國立交通大學

電信工程學系

碩士論文

應用進階變異數縮減技術的蒙地卡羅
軟性電子錯誤率分析

Applying Advanced Variance-Reduction Techniques to Monte-Carlo Based Soft Error Rate Analysis

研究生：吳欣恬

指導教授：溫宏斌 教授

中華民國一百年六月

應用進階變異數縮減技術的蒙地卡羅
軟性電子錯誤率分析
Applying Advanced Variance-Reduction Techniques
to Monte-Carlo Based Soft Error Rate Analysis

研究生：吳欣恬　　Student：Xin-Tian Wu

指導教授：溫宏斌　　Advisor：Hung-Pin Wen

國立交通大學
電信工程系
碩士論文

A Thesis
Submitted to Department of Communication Enginerring
College of Electrical and Computer Engineering National
Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Communication Engineering

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年六月

應用進階變異數縮減技術的蒙地卡羅
軟性電子錯誤率分析

研究生：吳欣恬　　　　　　　　　　　　　指導教授：溫宏斌

國立交通大學　　　電信工程研究所碩士班

摘要

使用統計性的方法在製程變異下準確估計電路的軟性電子錯誤率分析是很重要的。製程變異參數可以分成晶圓間和晶圓內的變異兩個部分，晶圓內的變異存在空間相關性使得越接近彼此的製程變異參數會越相似，此外我們考慮了空間相關性的因素。然而，因為沒有考慮降低變異數，使得現今的軟性錯誤率統計分析研究無法達到良好的準確性。在這篇論文裡，我們提出了一個高準確性的統計模型，利用蒙地卡羅去分析這些統計模型，並且達到了比較好的收斂與增加速度。此外，我們利用降低變異數的方法來分析這些統計模型。實驗結果顯示，我們可以在更短的時間內更準確的估計出軟性錯誤率。

Applying Advanced Variance-Reduction Techniques to Monte-Carlo Based Soft Error Rate Analysis

Student: Xin-Tian Wu Advisor: Hung-Pin Wen

Department of Communication
Engineering
National Chiao Tung
University

ABSTRACT

Statistical methods are important to accurately estimate soft error rates (SERs) of circuits with process variations. Process variations can be classified into the inter-die variations and the intra-die variations. The intra-die variations exist spatial correlations where the devices that are close to each other are more alike. Therefore, a SER analysis frameworks should include spatial correlations. However, without variance reduction, current Monte-Carlo-based SER analysis can not achieve a satisfactory accuracy with reasonable speed. In this work, we first review statistical soft error rate analysis based on which a Monte-Carlo framework is built. We further employ the quasi-random sequences, which successfully speeds up the convergence of simulation error and shortens the runtime. Moreover, advanced sampling techniques are incorporated for variance reduction of SSERs. Experimental results show that this framework is capable of more precisely estimating circuit SSERs and reaches better speedups.

誌謝

這篇論文的順利完成，首先要感謝我的指導老師溫宏斌教授。感謝老師在研究知識方面給我很多的指引，更感謝溫老師在寫論文困惑時期不辭辛勞的共同討論與溝通，真的很感謝老師。在研究所的這兩年中，感謝老師在研究方面與待人處世方面，給予寶貴的意見，使我獲益良多。

接著要感謝CIA實驗室的成員佳伶、千惠、宣銘、家慶、玗璇、釗炯、竣惟、凱華、鈞堯、昱澤、鉉崴，謝謝你們在研究所時期提供我許多寶貴的意見，以及在研究道路上的陪伴，使我在研究的路途上學到了很多，也豐富了我的研究生活。最後要感謝研究所的同學，謝謝他們一起打球與陪伴的日子，給予我最美好的回憶。

最後僅以此文獻給我摯愛的父母及哥哥。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Soft error rate analysis is crucial for both logic and memory circuits in sub-90nm technologies. A soft error results from radiation-induced transient faults latched by state-holding elements and depends on three masking effects [1]: *logical*, *electrical* and *timing* maskings. Logical masking occurs when the input value of one cell blocks the propagation of the transient fault under one input pattern. Due to electrical properties of cells, one transient fault attenuated by electrical masking may further disappear. The survival transient faults arrives one state-holding element outside its window of clock transition is called timing masking.

Numerous researches have been proposed to evaluate soft error for logic circuits subject to three mechanisms. Many previous works use analytical models to electrically evaluate the change of transient faults and propagates transient faults through one gate based on the logic functions. A refined model [3] is further applied to all gates with different charges deposited and to incorporate non-linear transistor current. By computing backwards the propagation of the error-latching windows efficiently, a static analysis is also proposed in [9] for timing masking.

As a result, circuit reliability has been extensively investigated where soft error rate (SER) is a key metric. SER computed by SERA [4] considers the electrical attenuation effect and error-latching probability by means of a waveform model while ignoring logical masking. AnSER [9] estimates SER for circuit hardening by applying signature observability and latching-window computation for logical and timing maskings. MARS-C [8] scales the error probability according to the specified clock period and applies the symbolic technique to both logical and electrical maskings. By waveform models, SEAT-LA [5] and the algorithm in [6] simultaneously characterize cells, flip-flops and propagation of transient faults, and compute good SER estimates when comparing to SPICE simulation.

Recently, process variation has revived as an important issue and also needed to be concerned in soft-error reaserch. The authors in [10] first investigate the different sources of process variations. The paper [11] concludes that the traditional static approaches underestimate circuit is SER. From [2], static approaches underestimate circuit is SER by up to 50% under the process variation $\sigma_{proc} = 5\%$ (± 3 σ_{proc} covers 99.73% of the distribution),

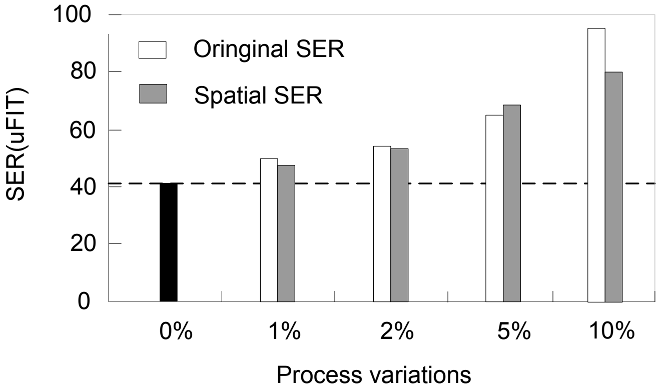


Figure 1.1: In different process variations, the SSER comparation between the circuit with spatial correlations and without spatial correaltions

or over 100% under $\sigma_{proc} = 10\%$. Moreover, process variations are classified into inter-die variations and intra-die variations. Intra-die variations exist spatial correlations, where devices that are more closer to each other will have a higher probability of being alike [15]. According to Figure 1.1, as the process variations increase, the SER with spatial correlations will increase in one circuit. Therefore, we need to consider the impact of spatial correlations. Moreover, the authors in [11] propose a symbolic frameworks for statistical SER(SSER) analysis and authors in [2] propose a learning-based framework for statistical SER analysis. Their SSER results are not accurate enough and cannot be computed efficiently where the main challenge comes from the difficulty of constructing quality cell models for transient-fault distributions and the lack of variance reduction.

