國 立 交 通 大 學

電信工程研究所

碩 士 論 文

準確的軟性錯誤率分析

Accurate Statistical Soft Error Rate (SSER) Analysis

研究生：郭雨欣

指導教授：溫宏斌 教授

中 華 民 國 九 十 九 年 三 月

準確的軟性錯誤率分析

Accurate Statistical Soft Error Rate (SSER) Analysis

研 究 生：郭雨欣　　Student：Yu-Shin Kuo

指導教授：溫宏斌　　Advisor：Hung-Pin Wen

國 立 交 通 大 學

電 信 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Communication Engineering

College of Electrical and Computer Engineering National

Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Communication Engineering

March 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年三月

準　確　的　軟　性　錯　誤　率　分　析

學生：郭雨欣　　指導教授：溫宏斌 教授

國立交通大學電信工程研究所碩士班

摘　　要

在 90 奈米製程以下，電路因為宇宙射線而產生軟性錯誤的影響越來越大。尤其在製程變異下，更需要用統計的方法去估計電路的軟性錯誤率。然而，因為缺少高品質的統計模型，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，我們考慮在在 90 奈米製程下，由於宇宙輻射線索引起的軟性錯誤。並且提出了一個高準確性查表法的統計模型，並利用蒙地卡羅去分析這些統計模型。我們更進一步探索如何使用似隨機的序列，已達到比較好的收斂並且增加速度。實驗結果顯示，我們可以在合理的時間內更準確的估計出軟性錯誤率。

Accurate Statistical Soft Error Rate (SSER) Analysis

student：Yu-Hsin Kuo　　Advisors：Dr. Hung-Pin Wen

Institute of Communication Engineering
National Chiao Tung University

ABSTRACT

For CMOS designs in sub 90nm technologies, statistical methods are necessary to accurately estimate circuit soft error rate (SER) considering process variations. However, due to the lack of quality statistical models, current statistical SER (SSER) frameworks have not yet achieved satisfactory accuracies. In this work, we consider the soft error effect caused by cosmic radiation in sub 90nm technologies, and present accurate table-based cell models, based on which a Monte Carlo SSER analysis framework is built. We further propose a heuristic to customize the use of quasirandom sequences, which successfully speeds up the convergence of simulation error and hence shortens the runtime. Experimental results show that this framework is capable of more precisely estimating circuit SSERs with a reasonable speed.

誌　　謝

這篇論文得以順利完成，首先要感謝的人是我的指導教授溫宏斌老師。在研究所這兩年以來，老師給予了我很多幫助。除了熱心的和我討論研究上的問題，並且教導我許多專業上的知識外此，老師也樂於和學生們分享生活上的經驗。這些都使我獲益良多並且日益成長。

接著要感謝煥凱學長和佳伶學姊。煥凱學長在我討論了許多研究上的問題並且提供我許多寶貴的意見，這些都帶給我莫大的幫助。佳伶學姊不只提供我許多有用的資料，當我有困難或是不順心時，陪我一起發洩抱怨談心。此外，CIA 實驗室的成員，振源、千慧、彥后、韋廷、家慶、心恬、明經，謝謝你們的陪伴讓我的研究生活多采多姿。最後要感謝摯友綺瑩、亭蓉，你們的陪伴就是給我最大的鼓勵。

最後，僅以此文獻給我最愛的父母和姊姊。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recently, process variations that worsens in sub-90nm technologies have brought a paradigm shift to soft-error research, and it have grown to be one of the major challenges to scaled CMOS designs [2] [3]. The authors of [19] first investigate the various sources of process variations, and conclude that the traditional static approach will underestimate circuit SER in presence of process variations [20]. More specifically, according to Figure 1.1 from [17], static approaches will underestimate circuit SER by up to 50% under the process variation $\sigma_{proc} = 5\%$ ($\pm 3\ \sigma_{proc}$ covers 99.73% of the distribution), or over 100% under $\sigma_{proc} = 10\%$.Such an phenomenon represents that the impact of process variations to soft error analysis may no longer be ignored beyond deep sub-miciron era.However, although [20] and [17], respectively, proposes a symbolic- and statistical-learning-based frameworks for statistical SER (SSER) analysis, their SSER results are not accurate enough, where the main challenge comes from the difficulty of constructing quality cell models for transient-fault distributions.

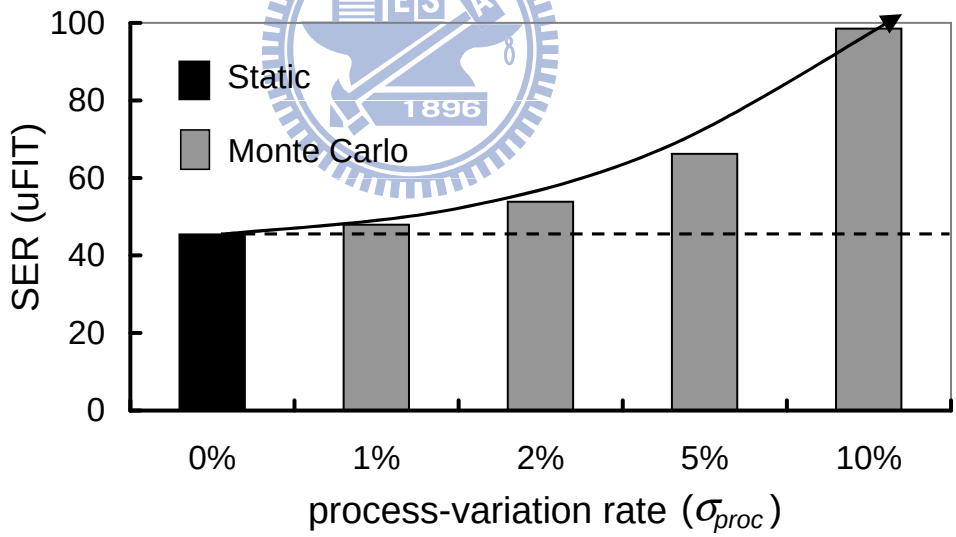


Figure 1.1: SER discrepancies between static and Monte Carlo SPICE simulation w.r.t. process-variation rates

1.2 Preliminaries and Previous Work

Formerly only concerned in memory, soft errors have emerged to be one of the major failure mechanisms for logic circuits in sub-90nm technologies. As predicted in [4] [5] [1], the soft error rate in combinational logic will be comparable to that of unprotected memory cells in 2011. Therefore, numerous studies have been dedicated to modeling of transient faults [7] [6] [12] [9], propagation and simulation/estimation of soft error rates [10] [11] [12] [13] and circuit hardening techniques including detection and protection [27] [26] [14].

It is important to first define the causes and effects of single-event upsets in Figure1.2 [28]. SEEs (single-event effects) are associated with the change of states or transients in a device that energetic external radiation particles induce. The SEEs can be classified into soft errors or hard errors. Soft errors are nondestructive, because resetting or rewriting the device restores normal behavior; however, hard errors are permanent. A common example of a hard error is an SEL (single- event latch-up).

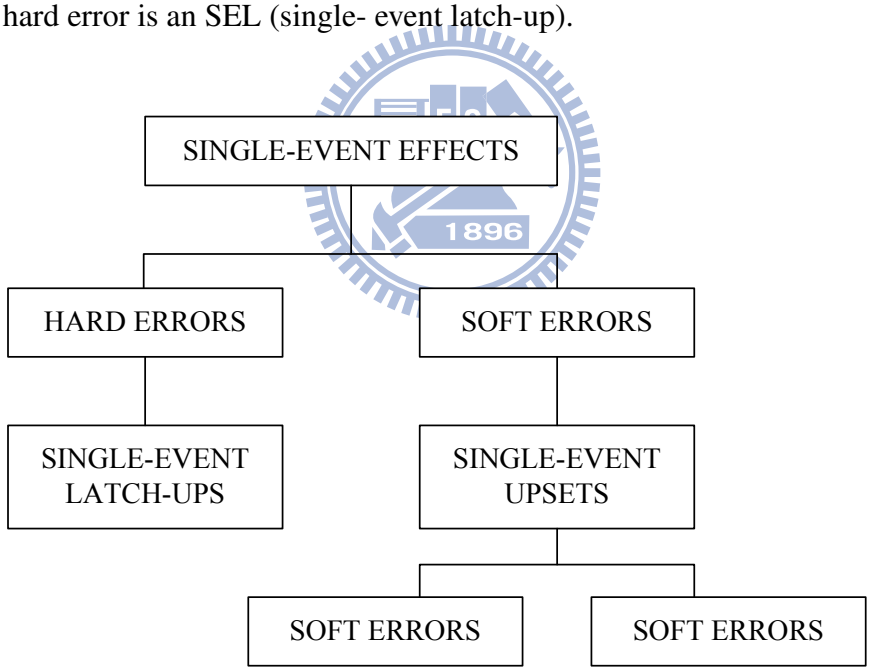


Figure 1.2: The cause and effects of single-event errors

Soft errors are also known as SEUs, causing transient or inconsistent error in data that are unrelated to components or manufacturing failures. Intrinsic noise and interference can also cause SEUs; however, design engineers can accommodate these causes. SEUs can be

further classify into SBUs (single-bit upsets) or MBUs (multiple-bit upsets). SBU refers to the flipping of one bit due to the passage of a single energetic radiation particle, where the physical separation from any other flipped bit is at least two memory cells. MBU refers to the flipping of several elements due to the passage of one or more radiation particles.

