size $1\mu m \times 1\mu m \times 25\mu m$ (width×thick×length), and *piCap* employs a uniform $3\times3\times50$ discretization. Fig. 5.3 shows its structure and surface discretization.

For each example, we use a different number of processors to calculate the MVP of $P \times q$ by the parallel FMM. Here we assume that only d has a 10% perturbation range with Gaussian distribution. As shown in Table 5.3, the runtime of the parallel MVP decreases evidently when more processors are involved. Due to the use of the complement interaction list, the latency of communication is largely reduced and the runtime shows a good scalability versus the number of

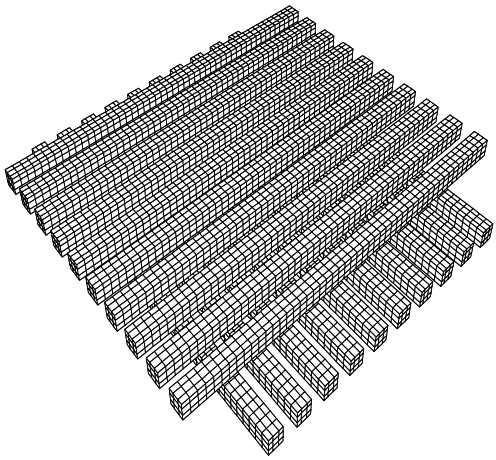


Figure 5.3: The structure and discretization of two-layer example with 20 conductors.

Table 5.3: MVP Runtime (seconds)/Speedup Comparison for four different examples

#wire	20	40	80	160
#panels	12360	10320	11040	12480
1 proc	0.737515/1.0	0.541515/1.0	0.605635/1.0	0.96831/1.0
2 procs	0.440821/1.7X	0.426389/1.4X	0.352113/1.7X	0.572964/1.7X
3 procs	0.36704/2.0X	0.274881/2.0X	0.301311/2.0X	0.489045/2.0X
4 procs	0.273408/2.7X	0.19012/2.9X	0.204606/3.0X	0.340954/2.8X

processors. Moreover, the total MVP runtime with four processors is about $3X$ faster on average than runtime with a single processor.

It is worth mentioning that MVP needs to be performed many times in the iterative solver such as GMRES. Hence, even a small reduction of MVP runtime can lead to an essential impact on the total runtime of the solution, especially when the problem size increases rapidly.

5.6.2.2 Deflated GMRES

piCap has been used to perform analysis for three different structures as shown in Fig. 5.4. The first is a plate with size $32\mu m \times 32\mu m$ and discretized as 16×16 panels. The other two examples are Cubic capacitor and Bus2x2 cross-over

structures. For each example, we can obtain two stochastic equation systems in (5.17) by considering variations separately from width h of each panel and from the centric distance d between two panels, both with 20% perturbation ranges from their nominal values which should obey the Gaussian distribution.

To demonstrate the effectiveness of the deflated GMRES with a spectral preconditioner, two different algorithms are compared in Table 5.4. In the baseline algorithm (column "diagonal prec."), it constructs a simple preconditioner using diagonal entries. As the fine mesh structure in the extraction usually introduces degenerated or small eigen values, such a preconditioning strategy within the traditional GMRES usually needs much more iterations to converge. In contrast, since the deflated GMRES employs the spectral preconditioner to shift the distribution of non-dominant eigen values, it accelerates the convergence of GMRES leads to a reduced number of iterations. As shown by Table 5.4, the deflated GMRES consistently reduces the number of iterations by 3X on average.

Table 5.4: Runtime and Iteration Comparison for different Examples.