Therefore, an accurate and effective QMC-based method for the SSER problem [16] can be built. However, there is still a problem in the QMC samples which is the uniformity of the quasi-sequence samples in multivariate distribution. To solve this problem, we use importance sampling to reduce variance in quasi-sequence samples.

In this work, we adopt accurate table-based models for transient-fault distributions from [21], according to which a Monte Carlo SSER framework is built. Furthermore, we customize the use of quasirandom sequences, which successfully speed up the convergence of simulation error and hence shorten runtime. However, there is still a problem on the uni-

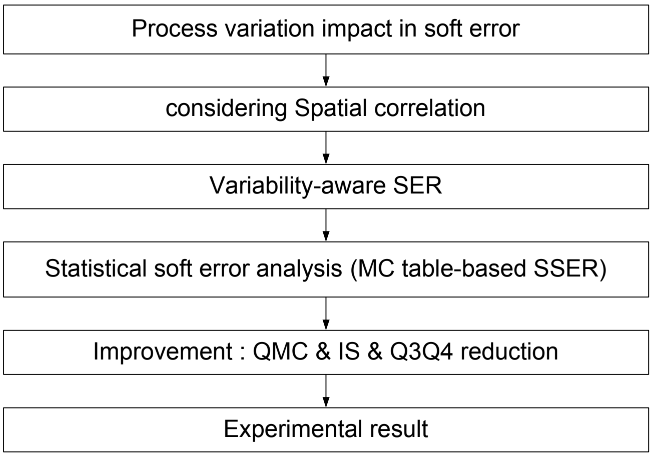

Process variation impact in soft error
considering Spatial correlation
Variability-aware SER
Statistical soft error analysis (MC table-based SSER)
Improvement : QMC & IS & Q3Q4 reduction
Experimental result

Figure 1.2: The proposed statistical SSER methodology

formity of the quasi-sequence samples in multivariate distribution. To solve this problem, we use importance sampling to reduce variance in quasi-sequence samples. From experimental results, the framework is capable of yielding more accurate SSER results compared to previous works and runs much faster.

The overview of the proposed methodology is shown as Figure 1.2. The first step considers the process variation impact on soft error. The second step considers the process variation impact and reflect it during transient fault generation and propagation. The third step builds the variability-aware cell models, which include intrinsic, systematic and random variation sources. Once the models are built, the next step is to analyze statistical soft error rates by a MC table-based framework with several key improvments, including quasi-random sequence and importance sampling. Finally, we report the full-chip reliability in terms of SSERs.

The rest of the paper is organized as follows. The fundamentals of SSER is provided in Section 2.1. In Section 3.1, the generation of our table-based cell models is detailed. Then, we propose a heuristic of using quasirandom sequences to speed up the framework and importance sampling to reduce variance in Section 4.1. Section 5.1 describes the experimental results, including the accuracy of our models, the Monte Carlo convergence with and without quasirandom sequences and importance sampling. In Section 6.1, we draw our conclusion.

Chapter 2

Fundamentals of SSER

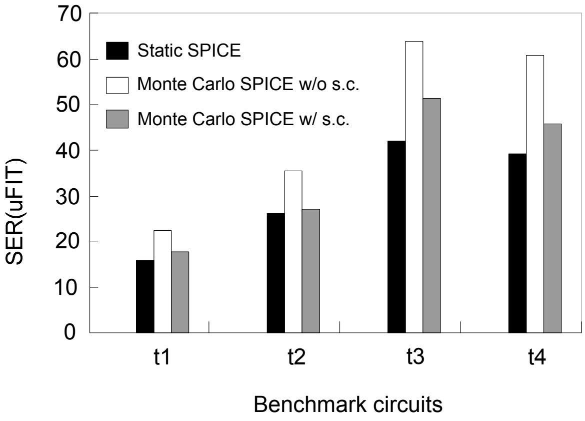


Figure 2.1: SSER comparison from static and Monte Carlo SPICE simulations, the proposed MC with spatial correlations and without spatial correlations frameworks.

2.1 Fundamentals of SSER

In this paper, the proposed SSER analysis framework needs to consider process variation impact for cell-based designs, and mainly consists of five stages: (1) correlation impact, (2) cell modeling, (3) electrical probability computation, (4) signal probability computation and (5) SER estimation. We will explain each component in detail in the following sections.

2.1.1 Correlation impact

Variations have emerged as technology scales further. High levels of device parameter variations are changing the design flows from deterministic to probabilistic as technology nodes beyond 90nm experience increasingly.

Process variations can be classified into the two categories [15]. One is the inter-die variations and the other is intra-die variations. Inter-die variation are variations that occur from one die to the another die. Intra-die variations can significantly affect the variability of performance parameters on a chip. Intra-die variations are locally layout-dependent; therefore, it is spatially correlated.

Devices tend to have similar characteristics as with similar layout patterns and proxim-

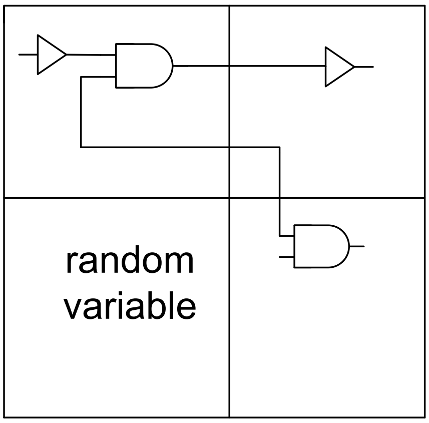


Figure 2.2: The Grid model.

ity structures. In other words, it is globally location-dependent. Devices have the similar characteristics than placed far away as it located close to each other. With increased process scaling, intra-die variations are becoming a more dominant portion of the overall variability of device features.