Soft errors occurred when the electric charges created by absorption of high-energy radiation are collected by the critical nodes in the circuit. The four common sources of soft errors are low energy alpha particles, high energy cosmic particles, thermal neutrons, and poor system design. These all cause reliability problems of electronics.

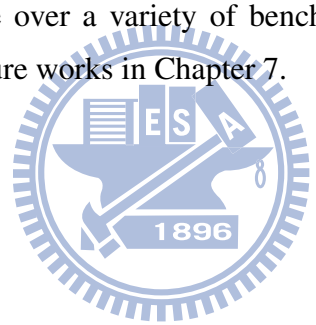
To achieve a reasonable balance between reliability and performance, we must estimate the overall soft error rate of the devices. Soft error rate(SER) is an measurement of reliability and availability, and it is commonly expressed by failure-in-time, or FIT rates, where 1 FIT means 1 failure per 10^9 hours of operation.

Consequently, soft error rate (SER) has become a key metric for circuit reliability and been extensively investigated. SERA [10] computes SER by means of a waveform model to consider the electrical attenuation effect and error-latching probability while ignoring logical masking. Whereas MARS-C [15] applies the symbolic technique to both logical and electrical maskings and scales the error probability according to the specified clock period, An SER [16] applies signature observability and latching window computation for logical and timing maskings to estimate SER for circuit hardening. SEAT-LA [12] and the algorithm in [13] simultaneously characterize cells, flip-flops and propagation of transient faults by waveform models and result in good SER estimate when comparing to SPICE simulation.

1.3 Organization of Our Framework

In this work, we first build accurate table-based models for transient-fault distributions, according to which a Monte Carlo SSER analysis framework is built. Further, we propose a heuristic to customize the use of quasirandom sequences, which successfully speed up the convergence of simulation error and hence shorten the runtime. From the experimental results, the framework is capable of yielding more accurate SSER results compared to previous works with reasonable speed.

The rest of this paper is organized as follows. In Chapter 3, we presents the SSER analysis framework. In Chapter 4, the generation of our table-based cell models is detailed. Then, we propose a heuristic of using quasirandom sequences to speed up the framework in Chapter 5. Chapter 6 describes the experimental results, including the accuracy of our models, the Monte Carlo convergence with and without quasirandom sequences, and the SSERs as well as runtime over a variety of benchmark circuits. Finally, we draw our conclusion and outline future works in Chapter 7.



Chapter 2

Background

2.1 The Basic Form SER & Charge to Voltage Pulse Model

The basic form of SER model in [1] is as follow:

$$SER \propto F \times A \times \exp\left(-\frac{Q_{CRIT}}{Q_S}\right) \tag{2.1}$$

where

F is the neutron flux with energy $> 1MeV$, in particles/(cm^2*s)

A is the area of the circuit sensitive to particle strikes, in cm^2

Q_{CRIT} is the critical charge, in fC, and

Q_S is the charge collection efficiency of the device, in fC

Both Q_{CRIT} and Q_S decrease with decreasing feature size. From equation2.1, the value of Q_{CRIT} relative to Q_S have large impact on SER, and is proportional to the sensitive area A. This model estimates SER due to atmospheric neutrons for a range of submicron feature sizes.

When a particle strikes a sensitive region of circuit elements, it will produce current pulse , and it is traditionally described as a double exponential function [9] as follows:

$$I(q,t) = \frac{q}{\tau_\alpha - \tau_\beta} \times (e^{-\frac{t}{\tau_\alpha}} - e^{-\frac{t}{\tau_\beta}}) \tag{2.2}$$

where

q is the amount of charge deposited as a result of the ion strike

τ_α is the collection time constant for the junction

τ_β is the ion track establishment time constant

Time constants τ_α and τ_β depends upon several process related parameters, and typically τ_α is on the order of 200ps, and τ_β is on the order of tens of picoseconds.

2.2 Three Masking Mechanisms

Whether the soft errors result from radiation-induced transient faults latched by state-holding elements depends on three masking effects *logical*,*electrical* and *timing* maskings [1]. As shown in Figure 2.1, logical masking occurs when the input value of one cell blocks the propagation of the transient fault under one input pattern. One transient fault attenuated by electrical masking may further disappear due to cell's electrical properties. Timing masking occurs when the survival transient faults arrives one state-holding element outside its window of clock transition. These three masking effects will lower the SER; however, these effects will decline continually in the Deep Sub-Micron(DSM) technology.

Subject to the three mechanisms, numerous researches are presented to evaluate soft error for logic circuits. The work in [8] propagates transient faults through one gate according to the logic function and meanwhile uses analytical models to electrically evaluate the change of transient faults. A refined model is presented in [9] to incorporate non-linear transistor current, which is further applied to all gates with different charges deposited. A static analysis is also proposed in [16] for timing masking by computing backwards the propagation of the error-latching windows efficiently.

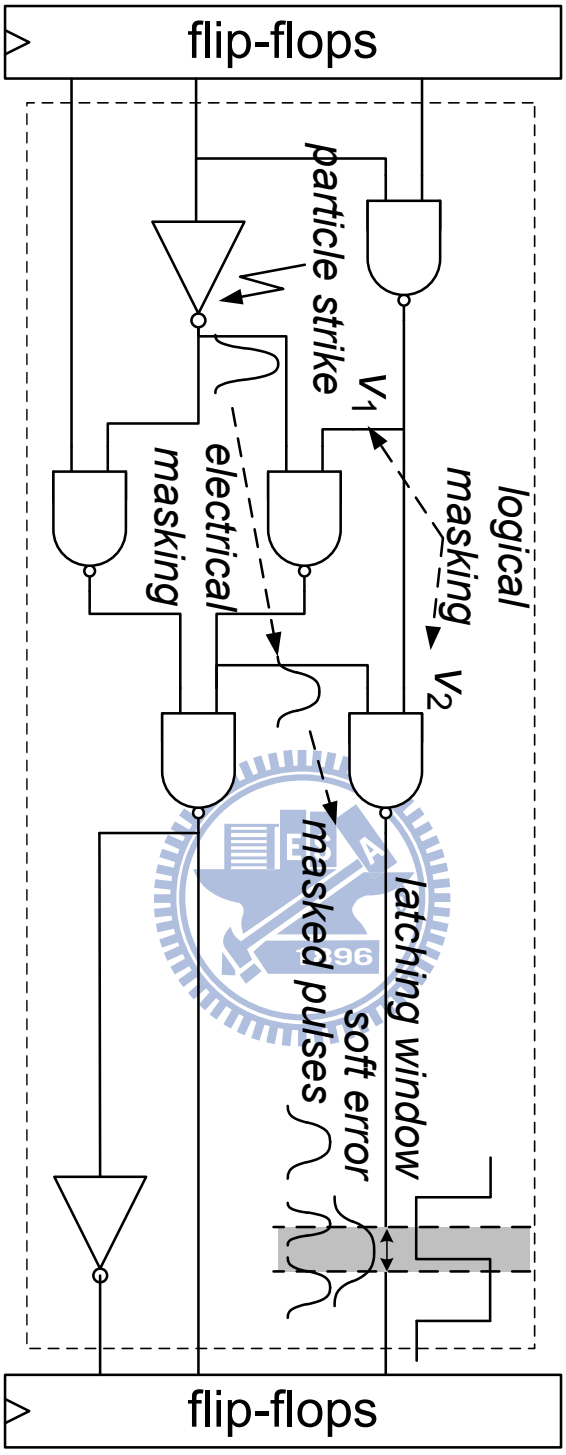


Figure 2.1: Three masking mechanisms for soft errors

2.3 Statistical View of Transient Faults

Under the process variation, transient faults have different statistical properties than before. We will discuss these from electrically and timing masking mechanisms, respectively.

The first observation is conducted on running Monte-Carlo SPICE simulation over a path considering 5% process variation on circuit geometry. In Figure 2.2, the radiation particle first strikes on the output of the NOT gate and then propagate the transient fault along two OR gates and two AND gates by setting up all side inputs properly. The pulse widths of the transient fault along the path are then measured in each Monte Carlo round, based on which a normal distribution curve are drawn using the empirical mean and standard deviation. The solid and dotted lines marks the empirical means before and after passing the current gate, respectively.