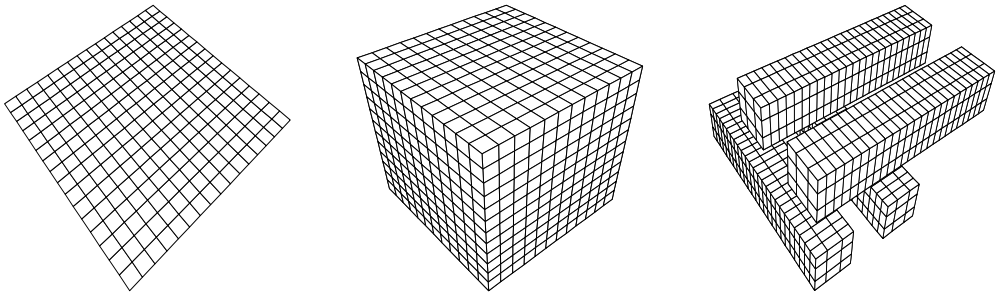
	#panel	#variable	diagonal prec.		spectral prec.	
			# iter	time	# iter	time
single plate	256	768	29	24.594	11	8.625
cubic	864	2592	32	49.59	11	19.394
cross-over	1272	3816	41	72.58	15	29.21

5.6.2.3 Incremental Preconditioner

With the spectral preconditioner, an incremental GMRES can be designed easily to update the preconditioner when considering different stochastic variations. It quite often happens that a change occurs in the perturbation range of one geometry parameter or in the variation type from one geometry parameter to the other. As the system equation in (5.17) is augmented to 3X larger than the nominal system, it becomes computationally expensive to apply any non-incremental

GMRES methods whenever there is a change from the variation. As shown by the experiments, the incremental preconditioning in the deflated GMRES can reduce the computation cost dramatically.

As described in Section 5.5, iGMRES needs to perform the precondition only one time for the nominal system and to update the preconditioner with perturbations from matrix block $P^{(1)}$. In order to verify the efficiency of such an incremental preconditioner strategy, we apply two different perturbation ranges for h_1 for panels of the two-layer 20 conductors shown in Fig. 5.3. Then, we compare the total runtime of the iGMRES and GMRES, both with the deflation. The results are shown in Table 5.5.



(a) plate (b) cubic (c) bus2x2

Figure 5.4: Test structures:(a)plate;(b)cubic;(c)cross-over2x2

Table 5.5: Total Runtime (seconds) Comparison for 2-layer 20-conductor by different methods

discretization w×t×l	#panel	#variable	Total Runtime(s)	
			non-incremental	incremental
$3 \times 3 \times 7$	2040	6120	419.438	81.375
$3 \times 3 \times 15$	3960	11880	3375.205	208.266
$3 \times 3 \times 24$	6120	18360	-	504.202
$3 \times 3 \times 60$	14760	44280	-	7584.674

From Table 5.5, we can see that a non-incremental approach needs to construct its preconditioner whenever there is an update of variations, which is very

time consuming. Our proposed *iGMRES* can reduce CPU time greatly during the construction of the preconditioner by only updating the nominal spectral preconditioner incrementally with (5.25). The result of iGMRES shows a speedup up to $15X$ over non-incremental algorithms and only iGMRES can finish all large-scale examples up to 14760 panels.

5.7 Conclusion

In this chapter, we have proposed the use of geometrical moments to capture local random variations for full-chip capacitance extraction. Based on geometrical moments, the stochastic capacitance can be calculated via stochastic orthogonal polynomials (SoPs) by fast multi-pole method (FMM) in a parallel fashion. As such, the complexity of the matrix-vector product (MVP) can be largely reduced to evaluate both nominal and stochastic values. Moreover, one incrementally preconditioned GMRES is developed to consider different types of updates of variations with an improved convergence by spectrum deflation.

A number of experiments show that our approach is $\sim 347X$ faster than the Monte-Carlo based evaluation of variation with a similar accuracy, up to $3X$ faster than the serial method in MVP, and up to $15X$ faster than non-incremental GMRES methods. The observed speedup of our approach is analyzed in two manners: the first is from the efficient parallel FMM, and the other is from the non-Monte-Carlo evaluation by SoPs. Future work is planned to extend our approach to deal with the general capacitance extraction with a non-square-panel geometry.