If we don't take the value of process variations into account, it will lead to underestimation. However, previous work considers the impact of process variations but did not consider spatial correlations in the statistical soft error rate. It may result in overestimation. Therefore, we need to consider the impact of spatial correlations in order to increase the accuracy of SSER, which can be witnessed in Figure 2.1.

According to Figure 2.1, circuit SER will overestimate circuit SER under the process variation 5% without considering spatial correlations. Circuit SER considers spatial correlations under the process variation 5% will not be overestimated comparing with the circuit SER under the process variation 5% without considering spatial correlations.

We propose an effective model considering spatial correlations of statistical soft error rate. The analysis is extended to include spatial correlations. Then we explain the model used for process variations and spatial correlations of intra-die variations.

There exists a few models in order to handle parameter correlations [17]. First, we introduce the Grid model shown in Figure 2.2. Grid model is a die area divided by square grids. A group of fully correlated devices is assumed to correspond to each square of

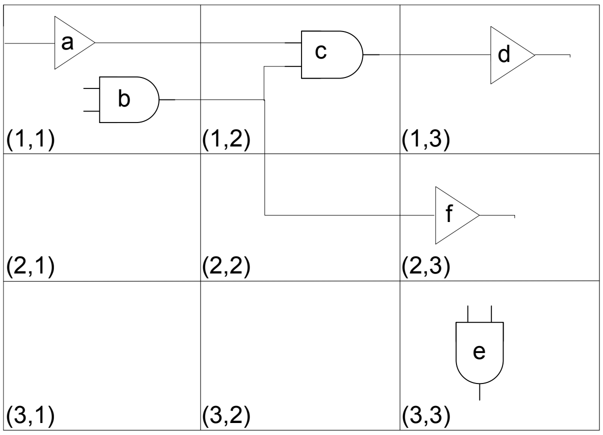


Figure 2.3: The gates in different grid have different process variations.

the grid. Each grid is modeled as an random variable which correlated with the random variables corresponding to the rest of the squares.

Another model is called the Quadtree model [18]. This method is recursively dividing the die area into four squares until individual gates into the grid. The partitions are stacked on top of another level. We then assign each of them an independent random variables. By summing all areas that cover this particular device, the random variable corresponding to the gate is computed. Due to sharing common random variables on higher levels, spatial correlations arise.

In this section, we use the Grid model to apply spatial correlations in soft error. We partition the region of die or reticle field into $nrow * ncol = n^2$ grids for modeling the intra-die spatial correlations of parameters. We assume that perfect correlations among the devices are in the same grid. Low or zero correlations are in far-away grids, and high correlations among those in close grids. The devices are more likely to have more similar characteristics than those placed far away due to they are close to each other.

For example, in Figure 2.3, the figure shows that gate a in grid (1,1) and gate e in grid (3,3). Since they are not in neighboring grids, we assume that their parameters may be uncorrelated. Gate c in grid (1,2), gate a and c lie in neighboring grids, and due to their spatial proximity, their parameter variations are not identical but highly correlated.

Our algorithm makes a second assumption. Assume that there is no correlations between different types of process parameters, and nonzero correlations may exist only among the same type of process parameters in different grids. For instance, the L_g values for transistors in nearby grids are correlated, but the other parameters such as W_g or W_{int} in any grid are uncorrelated. In other words, we assume that interconnect parameters in different layers to be different types of parameters.

2.1.2 SER estimation

We introduce the estimation of the overall SER in our framework. The overall SER for the circuit under test (CUT) can be computed by summing up the SERs of each individual node in the circuit. That is,

$$\text{SER}_{CUT} = \sum_{i=1}^{N_{FF}} \text{SER}_{ff_i} \tag{2.1}$$

where N_{FF} denotes the total number of flip-flops in the circuit under test.

Each SER_{ff_i} can be further formulated by integrating the products of *particle-hit rate* and the probability that a soft error can survive over the range $q = 0$ to Q_{MAX}. Therefore,

$$\text{SER}_{ff_i} = \sum_{j=1}^{N_{node}} \left(\int_{q=0}^{Q_{MAX}} (\text{freq}(q) \times \psi_{soft-err}(V_j, q))\text{d}q \right) \tag{2.2}$$

Here $\psi_{soft-err}(V_j, q)$ represents the probability that a transient fault originated from the particle of charge q at node V_j can result in one soft error at any flip-flop. N_{node} represents the nodes in the circuit. freq(q) represents the effective frequency for a particle hit of charge q in unit time according to [1] [4]. That is,

$$\text{freq}(q) = F \times K \times A \times \frac{1}{Q_s} \times \exp\left(\frac{-q}{Q_s}\right) \tag{2.3}$$

where F, K, A and Q_s denote the constants for neutron flux($>$ 10MeV), the technology-independent fitting parameter, the susceptible area in cm^2 and the charge collection slope, respectively.

2.1.3 Signal probability computation

$\psi_{soft-err}(V_i, q)$ depends on all three masking effects and can be further decomposed into

$$\psi_{soft-err}(V_i, q) = \sum_{j=1}^{N_{FF}} \mathrm{P}_{l-mask}(V_i, ff_j) \times \mathrm{F}_{elec}(V_i, ff_j, q) \tag{2.4}$$

where $\mathrm{P}_{l-mask}(i, j)$ denotes the overall signal probability of propagating the transient faults through all cells along the path from node i to flip-flop j. It can be computed by multiplying the signal probabilities of all cells as follows.

$$\mathrm{P}^{(k+1)}_{l-mask}(V_i, ff_j) = \mathrm{P}^{(k)}_{l-mask}(V_i, ff_j) \times \mathrm{P}^{(k)}_{non-control} \tag{2.5}$$

$$\mathrm{P}^{0}_{l-mask} = \mathrm{P}_{v(0)} \tag{2.6}$$

$P_{v(0)}$ represents the probability of the node be striked when signal is zero. Accordingly, $P_{l-mask}(V_i, ff_j)$ denotes the probability that all input signals of node v jointly determine such that the transient fault is not logically masked on this path.

The handling of reconvergent fanout nodes (RFONs) is an issue of computing signal probability whereas omitting it may cause considerable error [12]. In this work, a linear-time algorithm, dynamic weighted averaging algorithm (DWAA), is employed to consider the RFON effect and fix the signal probability. The main idea behind DWAA is to consider the dependency of signals between the fanout cone and the reconvergent node by forcing the reconvergent signals to the value corresponding to their respective fanins. More details of DWAA can be formed in [12].