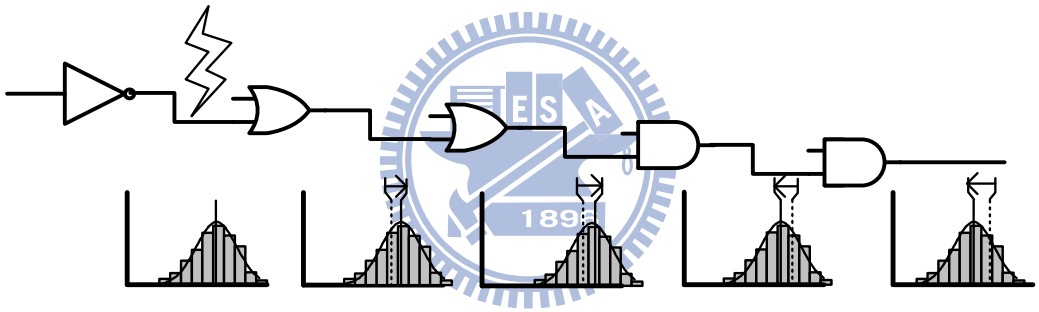


Figure 2.2: Monte Carlo SPICE simulation of a path

Along the path, the transient fault's pulse widths are approximated by normal distribution. We found that the mean of the pulse width goes larger(the first two stages), while sometimes it goes smaller(the last two stages). This phenomenon indicates that the pulse width of the transient fault is not always diminishing, which contradicts to some assumptions made in traditional static analysis. This phenomenon happens to the pulse height, too. The mean of the pulse height along the path dose not always attenuate, it goes larger sometimes.

The second observation is dedicated to the timing masking effect under process variations. In [12][29], the error-latching probability PL for one flip-flop is defined as

$$PL = \frac{pw - w}{t_{clk}} \tag{2.3}$$

where pw, w and t_{clk} denote the pulse width of the arrival transient fault, latching window of the flip-flop and the clock period, respectively.Statistically, however, process variations make pw and w become random variables. Therefore, we need to redefine Equation 2.3 as following.

Definition ($\mathrm{P}_{err-latch}$, **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. The latch probability is defined as:

$$\mathrm{P}_{err-latch}(pw, w) = \frac{1}{t_{clk}} \int_0^{\mu_x + 3\sigma_x} x \times \mathrm{P}(x > 0) \times dx \tag{2.4}$$

With the above definition, we further illustrate the contribution of process variations on SER analysis from two parts of Equation 2.3, $P(x > 0)$ and x, respectively. Figure 2.3 shows the 3 distributions for the transient fault with the same mean (95ps) for pulse widths under different process variation rates (σ = 1%, 5% and 10%). Given 100ps as the mean latching window of the flip-flop denoted as the solid line, the cumulative probabilities for $pw > w$ under 3 different σ are 17%, 40%, and 49%, respectively. Here we observe that the largest process variation rate corresponds to the largest $P(x > 0)$ in Equation 2.3. Besides, in Figure 2.3(b), we compute the pulse-width averages for the portion $x = pw - w > 0$ and they are 1, 13 and 26, respectively. Again, the distribution under σ = 10% has the largest x in Equation 2.3.

These two effects jointly explain the increasing discrepancy in Figure1.1. In summary, process variations make traditional static analysis no longer effective and should be considered in order to estimate SER accurately.

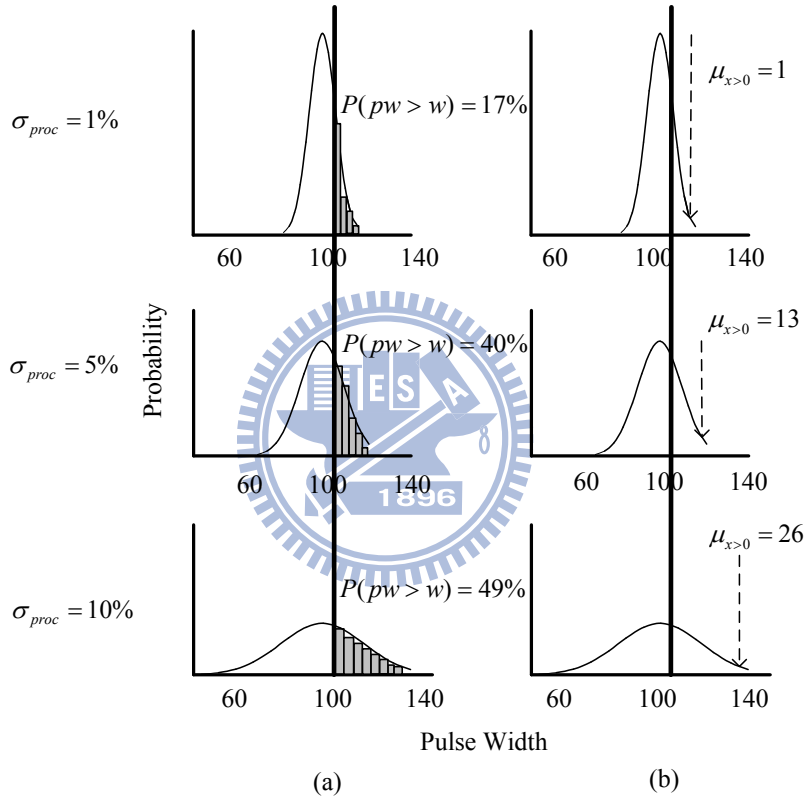


Figure 2.3: Latching probability vs. process variation

Chapter 3

SSER Analysis Framework

In this section, we describe the SSER analysis framework that considers process-variation impacts for cell-based designs. The proposed framework is illustrated in Figure 3.1 and mainly consists of four stages: (1) cell modeling, (2) electrical probability computation, (3) signal probability computation and (4) SER estimation. A stage-by-stage explanation of each component will start reversely from SER estimation to cell modeling.

3.1 SER Estimation

We will first 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 SER's of each individual node in the circuit. That is,

$$\text{SER}_{CUT} = \sum_{i=0}^{N_{node}} \text{SER}_i \tag{3.1}$$

where N_{node} is the total number of possible nodes to be struck by radiation particles in the CUT.

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

$$\text{SER}_i = \int_{q=0}^{Q_{MAX}} (\text{R}(q) \times \text{P}_{soft-err}(i, q))\text{d}q \tag{3.2}$$

Here $\text{P}_{soft-err}(i, q)$ represents the probability that a transient fault originated from the particle of charge q at node i can result in one soft error at any flip-flop. $\text{R}(q)$ represents the effective frequency for a particle hit of charge q in unit time according to [1] [10]. That is,

$$\text{R}(q) = F \times K \times A \times \frac{1}{Q_s} \times \exp(\frac{-q}{Q_s}) \tag{3.3}$$

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.

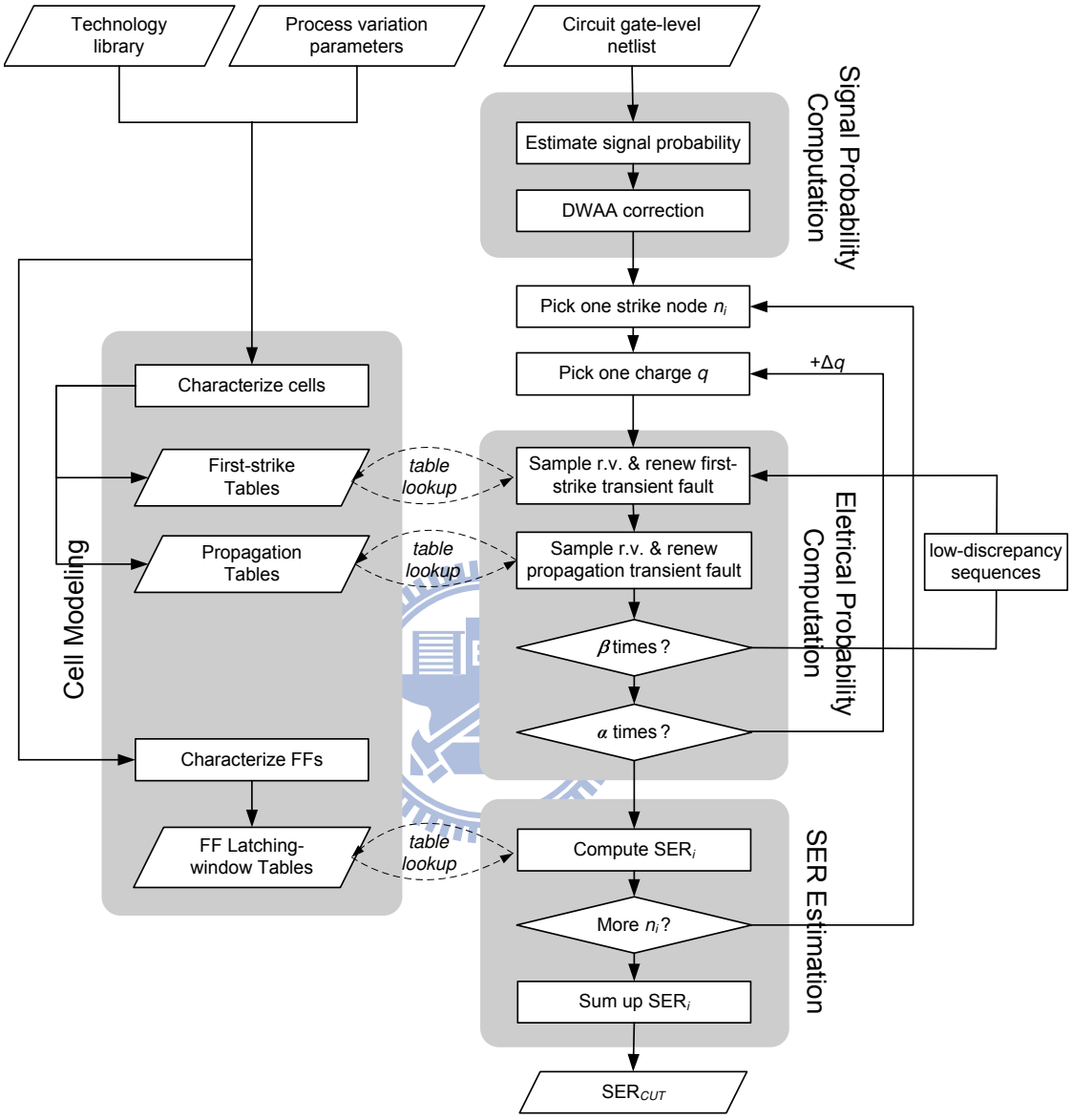


Figure 3.1: The proposed SSER analysis framework

3.2 Signal Probability Computation

$\mathrm{P}_{soft-err}(i,q)$ depends on all three masking effects and can be further decomposed into

$$\mathrm{P}_{soft-err}(i,q) = \sum_{j=0}^{N_{ff}} \mathrm{P}_{logic}(i,j) \times \mathrm{P}_{elec}(i,j,q) \tag{3.4}$$

where N_{ff} denotes the total number of flip-flops in the circuit under test. $\mathrm{P}_{logic}(i,j)$ 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.