CHAPTER 6

Conclusion

As integrated circuits enter into the nanometer era, process variation has become the dominant challenge for nanoscale circuit design and fabrication. Many uncertainties can be introduced during the manufacturing process such as lithography, chemical mechanical polishing (CMP), etching, etc. Consequently, circuit parameters can deviate significantly from their nominal values specified by designers. This in turn will cause circuit behavior or performance merits to differ from design specifications under the nominal condition. To address this issue, efficient and accurate statistical analysis methodologies are needed to model the effects of process variations and further predict the stochastic behavior of custom circuit designs.

In Chapter 2, an efficient algorithm is presented to accurately predict the failure rate of SRAM cells based on importance sampling scheme. Specifically, the "Kullback-Leibler (KL) distance" is adopted from information theory [CT91] to measure the distance between the given distribution and the optimal proposed distribution. Then, the KL distance is minimized by parameterizing Gaussian distributions to approximate the optimal proposed distribution as closely as possible. As such, the convergence of failure rate estimation is significantly expedited. The extensive experiments show that the proposed algorithm can accurately estimate the failure probability of SRAM cells with $5200X$ speed-up over Monte Carlo and can be more than $40X$ faster than other existing methods [QTD10, KJN06].

In Chapter 3, a fast statistical algorithm is proposed to predict the failure prob-

ability of memory circuits in high dimensions (e.g., SRAM bit-cell, delay chain). This is the first work that successfully applies the importance sampling paradigm to high dimensional problems. Experiments show that the proposed method can provide 708X speedup over MC with the same accuracy for a 108-dimensional problem. Also, the proposed approach is 17X faster than the Statistical Blockade method [SR09] and trumps existing importance sampling methods that completely fail to provide any reasonable accuracy.

In Chapter 4, a new mapping algorithm is developed to obtain the "arbitrary" circuit behavioral distributions with great computational complexity reduction. This method utilizes Latin Hypercube Sampling (LHS) coupled with a correlation control technique to generate a few samples and further analytically evaluate the high-order moments of the circuit behavior. Afterwards, the "arbitrary" probabilistic distributions of circuit behavior can be extracted using moment-matching method. The proposed method has been successfully applied to high-dimensional and strongly nonlinear problems with linear complexity. The experiments demonstrate that the proposed method can provide several orders of magnitude speedup over crude Monte Carlo method while retaining the accuracy.

In Chapter 5, a parallel full chip capacitance extraction algorithm named *p-iCAP* is proposed. With the use of stochastic-polynomial expanded geometrical moments, the parallel fast multipole method (FMM) can efficiently solve the large-scale dense linear system in parallel and further evaluate the parasitic capacitance and its variation. Experiments on a few different large examples show that *piCAP* is hundreds of times faster than the Monte-Carlo method without compromising the accuracy.

References

[AB03] S.K. Au and J.L. Beck. "Important sampling in high dimensions." *Structural Safety*, **25**(2):139 – 163, 2003.

[AN06] Kanak Agarwal and Sani Nassif. "Statistical analysis of SRAM cell stability." In *Proceedings of the 43rd annual Design Automation Conference*, DAC '06, pp. 57–62, 2006.

[Ass05] Semiconductor Industry Associate. "International Technology Roadmap for Semiconductors." 2005.

[BA97] A. W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. New York:Oxford University Press, 1997.

[BB05] T. Bengtsson B. Li and P. Bickel. "Curse-of-dimensionality revisited: Collapse of importance sampling in very high-dimensional systems." *Technical Report No.696,Department of Statistics, UC-Berkeley*, 2005.

[BDG09] Alberto Bosio, Luigi Dilillo, Patrick Girard, Serge Pravossoudovitch, and Arnaud Virazel. *Advanced Test Methods for SRAMs: Effective Solutions for Dynamic Fault Detection in Nanoscaled Technologies*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[BDM02] K.A. Bowman, S.G. Duvall, and J.D. Meindl. "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration." *Solid-State Circuits, IEEE Journal of*, **37**(2):183 –190, feb 2002.