2.1.4 Electrical probability computation

Electrical probability $\mathrm{P}_{elec}(V_i, ff_j, q)$ considers the electrical masking (e-mask) and timing masking (t-mask) effects and can be defined as

$$\begin{aligned}\mathrm{F}_{elec}(V_i, ff_j, q) &= \mathrm{P}_{t-mask}(pw_{i\to j}, w_j) \\ &= \mathrm{P}_{t-mask}(\xi_{e-mask}(V_i, ff_j, q), w_j) \end{aligned} \qquad (2.7)$$

where P_{t-mask} is defined as follows.

Definition (P_{t-mask}, **error-latching probability**)
Assume that the pulse width of one arrival transient fault and the latching window ($t_{setup} + t_{hold}$) of one flip-flop are random variables and denoted as pw and w, respectively. Let $x = pw - w$ be a new random variable where μ_x and σ_x are its mean and variance.

Then we can get the mean and sigma are $\mu_x = \mu_{pw} - \mu_w$ and $\sigma_x = \sqrt{{\sigma_{pw}}^2 + {\sigma_w}^2}$. The P_{t-mask} can be defined as following equation.

$$\mathrm{P}_{t-mask}(pw, w) = \frac{1}{t_{clk}} \int_0^{\mu_x + 3\sigma_x} x \times \mathrm{P}(x > 0) \times dx \qquad (2.8)$$

In Equation above, ξ_{e-mask} can be decomposed into two parts: δ_{strike} and δ_{prop}, respectively, represent the *first-strike* function and the *propagation* distribution function of transient faults.

2.1.5 Cell modeling

In this paper, we use the table-lookup Monte-Carlo framework. Since δ_{strike} and δ_{prop} are both non-linear functions of distributions, they are non-deterministic in nature and can only be only approximated by efficient and accurate models M_{strike} and M_{prop}. They are also the most critical components for an accurate SSER analysis framework due to the difficulty from integrating process variation impact. Therefore, to compute effectively SSER with process variation impact due to various sources of process variations, we adopt quality cell models from [21]. In [21], $M_{strike}$and M_{prop} are mapping functions, modeled into a form of look-up tables (i.e.$M_{strike}$and M_{prop}). Such models are important for enabling our accurate SSER framework.

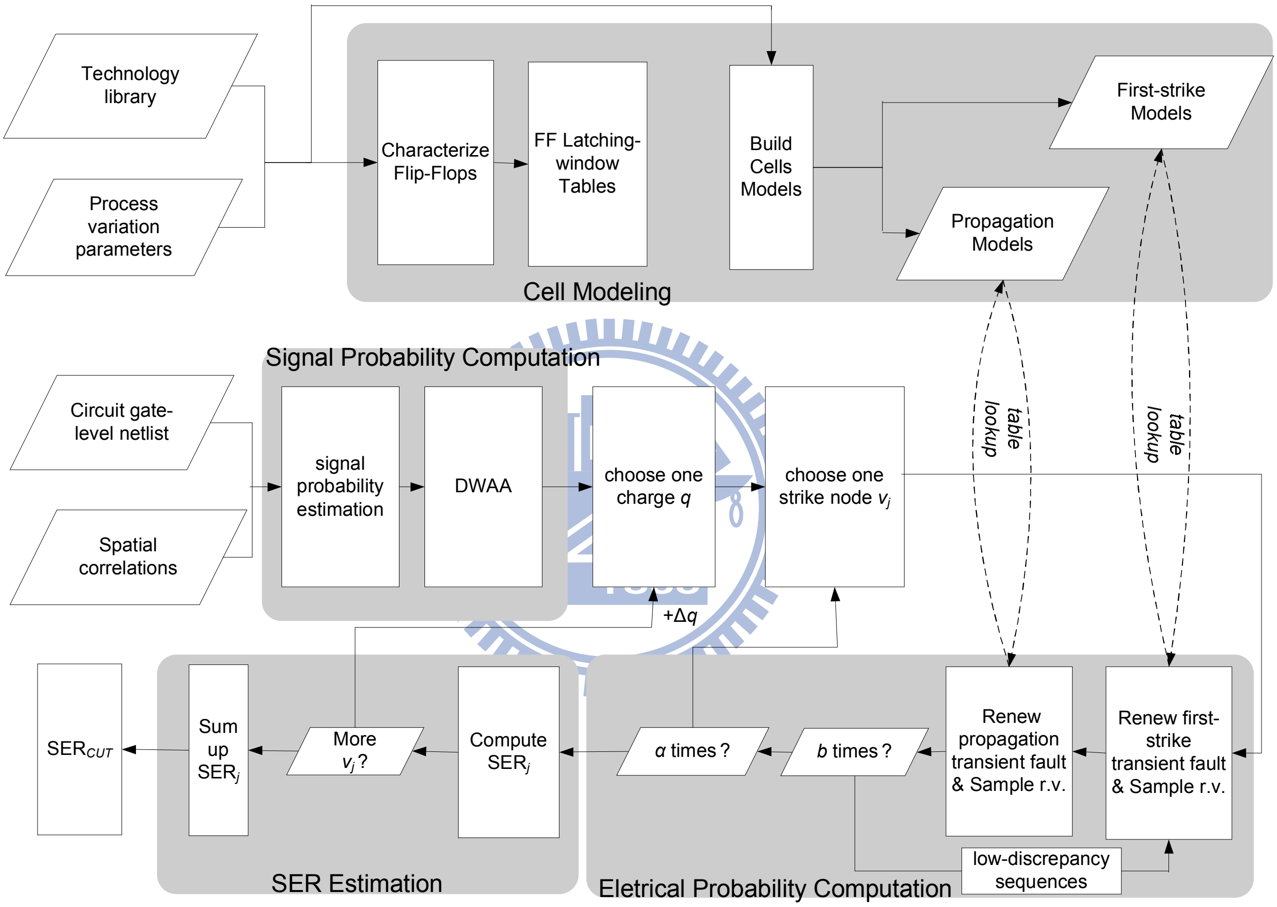

Technology library
Process variation parameters
Characterize Flip-Flops
FF Latching-window Tables
Build Cells Models
First-strike Models
Propagation Models
Cell Modeling
Signal Probability Computation
Circuit gate-level netlist
Spatial correlations
signal probability estimation
DWAA
choose one charge q
choose one strike node v_j
table lookup
table lookup
$+\Delta q$
SER_{CUT}
Sum up SER_j
More v_j ?
Compute SER_j
α times ?
b times ?
Renew propagation transient fault & Sample r.v.
Renew first-strike transient fault & Sample r.v.
low-discrepancy sequences
SER Estimation
Eletrical Probability Computation