$$\mathrm{P}_{logic}(i,j) = \prod_{k \in i \rightsquigarrow j} \mathrm{P}_{sig}(k) \tag{3.5}$$

where k denotes one node on the path $i \rightsquigarrow j$ and $\mathrm{P}_{sig}(k)$, accordingly, denotes the probability that all input signals of node k 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 [21]. 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.Here are the algorithm of DWAA [22] used in our framework.

1. Levelize the circuit and decide the order of RFONs to be processed. Those RFONs will be processed from lower to higher level. For those within the same level, whose branches reconverge at the lower level will be processed first.

2. Calculate the signal probabilities by means of the 0-algorithm, and refer these values as $p(j,0)$ which means the probability of node j after processed 0 RFON.

3. For each RFONs, calculate signal probability $p_f(j)$

$$P_f(j) = P(j/f=o)P(f=o) + P(j/f=l)P(f=l) \tag{3.6}$$

using as $p_f(j, t-1)$ which means the probability of element j in the fan-out cone f after having processing $(t-1)$ RFONs.

And update the value of $p(j)$ in each step as follows:

$$p_f(j,t) = \frac{p_f(j,t-1)w_s(j,t-1) + p_f(j)w_f(j)}{w_s(j,t-1) + w_f(j)} \tag{3.7}$$

where the weighting factor

$$w_(j) = |p_f(j) - p_j(0)| \tag{3.8}$$

which measures the deviation of $p(j)$ according to the influence of RFON under consideration,and

$$w_s(j,t-1) = \sum_{k=1}^{t-1} w_k(j) \tag{3.9}$$

represents the sum of all w_f on all the $(t-1)$ RFONs already processed.

4. Stop when t equals the number of RFONs.

The following example shows how signal probability calculation influences the estimation of SSER. Consider a sample circuit in Figure 3.2 with two RFONs i2 and n3. The comparison of signal probability calculation with DWAA and without DWAA to the exact signal probability by means of exhaustive simulation is given in Table 3.1. It can be seen that the SSER obtained using DWAA shows only 0.4% difference to the exact signal probability, while there is 3.3%difference without using DWAA. This is because DWAA takes into account RFONs, thus improve the accuracy of SSER.

3.3 Electrical Probability Computation

Electrical probability $P_{elec}(i, j, q)$ considers the electrical and timing masking effects and can be defined as

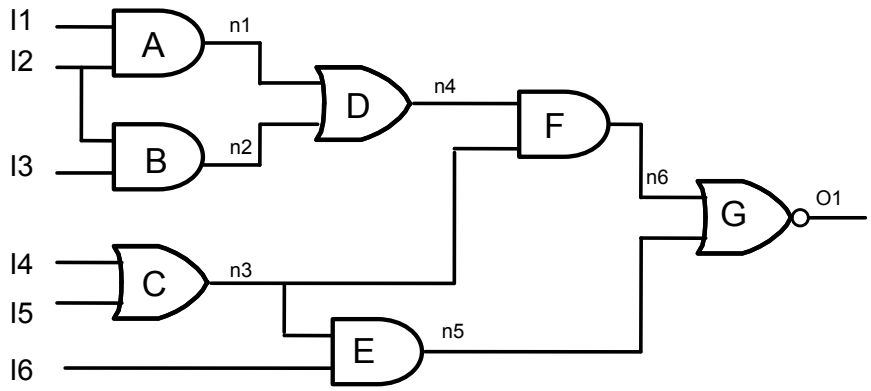


Figure 3.2: circuit example

Table 3.1: Comparison of signal probability estimation

signal probability									
method	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	$O1$	$SSER$	error rate
`Estimate`	0.25	0.25	0.75	0.437	0.375	0.328	0.42	49.24	3.29
`DWAA`	0.25	0.25	0.75	0.375	0.375	0.281	0.474	50.71	0.41
`Exact`	0.25	0.25	0.75	0.375	0.375	0.281	0.485	50.92	0

$$\begin{aligned} \mathrm{P}_{elec}(i,j,q) &= \mathrm{P}_{err-latch}(pw_j, w_j) \\ &= \mathrm{P}_{err-latch}(\lambda_{elec-mask}(i,j,q), w_j) \end{aligned} \tag{3.10}$$

While $\mathrm{P}_{err-latch}$ in equation 2.3 accounts for the timing making effect, $\lambda_{elec-mask}$ accounts for the electrical masking effect with the following definition.

Definition ($\lambda_{elec-mask}$, **electrical masking function**)
Given the node i where the particle strikes to cause a transient fault and flip-flop j is the destination that the transient fault finally ends at, assume that the transient fault propagates along one path $i \rightsquigarrow j$ through $v_0, v_1, ..., v_m, v_{m+1}$ where v_0 and v_{m+1} denote node i and flip-flop j, respectively.

$$\lambda_{elec-mask}(i,j,q) = \underbrace{\delta_{prop}(\cdots(\delta_{prop}(\delta_{prop}}_{m \text{ times}}(pw_0,1),2),\cdots),m) \tag{3.11}$$

where $pw_0 = \delta_{strike}(q,i)$.

In Equation (3.11), δ_{strike} and δ_{prop}, respectively, represent the *first-strike* function and the *propagation* distribution function of transient faults.

3.4 Cell Modeling

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}. As detailed in the next section, they are also the most critical components for an accurate SSER analysis framework due to the difficulty from integrating process-variation impacts.

Chapter 4

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 [17], pw follows the normal distribution, which can be written as:

$$pw \sim N(\mu_{pw}, \sigma_{pw}) \tag{4.1}$$

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:

$$M : \vec{x} \mapsto y \tag{4.2}$$

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 4.1, consisting of 3 steps: *random sample generation*, *table fill-up*, and *table lookup*.

4.1 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 4.1. 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 equation [9].

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.

4.2 Table Fill-up

In Step 2 of Figure 4.1, 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 4.1.

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:

$$\text{MEAN}_{C_i \in all\ cells} \left(\frac{\text{MAX}(C_i) - \text{MIN}(C_i)}{\text{MEAN}(C_i)} \right) \leq \hat{\epsilon} \tag{4.3}$$

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.

To promote the accuracy of our tables, we further adjust it according to the behaviors of transient faults observed. At the propagation depth one, the transient faults may have very large voltage drop compared to other propagation depth; otherwise, the transient faults have similar voltage variations. So we further separate our tables into two parts depend on whether its propagation depth is one, making our table more precise.

The Figure4.2 and Figure4.3 are our first strike and propagation tables. For first strike table, we use the input vectors, charge q and output loading as the index when the particle strikes on a node. In the process of propagation, we first decide whether the transient pulse is positive or negative, and combine with charge q and output loadings to form the labels of propagation tables. Next, we derive the transient pulses according to two continuous variables pw and vm.

4.3 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 4.1, 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.