[BKM05] P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. "A tutorial on the cross entropy method." *Annals of Operations Research*, **134**:19–67, 2005.

[Bra04] C.A. Brau. *Modern Problems In Classical Electrodynamics*. Oxford Univ. Press, 2004.

[CCS04] Runzi Chang, Yu Cao, and C.J. Spanos. "Modeling the electrical effects of metal dishing due to CMP for on-chip interconnect optimization." *Electron Devices, IEEE Transactions on*, **51**(10):1577–1583, 2004.

[CCS08] Jian Cui, Gengsheng Chen, Ruijing Shen, Sheldon Tan, Wenjian Yu, and Jiarong Tong. "Variational capacitance modeling using orthogonal polynomial method." In *Proceedings of the 18th ACM Great Lakes symposium on VLSI*, pp. 23–28, Orlando, Florida, USA, 2008.

[Con80] W. J. Conover. *Practical Nonparametric Statistics*. Wiley, 1980.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[Den01] Mark Denny. "Introduction to importance sampling in rare-event simulations." *European Journal of Physics*, **22**(4):403–411, 2001.

[DL11] Changdao Dong and Xin Li. "Efficient SRAM Failure RatePrediction via Gibbs Sampling." In *Proceedings of the 43rd annual Design Automation Conference*, DAC'11, 2011.

[DM03] P.G. Drennan and C.C. McAndrew. "Understanding MOSFET mismatch for analog design." *IEEE J. of Solid State Circuits*, **38**(3):450 – 456, 2003.

[DQS08] Lara Dolecek, Masood Qazi, Devavrat Shah, and Anantha 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.

[DS11] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. Addison Wesley, 2011.

[EBS97] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf. "The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits." *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, **5**(4):360 –368, dec. 1997.

[GBD12] Fang Gong, Sina Basir-Kazeruni, Lara Dolecek, and Lei He. "A Fast Estimation of SRAM Failure Rate Using Probability Collectives." In *Proc. ACM ISPD*, pp. 41–47, 2012.

[GGM07] L. Giraud, S. Gratton, and E. Martin. "Incremental spectral preconditioners for sequences of linear systems." *Appl. Num. Math.*, pp. 1164–1180, 2007.

[GS96] Fishman G.S. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag New York, Inc., 1996.

[GYH09] F. Gong, H. Yu, and L. He. "PiCAP: a parallel and incremental capacitance extraction considering stochastic process variation." In *Proc. ACM/IEEE Design Automation Conf. (DAC)*, pp. 764–769, 2009.

[GYH11] F. Gong, H. Yu, and L. He. "Stochastic analog circuit behavior modeling by point estimation method." In *2011 International Symposium on Physical Design 2011.*, pp. 175–182, March 2011.

[GYS10] Fang Gong, Hao Yu, Yiyu Shi, Daesoo Kim, Junyan Ren, and Lei He. "QuickYield: an efficient global-search based parametric yield estimation with performance constraints." In *Proc. ACM/IEEE Design Automation Conf. (DAC)*, pp. 392–397, 2010.

[HF08] Tibshirani R. Hastie, T. and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, NY, USA, 2008.

[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.

[IC82] R. Iman and W.J. Conover. "A distribution-free approach to inducing rank correlation among input variables." *Commun Stat: Simul Comput*, **B11**(3):311–334, 1982.

[Jac75] J. D. Jackson. *Classical Electrodynamics*. John Wiley and Sons, 1975.

[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 *Computer-Aided Design (ICCAD), 2010 IEEE/ACM International Conference on*, pp. 703 –708, 2010.

[KJN06] Rouwaida Kanj, Rajiv Joshi, and Sani 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.

[Li10] X. Li. "Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/RF circuits." *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, **29**(11):1661 –1668, nov. 2010.