Figure 2.4: The proposed SSER analysis framework

Chapter 3

Table-based Statistical Models

3.1 Table-based Statistical Models

M_{strike} and M_{prop} are respectively the generation and propagation models of pw that is a *random variable*. According to [2], pw follows the normal distribution, which can be written as:

$$pw \sim N(\mu_{pw}, \sigma_{pw}) \tag{3.1}$$

Therefore, we decompose M_{strike} and M_{prop} into four models: M^{μ}_{strike}, M^{σ}_{strike}, M^{μ}_{prop}, and M^{σ}_{prop} where each can be defined as:

$$M : \vec{x} \mapsto y \tag{3.2}$$

where $\vec{x}$ denotes a vector of *input variables* and y is called the model's *label* or *target value*. For M^{μ}_{strike} and M^{σ}_{strike}, we use input variables including charge strength, driving gate, input pattern, and output loading. For M^{μ}_{prop} and M^{σ}_{prop}, we use input variables including input pattern, pin index, driving gate, input pulse-width distribution (μ^{i-1}_{pw} and σ^{i-1}_{pw}), propagation depth, and output loading.

To build these models, a traditional approach is to construct tables according to manually-selected corner cases. However, such approach has two difficulties: first, these models have a lot of input variables so that their combinations enumerating all corner cases are prohibitively expensive. Second, input variables such as input pulse-width distribution are dependent variables in nature, which cannot be specified directly according to pre-selected combinations. Therefore, we use a different approach, as shown in Figure 3.1, consisting of 3 steps: *random sample generation*, *table fill-up*, and *table lookup*.

3.1.1 Random sample generation

We use a unified Monte Carlo SPICE simulation framework to build the two kinds of models (M_{strike} and M_{prop}) of distinct mapping spaces, as illustrated by Step 1 of Figure 3.1. The framework first generates a random path loaded with additional random cells. A charge is then injected as a current source at the beginning of the path according to the following equation [3]:

$$I(q, t) = \frac{q}{\tau_\alpha - \tau_\beta} \times (e^{-\frac{t}{\tau_\alpha}} - e^{-\frac{t}{\tau_\beta}}) \tag{3.3}$$

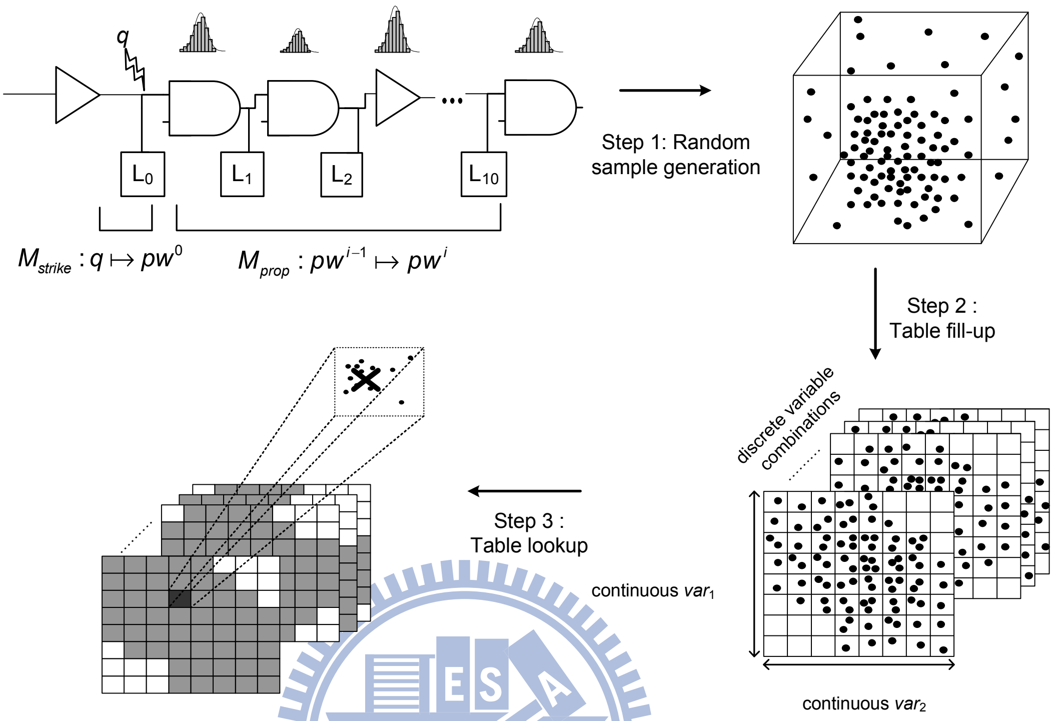

q
L₀
L₁
L₂
L₁₀
$M_{strike} : q \mapsto pw^0$
$M_{prop} : pw^{i-1} \mapsto pw^i$
Step 1: Random sample generation
Step 2 : Table fill-up
discrete variable combinations
continuous var_1
continuous var_2
Step 3 : Table lookup

Figure 3.1: Construction of table-based models

In each Monte Carlo instance, the pulse-width distributions are recorded along the path, which are later collected separately for different models. Note that this framework can be applied to all sources of process variations, as long as each of their impacts can be reflected using SPICE simulation. Also, to build accurate models, it is essential to acquire sufficiently large amount of samples in this step; in our case, for example, 500K.

3.1.2 Table fill-up

In Step 2 of Figure 3.1, we classify all samples according to their corresponding input variables to fill up the tables. For discrete variables such as charge strength, driving gate, input pattern, pin index, propagation depth, and output loading (in terms of equivalent-INVs), this can be done directly, which is like having multiple slices of tables, as illustrated in Figure 3.1.

For continuous variables such as the width and height of input pulse, however, we must discretize them to form a number of table cells. It can be done through determining the upper/lower bounds and the number of partitions. For the two bounds, we use the MIN and MAX values of samples sharing the same discrete input variable combination. For the number of partitions, there is a trade-off between table resolution and size: with sufficient samples, a larger number of partitions leads to finer table resolution and accuracy, in expense of a larger table size.

To achieve the balance the table size and resolution, an estimate of the table error is:

$$\text{MEAN}_{C_i \in all\ cells}\left(\frac{\text{MAX}(C_i) - \text{MIN}(C_i)}{\text{MEAN}(C_i)}\right) \leq \hat{\epsilon} \tag{3.4}$$

C_i represents the samples within a specific cell; $\hat{\epsilon}$ represents the error rate threshold. MAX, MIN, and MEAN respectively represent the maximum, minimum, and mean of the sample labels of C_i. We iteratively increase the number of partitions and calculate the mean error estimate until it falls below the target threshold. In our case, we found good accuracy can be reached with the number of partitions no more than 25 for all tables.