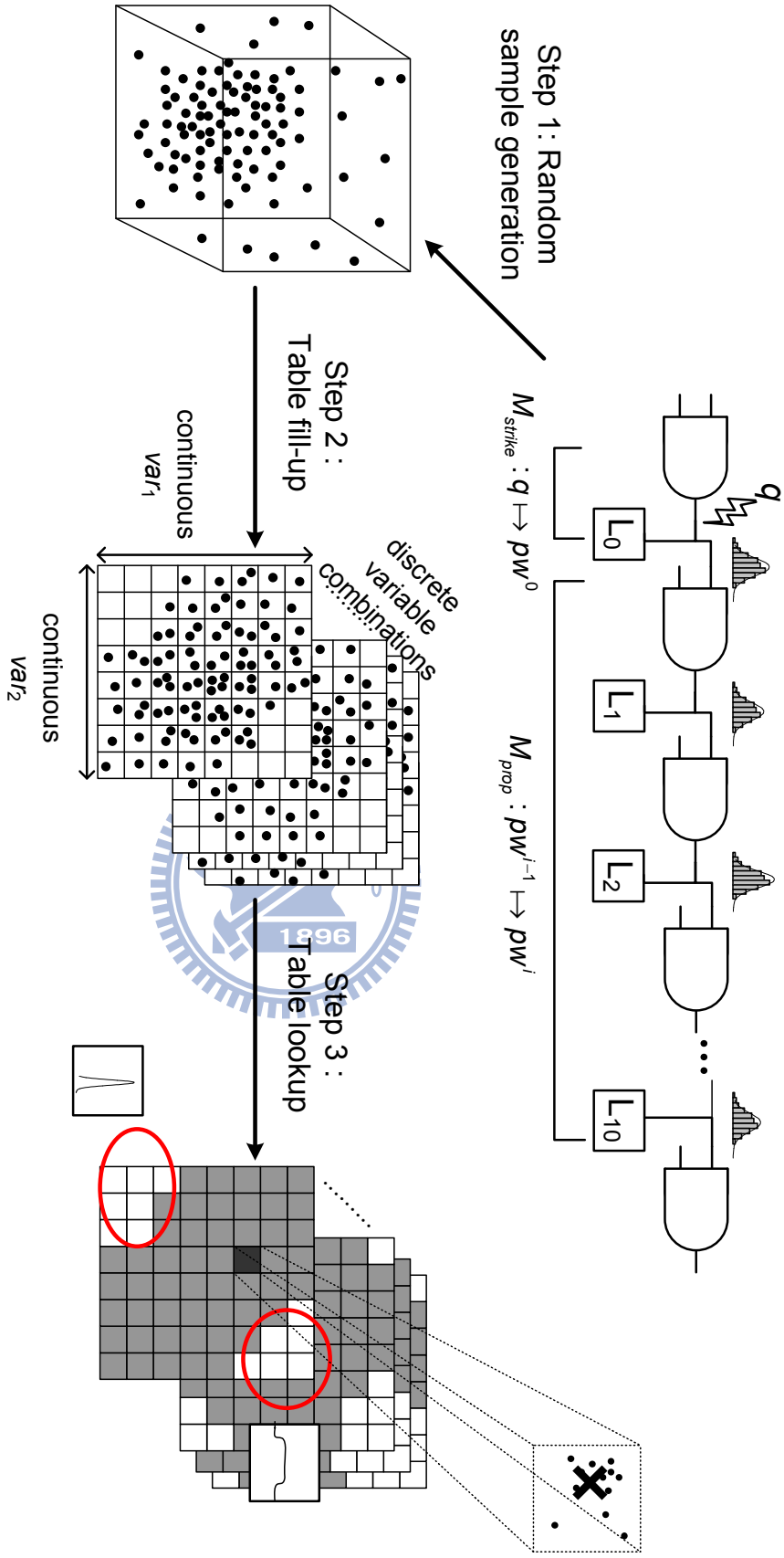


Figure 4.1: Construction of table-based models

	loading (equivalent INVs)					
vector	1	2	3	4	• • •	*k*
00 01 10	M^{μ}_{pw}	M^{σ}_{pw}	M^{μ}_{vm}	M^{σ}_{vm}		

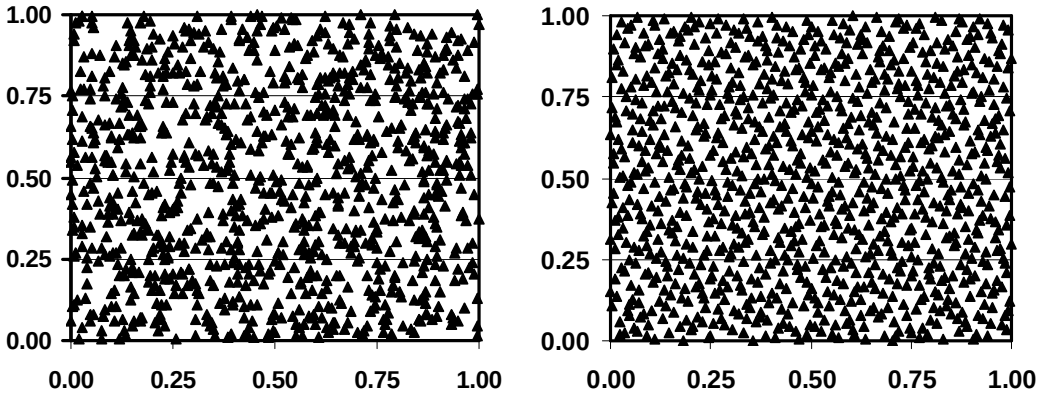
Figure 4.2: The first strike tables

D Q L	pw_o	pw_1	…	pw_m
vm_o	M^{μ}_{pw} M^{μ}_{vm} M^{σ}_{pw} M^{σ}_{vm}	M^{μ}_{pw} M^{μ}_{vm} M^{σ}_{pw} M^{σ}_{vm}		M^{μ}_{pw} M^{μ}_{vm} M^{σ}_{pw} M^{σ}_{vm}
vm_1				
⋮				
vm_m	M^{μ}_{pw} M^{μ}_{vm} M^{σ}_{pw} M^{σ}_{vm}	M^{μ}_{pw} M^{μ}_{vm} M^{σ}_{pw} M^{σ}_{vm}		M^{μ}_{pw} M^{μ}_{vm} M^{σ}_{pw} M^{σ}_{vm}

Figure 4.3: The propagation tables

Chapter 5

Using Quasirandom Sequences



(a) (b)

Figure 5.1: Distributions from the Monte Carlo methods with random number generation and quasirandom sequences

Pseudorandom number generation plays a key role to the success of the Monte Carlo method. However, using *rand*() function for sampling points often suffers from the *clustering* problem [23] in high dimensional spaces. Figure 5.1(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 not evenly scattered among the (X,Y) plate, which means that these sampling points from pseudorandom generation may not be *representative* enough for the entire space.

5.1 Concept of QMC

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 5.1(b) shows the same number of sampling points using quasirandom sequences on the (X,Y) plate. From Figure 5.1(b), new sampling points are observed more uniformly distributed over the (X,Y) plate and thus have better representativeness.

The quasi-random sequences, also called low-discrepancy sequences, in several cases permit to improve the performance of Monte Carlo simulations, offering shorter computational times and/or higher accuracy. In fact, the low discrepancy sequences are totally deterministic, so the name quasi-random can be misleading.

The concept of low-discrepancy is associated with the property that the successive numbers are added in a position as away as possible from the others numbers, that is, avoiding clustering (groups of numbers close to other). The sequence is constructed based on the idea that each point is repelled from the others. So, if the idea for the points is maximally avoiding of each other, the job for the numbers generated sequentially is to fill in the larger "gaps" between the previous numbers of the sequence. In this way, we can improve the uniformity of the sequences.

The van der corput sequence is the basic of one dimensional low discrepancy sequence. For different base b,the number n in van der corput sequence can be generated from three steps as follow.

1. Expand the decimal-base number n to base b.

2. Reflect the number in base b.

3. Get the reflected number in base b.

$$n = \sum_{j=0}^{m} a_j(n) b^j$$

van der corput sequence

$$b(n) = \Phi_b(n) a_j(n) b^{-j-1}$$

The figure 5.2 is an example of van der corput sequence with base 2 distributed over interval [0,1).

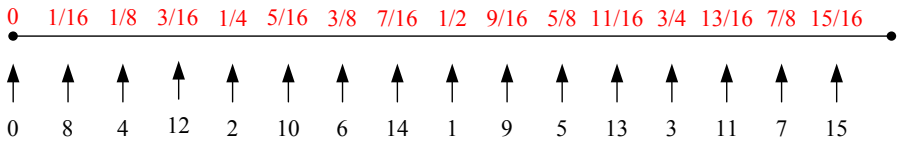


Figure 5.2: The first 16 number of van der corput sequence in interval [0,1)

5.2 Different Types of Low Discrepancy Sequences

The van der corput sequence is the basic one dimensional low discrepancy sequence. For multi-dimensional low discrepancy sequences, there are some famous algorithms. Three most common algorithms for computing quasirandom sequences include *Halton* algorithm, *Faure* algorithm and *Sobol* algorithm. However, because Halton and Faure sequences may suffer from the correlation problems in high dimension with longer computation time, Sobol algorithm is chosen as the quasirandom number generation scheme in our framework. Here outlines the Sobol $\mathcal{Z}^2$-algorithm[1]:

1. Given the sequence length N, choose a polynomial of degree d :

$$p_i(x) = x^d + c_1 x^{d-1} + ... + c_{d-1}x^1 + 1 \text{ where } i \leq N,$$

2. Assign r direction numbers $v_1, v_2, ..., v_d$ with :

$$1 \leq \frac{v_j}{2^{m-j}} \leq (2^j - 1) \text{ where } \frac{v_j}{2^{m-j}} \text{ is one odd integer}$$

That is, $v_j = 2^{m-j}(2^j - 1)$.