[LL08] X. Li and H. Liu. "Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations." In *Design Automation Conference, 2008. Proceedings*, pp. 38–43, june 2008.

[LLG04] Xin Li, Jiayong Le, P. Gopalakrishnan, and L. T. Pileggi. "Asymptotic probability extraction for non-normal distributions of circuit performance." In *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, pp. 2–9, 2004.

[LNP00] Ying Liu, Sani R. Nassif, Lawrence T. Pileggi, and Andrzej J. Strojwas. "Impact of interconnect variations on the clock skew of a gigahertz microprocessor." In *Proceedings of the 37th Annual Design Automation Conference*, pp. 168–171, Los Angeles, California, United States, 2000.

[LV06] F. Liese and I. Vajda. "On divergences and informations in statistics and information theory." *IEEE Transactions on Information Theory*, **52**(10):4394C4412, 2006.

[LZP08] Xin Li, Yaping Zhan, and Lawrence Pileggi. "Quadratic statistical MAX approximation for parametric yield estimation of analog/RF integrated circuits." *IEEE Tran. on Computer-aided-design (TCAD)*, **27**:831–843, 2008.

[Mel07] T. Homem de Mello. "A study on the cross-entropy method for rare event probability estimation." *INFORMS Journal on Computing*, **19**(3):381–394, 2007.

[MMR10] Debasis Mukherjee, Hemanta Kr. Mondal, and B.V.R. Reddy. "Static noise margin analysis of SRAM cell for high speed application." *International Journal of Computer Science Issues*, **7**(5):175–180, 2010.

[Moo75] G.E. Moore. "Progress in digital integrated electronics." *Electron Devices Meeting, 1975 International*, **21**:11 – 13, 1975.

[MR02] T. Homem-de Mello and R. Y. Rubinstein. "Estimation of rare event probabilities using cross-entropy." In *Proc. of the Winter Simulation Conference*, pp. 310–319, 2002.

[Nas01] S. Nassif. "Modeling and analysis of manufacturing variations." *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 223–228, 2001.

[NW91] K. Nabors and J. White. "FastCap: A multipole accelerated 3D capacitance extraction program." *IEEE Tran. on Computer-aided-design (TCAD)*, **10**(11):1447–1459, 1991.

[PDW89] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers. "Matching properties of MOS transistors." *IEEE J. of Solid State Circuits*, **305**(3):1433–1439, 1989.

[PP01] A. Papoulis and S. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2001.

[PR90] L. T. Pillage and R. A. Rohrer. "Asymptotic waveform evaluation for timing analysis." *IEEE Tran. on Computer-aided-design (TCAD)*, **9**(4):352–366, 1990.

[PS08] Andrei Pavlov and Manoj Sachdev. "CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies: Process-Aware SRAM Design and Test." *Springer Publisher*, 2008.

[QTD10] 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.

[Ran99] William Theodore Rankin, III. *Efficient parallel implementations of multipole based n-body algorithms*. PhD thesis, Durham, NC, USA, 1999.

[RE00] Vijay K. Rohatgi and A. K. Md. Ehsanes Saleh. *An Introduction to Probability and Statistics*. Wiley-Interscience, 2000.

[RG09] R. Y. Rubinstein and P. W. Glynn. "How to deal with the curse of dimensionality of likelihood ratios in monte carlo simulation." *Stochastic Models*, **25**:547 – 568, 2009.

[RKW07] 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.

[Ros75] E. Rosenblueth. "Point estimation for probability moments." *Proc. Nat. Acad. Sci. U.S.A.*, **72**(10):3812–3814, 1975.

[RR07] A. Ridder and R. Y. Rubinstein. "Minimum cross-entropy methods for rare-event simulation." *Simulation: Transactions of the Society for Modeling and Simulation International*, **83**:769–784, 2007.

[RWK06] D. Rajnarayan, D. H. Wolpert, and I. Kroo. "Optimization under uncertainty using probability collectives." *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006.