3.1.3 Table lookup

After all samples are allocated into table cells, there are two types of cells: non-empty cells with a number of samples and empty cells with none. For non-empty cells, we calculate its lookup value according to the samples within. While there are many ways to do it, we found the *mean* a good and efficient representative.

For the lookup values of empty cells, a traditional approach would be extrapolating them from non-empty ones. However, under sufficiently large amount of random samples, it is very likely that the empty cells originate from unrealistic situations. For example, as in Step 3 of Figure 3.1, the empty cells are distributed only in the top-right and lower-left corners, representing the extremely flat and the extremely sharp transient faults, respectively. Although neither of the two kinds of transient faults exists in reality, accesses to these cells happen during the SSER analysis occasionally as a result of error propagation. In such cases, we use the lookup value of the nearest non-empty cell instead to offset the expected error.

Chapter 4

Monte-Carlo Analysis with Importance Sampling

4.1 Monte-Carlo Analysis with Importance Sampling

4.1.1 Standard Monte Carlo

For statistical simulation of circuits, Monte Carlo method has become the standard technique recently [22]. Our goal is to sample among the random variables in the most efficient manner. We review the simple sampling method called Standard Monte Carlo. By using Standard Monte Carlo, some sample points can be obtained. However, one Monte Carlo run may cost many SPICE simulations and need much time. Improvement on sample generator must be made to speed up Standard Monte Carlo. There is a more effective way to get more uniformly sample points. We then introduce two methods of quasi-Monte Carlo (QMC) and importance sampling, that are more effective to obtain uniform sample points.

4.1.2 QMC

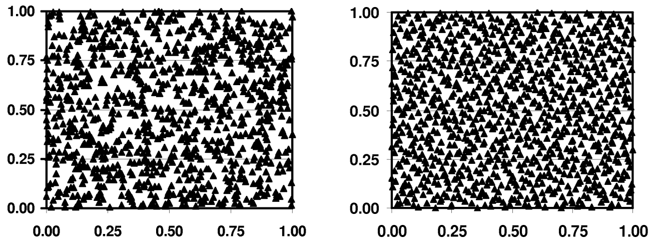
Pseudorandom number generation plays a key role to the success of the Monte Carlo method. Because pseudo-random sequence of numbers looks unpredictable, it looks like random numbers. However, by generating with deterministic algorithm pseudo random numbers are like the congruent random generator.

However, using *rand*() function for sampling points often suffers from the *clustering* problem [13] in high dimensional spaces.

Figure 4.1(a) illustrates this problem on an example of generating a (X,Y)-distribution by the Monte Carlo method using the *rand*() function. The sampling points are observed unevenly scattered among the (X,Y) plate, which means that these sampling points from pseudorandom generation may not be *representative* enough for the entire space.

The clustering problem motivates research of finding a deterministic sequence such that well-chosen points are distributed in the high-dimensional spaces uniformly. Such sequences are named *quasirandom* sequences.

Figure 4.1(b) shows the same number of sampling points using quasirandom sequences on the (X,Y) plate. Sobol algorithm [13] is used to generate the corresponding sequences. From Figure 4.1(b), new sampling points are observed more uniformly distributed over the (X,Y) plate and thus have better representativeness.



(a) (b)

Figure 4.1: Distributions from the Monte Carlo methods with random number generation and quasi-random sequences

Given a sampling number N and a dimension d, Monte Carlo methods converge with $O(1/\sqrt{N})$ simulation errors whereas QMC methods converge with $O(1/N)$ for optimal cases. Previous research works have demonstrated better results for QMC than MC methods for the problems with ≤ 360 dimensions in finance and physics.

Since each gate in the circuit becomes a free dimension (regardless of spatial correlations), the total dimension in the corresponding SSER system can be very high.

However, for a large d and moderate N, quasirandom sequences perform no better than the pseudorandom sequences [13]. Besides, high dimensional quasirandom sequences tend to suffer from the clustering problem again. In the worst cases, QMC's convergence rate, $O((lnN)^d/N)$, are even worse than MC's $O(1/\sqrt{N})$ as d goes larger. Therefore, we are motivated to apply importance sampling to ensure the effectiveness of the proposed QMC framework for SSER analysis.

4.1.3 Importance Sampling

Quasi-random numbers have an additional important property of being deterministically chosen based on equally distributed sequences. QMC methods can be viewed as deterministic versions of Monte Carlo methods [19] instead of working with random samples but deterministic points.

However, the quasi-random sequence applying appropriate transformations methods may be hard to obtain in multivariate distributions. It is because high dimensional quasirandom sequences tend to suffer from the clustering problem again. Therefore, we use importance sampling which is a variance reduction technique to conquer this problem.

The paper [20] proposes an approach combining IS and QMC. Importance sampling technique is one of the variance reduction techniques and needs to choose a good density function where to simulate a random variable. Considering a random variable Y has the distribution function r(y) and the probability density function q(y). Then we have $\int q\,(y)dy = 1$ and $\frac{q(y)}{q(y)}$ = 1

$$\int r\,(y)dy = \int \frac{r(y)}{q(y)} q\,(y)dy = \Gamma_q\left(\frac{r(Y)}{q(Y)}\right) \tag{4.1}$$

The difficulty of importance sampling is to choose a good sampling density function. First, proper density function q(y) should be proportional to r(y) to minimize Monte-Carlo variance. Then, q(y) should be chosen with the structure somewhat similar to $r(y) \cdot q(y)$. Generally, the density function should be larger than r(y) and q(y) and focuses on those sample points with significant contribution to the final result. A double-exponential density recommended from [20] is embeded for realizing importance sampling in our Monte-Carlo framework.

Chapter 5

Experimental Results

Table 5.1: Summary of first strike table error

error rate (%)				
cell	M^{μ}_{pw}	M^{σ}_{pw}	M^{μ}_{vm}	M^{σ}_{vm}
INV	0.99	2.6	0.44	7.3
AND	0.38	1.28	0.05	3.11
OR	0.84	3.34	0.11	8.75
Average	0.74	2.41	0.20	6.89

5.1 Experimental Results

We build and evaluate a series of table-based models accuratly. A series of table-based models are built and evaluated in accuracy. These models are then integrated into our SSER analysis framework to evaluate their SER estimation capability.