3. Compute other direction numbers ($j \leq m = lgN$) by recurrence :

$$v_j = c_1 v_{j-1} \oplus c_2 v_{j-2} \oplus \cdots \oplus c_d v_{j-d} \oplus \lfloor v_{j-d}/2^d \rfloor$$

4. Given the binary representation of n, compute the sequence for each dimension, recursively :

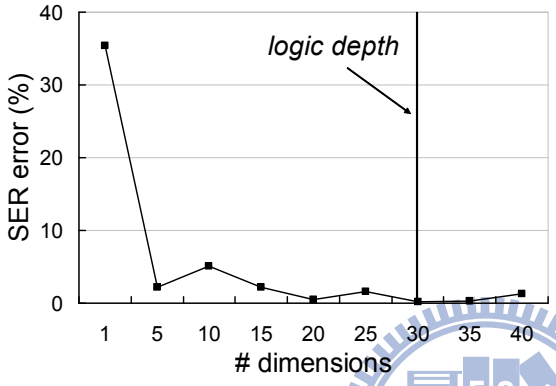
$$x_0^i = 0 \text{ and } x_k^i = \frac{(x_{k-1}^i \times 2^m) \oplus v_c}{2^m}$$

Monte Carlo methods with quasirandom sequences are termed Quasi-Monte Carlo (QMC) methods. 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.

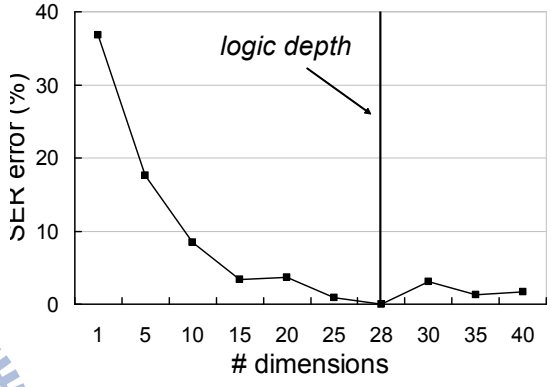
[1] 2 is the base number b that the user can specify.

5.3 Effective Dimensions in QMC

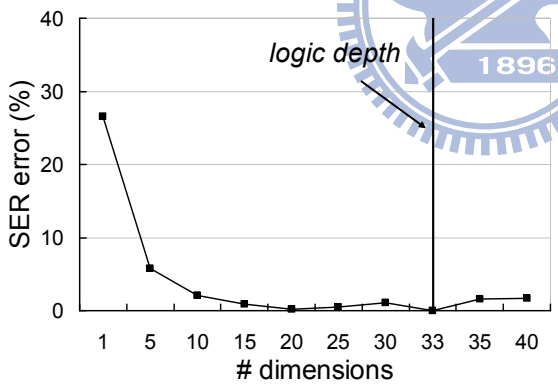
Since each gate in the circuit becomes a free dimension (regardless of spatial correlations), the total dimension in the corresponding SSER system can be very high. However, for a large d and moderate N, quasirandom sequences perform no better than the pseudorandom sequences [23]. Besides, high dimensional 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 dimension reduction to ensure the effectiveness of the proposed QMC framework for SSER analysis.