[Saa03] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[SD97] S.Kapur and D.Long. "IES3: a fast integral equation solver for efficient 3-dimensional extraction." In *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, pp. 448 –455, San Jose, CA , USA, 1997.

[SLK98] Weiping Shi, Jianguo Liu, Naveen Kakani, and Tiejun Yu. "A fast hierarchical algorithm for 3-D capacitance extraction." In *Proc. ACM/IEEE Design Automation Conf. (DAC)*, pp. 212–217, San Francisco, California, United States, 1998.

[SR07] A. Singhee and R.A. 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.

[SR09] Amith Singhee and Rob A. Rutenbar. "Statistical blockade: very fast statistical simulation and modeling of rare circuit events and its application to memory design." *IEEE Tran. on CAD*, **28**:1176–1189, 2009.

[SS86] Youcef Saad and Martin H. Schultz. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems." *SIAM Journal on Scientific and Statistical Computing*, **7**(3):856–869, 1986.

[SS07] Valeria Simoncini and Daniel B. Szyld. "Recent computational developments in Krylov subspace methods for linear systems." *Numerical Linear Algebra with Application*, **14**:1–59, 2007.

[Ste87] M. Stein. "Large sample properties of simulations using Latin Hypercube Sampling." *Technometrics*, **29**(2):143–151, May 1987.

[Ste01] G. W. Stewart. *Matrix algorithms (Volume II): Eigensystems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[VWG06] S. Vrudhula, J. M. Wang, and P. Ghanta. "Hermite polynomial based interconnect analysis in the presence of process variations." *IEEE Tran. on Computer-aided-design (TCAD)*, **25**(10):2001–2011, 2006.

[WS93] M. S. Warren and J. K. Salmon. "A parallel hashed Oct-Tree N-body algorithm." In *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, Supercomputing '93, pp. 12–21, Portland, Oregon, United States, 1993.

[WYL09] Jian Wang, Soner Yaldiz, Xin Li, and Lawrence T. Pileggi. "SRAM parametric failure analysis." In *Proc. ACM/IEEE Design Automation Conf. (DAC)*, pp. 496–501, 2009.

[XK02] Dongbin Xiu and George Em Karniadakis. "The Wiener-Askey polynomial chaos expansion for stochastic differential equations." *SIAM J. Scientific Computing*, **24**:619–644, 2002.

[YBZ03] L. Ying, G. Biros, D. Zorin, , and H. Langston. "A new parallel kernel-independent fast multi-pole method." In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, pp. 14–23, Phoenix, AZ, USA, 2003.

[YLS07] Y. Yang, P. Li, V. Sarin, and W. P. Shi. "Impedance extraction for 3-D structures with multiple dielectrics using preconditioned boundary

element method." In *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, pp. 7–10, San Jose, CA, 2007.

[YLW10] H. Yu, X. Liu, H. Wang, and S. Tan. "A fast analog mismatch analysis by an incremental and stochastic trajectory piecewise linear macromodel." In *Proc. IEEE/ACM Asia South Pacific Design Automation Conf. (ASPDAC)*, pp. 211–216, 2010.

[ZO00] Yan-Gang Zhao and Tetsuro Ono. "New point estimation for probability moments." *Journal of Engineering Mechanics*, **126**(4):433–436, 2000.

[ZW05] Zhenhai Zhu and J. White. "FastSies: A fast stochastic integral equation solver for modeling the rough surface effect." In *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, pp. 675–682, Los Alamitos, CA, USA, 2005.

[ZZC07] Hengliang Zhu, Xuan Zeng, Wei Cai, Jintao Xue, and Dian Zhou. "A sparse grid based spectral stochastic collocation method for variations-aware capacitance extraction of interconnects under nanometer process technology." In *Proc. IEEE/ACM Design, Automation, and Test in Europe (DATE)*, pp. 1514–1519, Nice, France, 2007.