5.1.1 Model accuracy

We build the table-based models for three cells under 45nm technology. Assuming various process variations which the range is $\sigma_{proc} = \pm 15\%$, the models are built using 500K training samples. The total size of cell models in our experiments is 9.5MB. Then, we examine these models' accuracy using another 10K test samples.

The average errors of the models are summarized in Table 5.1 and Table 5.2 according to model types. Accordingly, two messages can be observed: (1) For M^{μ}_{strike}, and M^{μ}_{prop}, the models are highly accurate with average errors no more than 0.4%. For the M^{σ}_{strike}, M^{σ}_{prop} models, the average error is still within 4.1%. (2) In [2], the M^{μ}_{strike}, M^{μ}_{prop}, and M^{σ}_{prop} models have average errors up to 3.9%. For its M^{σ}_{strike} models, the average error further reaches 12.9%. In summary, our models exhibit much better quality.

5.1.2 SSER measurement

The proposed framework is implemented in C/C++ and exercised on a Linux machine with a Pentium Core Duo (2.4GHz) processor and 4GB RAM. The 45nm Predictive Tech-

Table 5.2: Summary of propagation table error

error rate (%)				
cell	M_{pw}^{μ}	M_{pw}^{σ}	M_{vm}^{μ}	M_{vm}^{σ}
INV	0.14	2.69	0.05	4.42
AND	0.09	2.54	0.18	2.91
OR	0.06	2.49	0.12	2.12
Average	0.10	3.38	0.18	3.15

nology Model (PTM) [14] is used for cell modeling. Each node under every input pattern combination is injected with four levels of electrical charges for all circuit: $Q_0 = 34fC$, $Q_1 = 66fC$, $Q_2 = 99fC$ and $Q_3 = 132fC$, where $32fC$ is observed to be the weakest charge capable of generating a transient fault with positive pulse width under the settings in our experiments.

Both circuit SER and SSER are measured and compared. For SER, we use static SPICE simulation; for SSER, we use Monte Carlo SPICE simulation as well as the proposed framework considering spatial correlation with (QMC) and without (MC) quasirandom sequences; and the Monte Carlo SPICE simulation as well as the proposed framework considering spatial correlation with (QMC + importance sampling) and without (MC) quasirandom sequences.

And for SSER, we add the variance-reduction technique which is importance sampling into our proposed framework to increase the coverage. Considering the extremely long runtime of Monte Carlo SPICE simulation (w/ 100 runs), we can only afford to perform tests on small circuits (i4, i6, i18 and c17), with the largest containing 7 gates, 12 strike nodes and 5 inputs. The runtime of the Monte Carlo SPICE simulation ranges from 8 hours to slightly more than one day. The runtime of our framework requires less than 1 second with an average of 10^6 speedup.

Table 5.3: The number of nodes and primary output in the circuits

circuit	N_{po}	N_{node}	circuit	N_{po}	N_{node}
c432	7	233	c5315	123	1806
c499	32	638	c6288	32	2788
c880	26	443	c7552	126	2114
c1355	32	629	m4	8	158
c1908	25	425	m8	16	728
c2670	157	841	m16	32	3156
c3540	22	901	m24	48	7234

5.1.3 SSER estimation on benchmark circuits

For charge strength Q2 = 99fC and Q3 = 132fC in Static (SPICE) or in Statistical (SPICE), very little difference can be found between static and statistical results. Therefore, we can take Q2 = 99fC and Q3 = 132fC as a static value and only run static SPICE once. As a result, almost an half of time in SPICE simulation is saved.

Table 5.3 lists the total number of nodes, the name, and the total number of outputs for each circuit in the first three columns. Table 5.4 reports the SSER values required by the MC, QMC and QMC-IS (QMC + importance sampling) frameworks considering spatial correlation, respectively. The last column compute the SSER difference, by comparing results from the MC frameworks considering spatial correlation. Moreover, Table 5.5 reports the runtime required by the MC, QMC and QMC-IS (QMC + importance sampling) frameworks considering spatial correlation, respectively. The last column compute the speedup, by comparing results from the MC frameworks considering spatial correlation.

From Table 5.3 and Table 5.4, SSER is clearly related to the number of nodes and primary outputs of a circuit, which correspond to the possibility of the circuit struck by radiation particles and the possibility of the transient faults observed at primary outputs, respectively. The runtime, however, depend not only on the number of strike nodes, but also the number of convolutions between nodes.

From Table 5.4, SSER difference is computed by $|SSER_{MC} - SSER_{QMC}|/SSER_{MC}$

Table 5.4: Benchmark circuits, SER from the baseline MC, QMC and QMC-IS frameworks considering spatial correlations

	MC	QMC		QMC-IS	
circuit	SSER (FIT)	SSER (FIT)	SSER diff. (%)	SSER (FIT)	SSER diff. (%)
c432	897.37E-05	908.53E-05	1.24	905.01E-05	0.85
c499	1102.24E-05	1161.77E-05	0.97	1082.18E-05	1.82
c880	1199.94E-05	1193.65E-05	0.52	1191.71E-05	0.69
c1355	1111.32E-05	1127.01E-05	1.41	1087.1E-05	2.18
c1908	907.23E-05	917.83E-05	1.17	866.17E-05	4.53
c2670	2988.66E-05	2992.9E-05	0.14	2992.7E-05	0.14
c3540	2113.85E-05	2090.39E-05	1.11	2122.44E-05	0.41
c5315	7845.95E-05	7862.43E-05	0.21	7848.76E-05	0.04
c6288	3733.71E-05	3661.51E-05	2.35	3656.12E-05	2.08
c7552	5929.5E-05	6263.61E-05	5.63	5905.00E-05	0.41
m4	828.2E-05	829.74E-05	0.19	786.3E-05	5.06
m8	1973.04E-05	1988.19E-05	0.77	1977.49E-05	0.23
m16	4409.17E-05	4550.25E-05	3.20	4459.3E-05	1.14
m24	6927.18E-05	7109.2E-05	2.09	7036.49E-05	2.56
Average			1.59		1.68

and the average of 1.59% difference implies that the QMC and MC frameworks are of the same quality. SSER difference is computed by $|SSER_{MC} - SSER_{QMC-IS}|/SSER_{MC}$ and the average of 1.68% difference implies that the QMC-IS and MC frameworks are of the same quality. And From Table 5.5, for all benchmark circuits, the overall speedup brought by QMC is 2.55X in average. For all benchmark circuits, the overall speedup brought by QMC-IS is 3.72X in average.