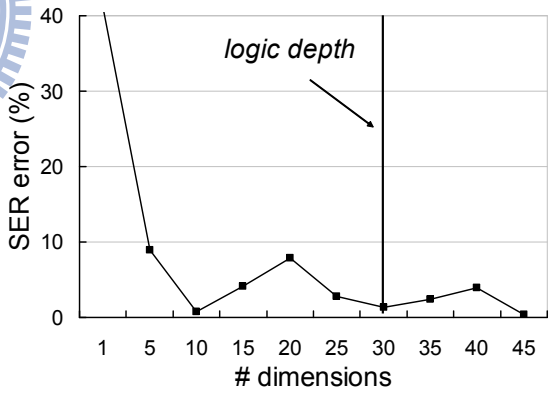
(a) c432



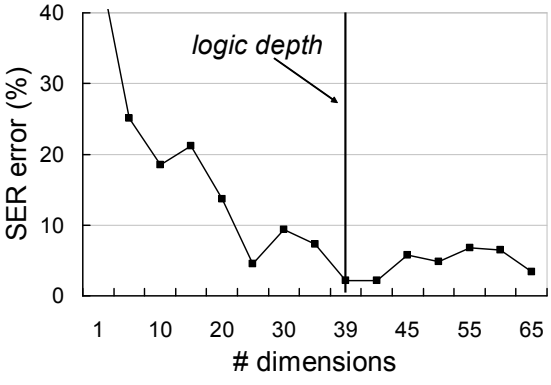
(b) c499



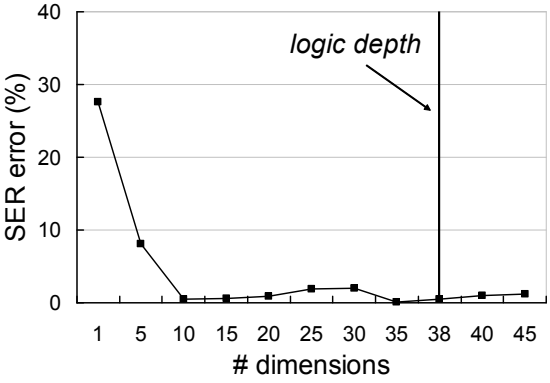
(a) c880



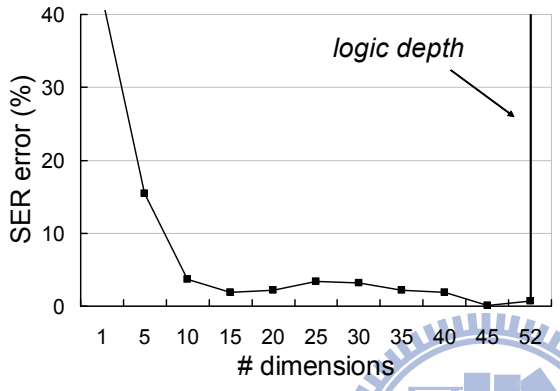
(b) c1355



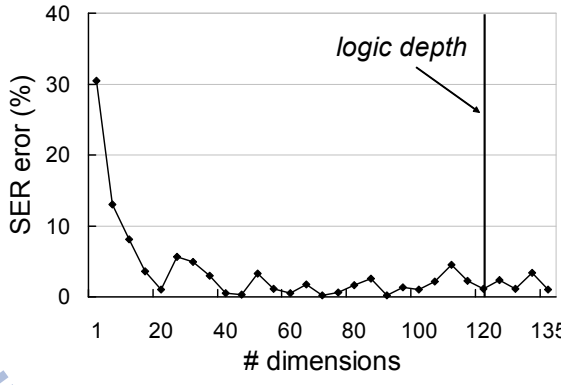
(a) c1908



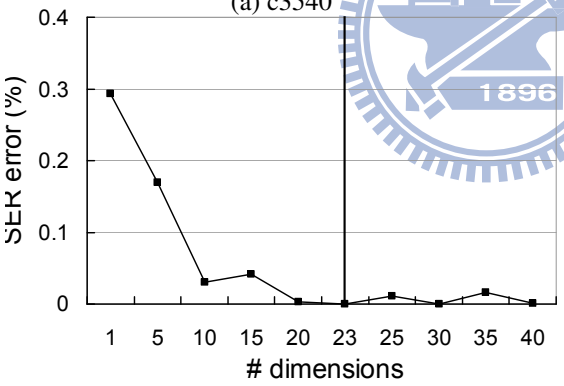
(b) c2670



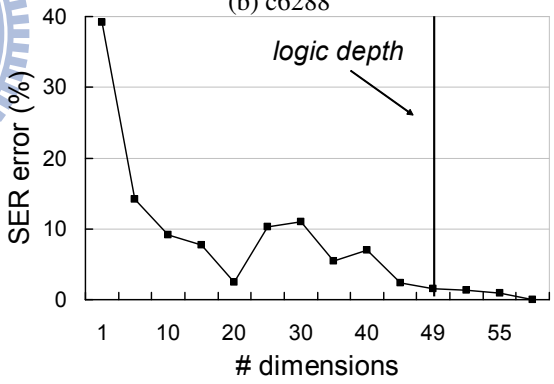
(a) c3540



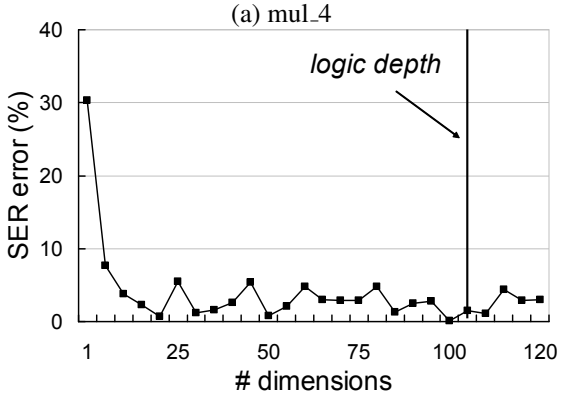
(b) c6288



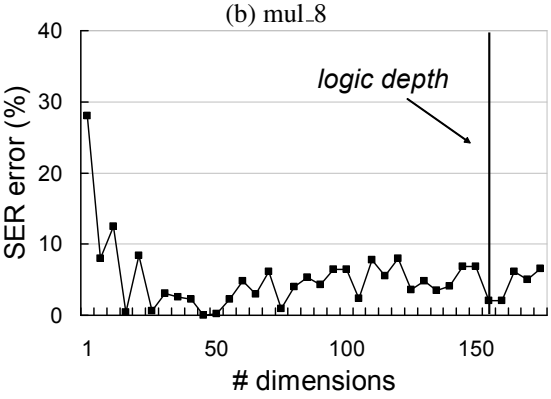
(a) mul_4



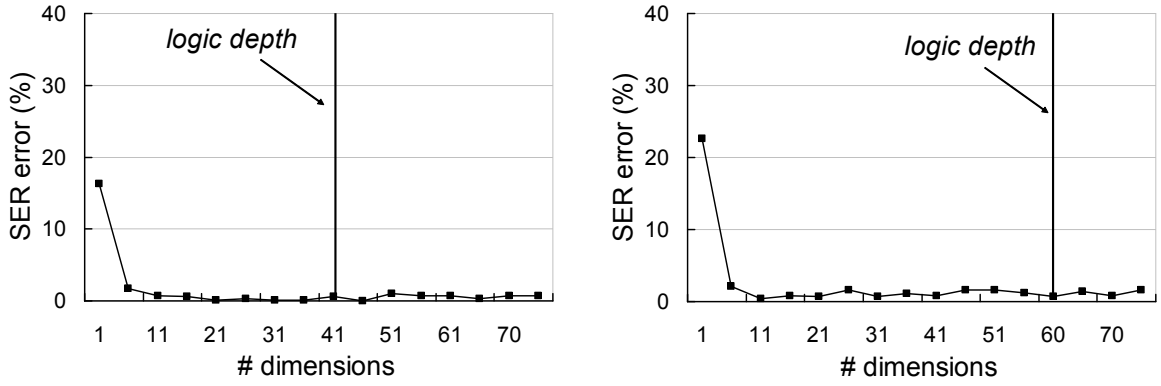
(b) mul_8



(a) mul_16



(b) mul_24



(a) c5315 (b) c7552

Figure 5.3: Convergence rate, dimension number, and logic depth of benchmark circuits

Effective dimensions of circuits can be observed through experiments. Figure 5.3 shows the convergence rates for four sample circuits where the vertical lines indicate the *logic depths* (a.k.a. levels) of each circuit. All convergence rates drops quickly as the dimension numbers increase. Such phenomenon implies their underlying SSER systems can be properly described using much lower dimensions. For example, the intuitive dimension number for the circuit c7552 is 2114, the total number of its nodes.From Figure 5.3(d), however, a dimension number of 60 is already good enough. Also, from Figure 5.3 states that the circuit level can suffice to represent the total dimension and thus converge SER faster. In Table 6.2 of the next section, more benchmark circuits are used to validate this hypothesis of using the circuit level as the reduced dimension.

Table 5.1: Comparison of dimension reduction methods

method	$level$	$sser(rand)$	$error(\%)$	$sser(level)$	$error(\%)$
i4	4	26.53	12.32	24.08	1.94
i6	3	38.92	4.20	37.56	0.01
i18	5	64.81	2.14	64.35	2.83
c17	5	62.88	2.77	62.98	2.94

5.4 A Heuristic for Dimension Reduction

Although we can assign the dimensions obtained from random function to each gate, there are some disadvantages of it. The random function may have higher probability of providing inferior results once the level of the circuit is small. That is, there may be dimension correlations among gates.

To solve this problem, we can use level number as each gate's dimension number. Since the correlations between each gate in the same level can be seen independently in the process of simulation, it is intuitive to assign each gate at different level with different dimensions. Otherwise, assign gates at the same level with the same dimension.It can be shown at Table5.1 that the dimension reduction method using level numbers have better accuracy compared to random decision method when the circuit's logic depth is too small.

Chapter 6

Experimental Results

Table 6.1: Summary of table error

error rate (%)				
cell	M^{μ}_{strike}	M^{σ}_{strike}	M^{μ}_{prop}	M^{σ}_{prop}
INV	0.35	0.19	0.38	1.07
AND	0.30	0.23	0.36	1.35
OR	0.39	0.21	0.37	2.07
Average	0.35	0.21	0.37	1.50

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.

6.1 Model Accuracy

We build the table-based models according to Figure 4.1 for three cells under 45nm technology. Assuming 5% process variation ($\sigma_{proc} = 5\%$), 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 6.1 according to model types. Accordingly, two messages can be observed: (1) For M^{μ}_{strike}, M^{σ}_{strike}, and M^{μ}_{prop}, the models are highly accurate with average errors no more than 0.4%. For the M^{σ}_{prop} models, the average error is still within 2.1%. (2) In [17], 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.

6.2 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 Technology Model (PTM) [24] is used for cell modeling. For all circuits, each node under every

input pattern combination is injected with four levels of electrical charges: $Q_0 = 34fC$, $Q_1 = 66fC$, $Q_2 = 99fC$ 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 with (QMC) and without (MC) quasirandom sequences. 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.

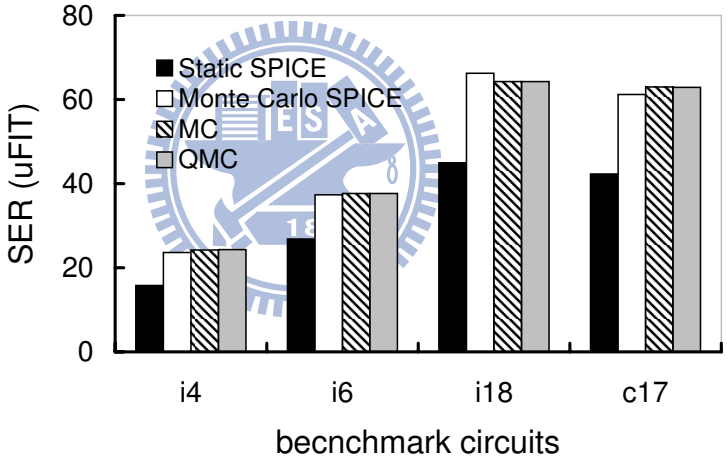


Figure 6.1: SSER comparison from static and Monte Carlo SPICE simulations, the proposed MC and QMC frameworks

Figure 6.1 compares the results from SPICE simulation and our frameworks. The three facts are observed: (1) Considering 5% process variations, the SSER obtained by Monte Carlo SPICE simulation are $35\% \sim 52\%$ above the SER obtained by static SPICE analysis (indicated by the black bars). Since the process variation worsens the stability of circuits beyond the deep submicron era, statistical effect should be considered to avoid increasingly underestimated circuit SER. (2) The proposed MC and QMC frameworks yield very similar SSERs with each others where the mismatches are within 0.4%. This means that we can

use the faster QMC without serious accuracy degradation. (3) Compared to the results of Monte Carlo SPICE simulation, the proposed QMC framework has error rates of 2.5%, 0.8%, 2.9%, and 2.8%, respectively. Compared to [20] and [17] where the error rates are around 10%, our framework is quite accurate, which can be well attributed to our models.

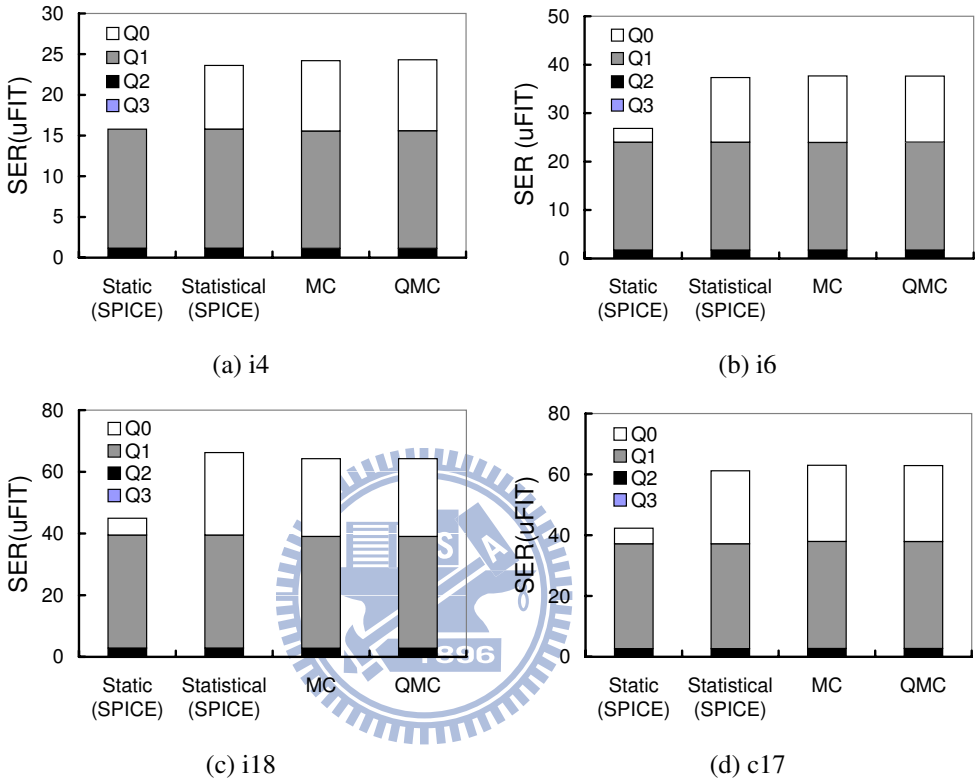


Figure 6.2: SER breakdown by charge strength

To more closely investigate the SER difference between static and statistical analysis, we breakdown the results in Figure 6.1 by charge strength levels, and present the results in Figure 6.2. Comparing the results between static and statistical SPICE simulations across all test circuits, it is observed that the results of the two SPICE simulations and the proposed framework are very similar for $Q_1 \sim Q_3$ parts (within 1% difference). However, the static SPICE simulation dramatically underestimate the SERs for the Q_0 part (indicated by the white bars), whereas the proposed framework gives much closer results with Q_0-part errors of 11.4%, 2.3%, 5.6%, and 4.1%, respectively. It also discloses that the analysis of weak charges is the most challenging task of SSER analysis.

Table 6.2: Benchmark circuits, SER and runtime from the baseline MC and QMC frameworks

circuit	N_{node}	N_{po}	MC		QMC		Comparison	
			SSER (E-05)(FIT)	T_{MC} (s)	SSER (E-05)(FIT)	T_{QMC} (s)	SSER diff. (%)	speedup (X)
i4	4	1	24.22	< 1	24.31	< 1	0.37	-
i6	6	2	37.66	< 1	37.65	< 1	0.03	-
i18	12	3	64.26	< 1	64.24	< 1	0.02	-
c17	12	3	63.00	< 1	62.89	< 1	0.17	-
c432	233	7	1047.12	114.37	1045.23	30.43	0.18	3.76
c499	638	32	1150.61	870.61	1161.77	269.71	0.97	3.23
c880	443	26	1519.24	173.23	1516.46	36.90	0.18	4.69
c1355	629	32	1188.16	891.80	1169.25	273.20	1.59	3.26
c1908	425	25	1124.75	365.07	1148.27	109.25	2.09	3.34
c2670	841	157	3479.23	401.02	3463.73	120.23	0.45	3.34
c3540	901	22	2411.57	1070.61	2395.72	309.53	0.66	3.46
c5315	1806	123	9764.66	818.22	9983.29	403.17	2.23	2.03
c6288	2788	32	3860.03	15703.05	3769.48	4710.04	2.35	3.33
c7552	2114	126	6074.29	1406.70	6098.88	658.37	0.40	2.14
mul_4	158	8	883.38	98.82	890.33	34.85	0.79	2.84
mul_8	728	16	2127.35	710.21	2094.05	271.03	1.57	2.62
mul_16	3156	32	4775.07	9565.03	4845.29	5010.40	1.47	1.91
mul_24	7234	48	7636.46	39628.50	7478.02	29930.01	2.07	1.35
						Average	0.88	2.95

6.3 SSER Estimation on Benchmark Circuits

Using the proposed MC/QMC frameworks, we conduct SSER analysis on a variety of circuits including the ones in Figure 6.1, the ISCAS'85 benchmark circuits, and a series of multipliers. Table 6.2 first lists the name, the total number of nodes, and the total number of outputs for each circuits. The following four columns report the SSER values and the runtime required by the MC and QMC frameworks, respectively. The last two columns compute the SER difference and speedup, respectively, by comparing results from the MC and QMC frameworks.

From Table 6.2, 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 on not only the number of strike nodes, but also the number of convolutions between nodes. SER difference is computed by $|SSER_{MC} - SSER_{QMC}|/SSER_{MC}$ and the average of 0.88% difference implies that the QMC and MC frameworks are of the same quality. For all benchmark circuits, the overall speedup brought by QMC is 2.95X in average and the QMC runtime is comparable to that of [17].

Chapter 7

Conclusion

Traditional SER analysis techniques intend to mimic the static SPICE simulation. However, in presence of process variation, all static techniques tend to unavoidably underestimate true SERs and thus the research of statistical SER analysis is emerging. In this paper, we propose a method for building quality statistical cell models, based on which a Monte Carlo SSER framework is built. A heuristic is particularly proposed to apply quasirandom sequences to the framework for faster convergence and shorter runtime. According to the experimental results, the SSER errors are within 3% compared to Monte Carlo SPICE simulations, which are more accurate than those from previous works. Furthermore, the use of quasirandom sequences demonstrates an average of 2.95X runtime improvement over the baseline MC framework while preserving the same SSER quality.

SSER analysis is a thriving research topic in sub-90nm technologies. Future works include further reduction of dimensions to improvement runtime, applications of various variance reduction techniques to the QMC framework, and the inclusion of spatial correlation over cells.

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. Int'l Conf. Dependable Systems and Networks (DSN), pp. 389-398, 2002.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz,A. Keshavarzi and V. De, "Parameter variations and impact on circuits and microarchitecture," Proc. Int'l Conf. Design Automation, pp. 338-342, 2003.

[3] K. Bowman, S. Duvall and J. Meindl, " Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," IEEE Jour. Solid-State Circuits , vol. 37, pp. 183-190, 2002.

[4] O. A. Amusan, L. W. Massengill, B. L. Bhuva,S. DasGupta, A. F. Witulski and J. R. Ahlbin, "Design techniques to reduce set pulse widths in deep-submicron combinational logic," IEEE Tran. Nuclear Science, vol. 37, pp. 2060-2064, 2007.

[5] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," IEEE Tran. Nuclear Science, vol. 50, pp. 583-602, 2003.

[6] Y. Tosaka, H. Hanata, T. Itakura and S. Satoh, "Simulation technologies for cosmic ray neutron-induced soft errors: models and simulation systems," IEEE Tran. Nuclear Science, vol. 46, pp. 774-780, 1999.

[7] H. Cha and J. H. Patel, "A logic-level model for α particle hits in CMOS circuits," Proc. Int'l Conf. Circuit Design (ICCD), pp. 538-542, 1993.

[8] M. Omana, G. Papasso, D. Rossi and C. Metra, "A model for transient fault propagation in combinational logic," Proc. Int'l On-Line Testing Symp. (IOLTS), pp. 111-115, 2003.

[9] 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.

[10] M. Zhang and N. Shanbhag, "A soft error rate analysis (SERA) methodology," Proc. Int'l Conf. Computer Aided Design (ICCAD), pp. 111-118, 2004.

[11] B. Zhang, W.-S. Wang and M. Orshansky, "FASER: fast analysis of soft error susceptibility for cell-based designs," Proc. Int'l Smyp. Quality Electronic Design (ISQED), pp. 755-760, 2006.

[12] R. Rajaraman, J. S. Kim, N. Vijaykrishnan, Y. Xie and M. J. Irwin, "SEAT-LA: a soft error analysis tool for combinational logic," Proc. Int'l Conf. VLSI Design (VLSID), pp. 499-502, 2006.

[13] 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 Conf. (DATE), pp. 164-169, 2006.

[14] M. Zhang, T. M. Mak, J. Tschanz, K. S. Kim, N. Seifert and D. Lu, "Design for resilience to soft errors and variations," Proc. Int'l On-Line Test Symp. (IOLTS), pp. 23-28, 2007.

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

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

[17] Pumbaa H.-K. Peng, Charles H.-P. Wen and Jayanta Bhadra, "On soft error rate analysis beyond deep submicron - a statistical perspective," Proc. Int'l Conf. Computer Aided Design, (ICCAD'09), November 2009.

[18] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter variations and impact on circuits and microarchitecture," Proc. Design Automation Conf. (DAC), pp. 338-342, Jul. 2003.

[19] K. Ramakrishnan, R. Rajaraman, S. Suresh, N. Nijaykrishnan, Y. Xie and M. J. Irwin, "Variation impacts on SER of combinational circuits," Proc. Int'l Smyp. Quality Electronic Design (ISQED), pp. 755-760, 2006.

[20] N. Miskov-Zivanov, K.-C. Wu and D. Marculescu, "Process variability-aware transient fault modeling and analysis," Proc. Int'l Conf. Computer Aided Design (ICCAD), pp. 685-690, 2008.

[21] D. Franco, M. Vasconcelos, L. NAviner and J.-F. Naviner, "Signal probability for reliability evaluation of logic circuits," Elsevier Microelectronics Reliability, vol. 48, pp. 1586-1591, 2008.

[22] S. Ercolani, M. Favalli, M. Damiani, P. Olivo and B. Ricc6, "Estimate of signal probability in combinational logic networks ," Proc. European Test Conf. pp. 132-138, 1989

[23] W. J. Morokoff and R. E. Caflisch, "Quasi-random sequences and their discrepancies," SIAM Jour. Sci. Computing (SISC), vol. 15, pp. 1251-1279, 1994.

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

[25] Parameters of Low Power SoC Design, Semiconductor Roadmap Committee of Japan, http://strj-jeita.elisasp.net/pdf-nenjihoukoku-0303-roadmap/3-13_setsukei_task_force.pdf, 2003.

[26] S. Mitra, N. Seifert, M. Zhang, Q. Shi and K. S. Kim, "'Robust system design with built-in soft error resilience," IEEE Tran. Computer, vol. 38, pp. 43-52, 2005.

[27] W. Bartlett and L. Spainhower, "Commercial fault tolerance: a tale of two systems," IEEE Tran. Dependable and Secure Computing, vol. 1, pp. 87-96, 2004.

[28] R. Mastipuram and E. C. Wee, Soft errors' impact on system reliability, Cypress Semiconductor, http://www.edn.com/contents/images/454636.pdf, 2004.