Table 5.5: Benchmark circuits, runtime from the baseline MC, QMC and QMC-IS frameworks considering spatial correlations

	MC	QMC		QMC-IS	
circuit	T_{MC} (sec)	T_{QMC} (sec)	speedup (X)	T_{QMC-IS} (sec)	speedup (X)
c432	145.20	44.76	3.24	31.04	4.68
c499	870.61	269.71	2.75	153.09	5.71
c880	174.43	49.62	3.51	31.93	5.46
c1355	913.07	280.46	3.26	198.36	4.60
c1908	341.71	139.59	2.45	103.46	3.30
c2670	463.91	142.52	3.26	96.11	4.83
c3540	1176.2	383.92	3.06	348.14	3.38
c5315	881.85	595.41	1.48	482.31	1.83
c6288	16111.8	4183.31	1.93	3671.84	4.39
c7552	1533.25	400.74	3.83	316.45	4.85
m4	114.23	47.65	2.4	37.159	3.07
m8	676.65	342.43	1.97	277.89	2.43
m16	9925.51	5636.89	1.76	2422.43	4.09
m24	37894.21	26687.6	1.42	10670.5	3.55
Average			2.55		3.72

Chapter 6

Conclusion

6.1 Conclusion

Due to the presence of process variation, all static techniques tend to unavoidably underestimate true SERs and the statistical SER analysis is built. In this paper, we adopt quality statistical cell models, based on which a Monte Carlo SSER framework is developed and consider spatial correlations into our framework. We further apply importance sampling to the framework for reducing variance and faster converging SSER. According to the experimental results, the average of SSER errors are within 1.68% compared to Monte Carlo SPICE simulations, more accurate than those from previous works. Furthermore, the use of quasi-random sequences and importance sampling demonstrates an average of 3.72X runtime improvement over the baseline MC framework considering spatial correlations while preserving the same SSER quality.

Bibliography

[1] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, ”Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic,” Proc. International Conference Dependable Systems and Networks (DSN), pp. 389-398, 2002.

[2] H. K. Peng, H. P. Wen, and J. Bhadra, ”On Soft Error Rate Analysis of Scaled CMOS Designs A Statistical Perspective,” Proc. International Conference ICCAD , pp. 157-163, 2009.

[3] R. Garg, C. Nagpal, and S. P. Khatri, ”A fast, analytical estimator for the SEU-induced pulse width in combinational designs,” Proc. Design Automation Conf. (DAC), pp. 918-923, 2008.

[4] M. Zhang and N. Shanbhag, ”A soft error rate analysis (SERA) methodology,” Proc. International Conference Computer Aided Design (ICCAD), pp. 111-118, 2004.

[5] R. Rajaraman, J. S. Kim, N. Vijaykrishnan, Y. Xie, and M. J. Irwin, ”SEAT-LA: a soft error analysis tool for combinational logic,” Proc. International Conference VLSI Design (VLSID), pp. 499-502, 2006.

[6] R. R. Rao, K. Chopra, D. Blaauw, and D. Sylvester, ”An Efficient Static Algorithm for Computing the Soft Error Rates of Combinational Circuits,” Proc. Design Automation and Test in Europe Conference (DATE), pp. 164-169, 2006.

[7] M. Zhang, T.M. Mak, J. Tschanz, K.S. Kim, N. Seifert, and D. Lu, ”Design for resilience to soft errors and variations,” Proc. International On-Line Test Symposium (IOLTS), pp. 23-28, 2007.

[8] N. Miskov-Zivanov and D. Marculescu, ”MARS-C: modeling and reduction of soft errors in combinational circuits,” Proc. Design Automation Conference (DAC), pp. 767-772, 2006.

[9] S. Krishnaswamy, I. Markov, and J. P. Hayes, ”On the role of timing masking in reliable logic circuit design,” Proc. Design Automation Conference (DAC), pp. 924-929, 2008.

[10] K. Ramakrishnan, R. Rajaraman, S. Suresh, N. Nijaykrishnan, Y. Xie, and M.J. Irwin, ”Variation impacts on SER of combinational circuits,” Proc. International Symposium Quality Electronic Design (ISQED), pp. 755-760, 2006.

[11] N. Miskov-Zivanov, K.-C. Wu, and D. Marculescu, ”Process variability-aware transient fault modeling and analysis,” Proc. International Conference Computer Aided Design (ICCAD), pp. 685-690, 2008.

[12] D. Franco, M. Vasconcelos, L. NAviner, and J.-F. Naviner, ”Signal probability for reliability evaluation of logic circuits,” Elsevier Microelectronics Reliability, pp. 1586-1591, 2008.

[13] W. J. Morokoff, and R. E. Caflisch, ”Quasi-random sequences and their discrepancies,” SIAM Journal on Scientific Computing (SISC), pp. 1251-1279, 1994.

[14] Predictive Technology Model, Nanoscale Integration and Modeling Group, http://www.eas.asu.edu/ ptm/, 2008.

[15] H. Chang and S. S. Sapatnekar, ”Statistical Timing Analysis Under Spatial Correlations,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 1467-1482, 2005.

[16] J. Jaffari and M. Anis, ”Advanced Variance Reduction and Sampling Techniques for Efficient Statistical Timing Analysis,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 1894-1907, 2010.

[17] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, ”Statistical Static-Timing Analysis: From Basic Principles to State of the Art,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 589-607, 2008.

[18] A. Agarwal, D. Blaauw, and V. Zolotov, ”Statistical timing analysis for intra-die process variations with spatial correlations,” Proc. International Conference Computer Aided Design (ICCAD), pp. 900-907, 2003.

[19] H. Niederreiter, ”Random Number Generation and Quasi-Monte Carlo Methods,” CBMS-NSF Regional Conference Series in Applied Mathematics, 1992.

[20] W. Hormann and J. Leydold, ”Quasi Importance Sampling,” http://epub.wu-wien.ac.at/english/, 2005.

[21] Y. H. Kuo , H. K. Peng, and H. P. Wen, ”Accurate Statistical Soft Error Rate (SSER) Analysis Using A Quasi-Monte Carlo Framework With Quality Cell Models,” Proc. International Symposium Quality Electronic Design (ISQED), pp. 831-838. 2010.

[22] A. Singhee, and R. A. Rutenbar, ”Why Quasi-Monte Carlo is Better than Monte Carlo or Latin Hypercube Sampling for Statistical Circuit Analysis,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 1763-1776, 2010.