University of Alberta

Stochastic Computational Approaches for the Reliability Evaluation of Nanoelectronic Circuits

by

Hao Chen

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Micro-Electro-Mechanical Systems (MEMS) and Nanosystems

Department of Electrical and Computer Engineering

Spring 2012
Edmonton, Alberta

To my family and friends

ABSTRACT

Reliability is fast becoming a major concern due to the nanometric scaling of CMOS technology. This thesis work initially presents novel computational models based on stochastic computation; in these models, probabilities are encoded in the statistics of random binary bit streams. A computational approach using the stochastic models is then proposed for the reliability evaluation of logic circuits. As it takes into account signal correlations and evaluates the joint reliability of multiple outputs, this approach accurately determines the reliability of a circuit; its precision is only limited by the random fluctuations inherent in the representation of the random binary bit streams. The proposed stochastic approach has a linear computational complexity and is therefore scalable for large circuit analysis. Extensive simulation results demonstrate the accuracy, scalability and execution simplicity of the proposed approach.

PREFACE

Following Moore's law, VLSI performance has increased by five orders of magnitude in the last three decades, realized by continuous technology scaling. As transistors become smaller, they switch faster, dissipate less power, and are cheaper to manufacture. However, scaling also exacerbates noise and reliability issues, thus posing new challenges. The future VLSI design methodology must account for those probabilistic behaviors in order to optimize performance, power and reliability. Several computational methodologies have been developed for evaluating the reliability of logic circuits. Accurate analytical approaches, however, have a computational complexity that generally increases exponentially with circuit size. This makes it intractable to analyze the reliability of large circuits.

In this thesis, we initially present novel computational models based on stochastic computation; in these models, probabilities are encoded in the statistics of random binary bit streams. A computational approach using the stochastic computational models (SCMs) is then proposed for the reliability evaluation of logic circuits. As it takes into account signal correlations and evaluates the joint reliability of multiple outputs, this approach accurately determines the reliability of a circuit; its precision is only limited by the random fluctuations inherent in the representation of the random binary bit streams. It is also able to account for various fault models as well as calculating the soft error rate (SER). Since it is based on both simulation and analysis, this approach takes advantages of the ease in implementation and accuracy in evaluation. The proposed stochastic approach has a low computational complexity and is therefore scalable for large circuit analysis.

Due to the deficiencies of existing reliability evaluation tools at the logic level, including the use of a constant error rate for gate failure, approximations in the assessment of gate topology and circuit analysis, etc., we further propose a more accurate and scalable approach that utilizes a transistor-level stochastic analysis for digital fault modeling. This approach accounts for very detailed measures, including the probability of failure of individual transistors, the topology of logic gates, diverse error models, timing sequences and the applied input vectors. Extensive simulation results and comparisons with existing approaches are presented; they demonstrate the accuracy, scalability and execution simplicity of the proposed approaches.

Parameter variations have been a major concern in circuit design due to their impact on the performance, power and robustness of CMOS circuits. To address this, analytical models are developed to quantitatively measure these effects by evaluating the functional variability, which is defined as the probability that a functional output of a circuit falls off the noise margin. Evaluation results show that while the impacts of variations are small and negligible for the current technology, it is increasingly becoming a factor that will impact a circuit's reliable operation as technology advances into 22nm and 16nm feature sizes. While delay errors caused by functional variability have been a focus of recent study, the effects of variations on the reliable functions of CMOS transistors, gates and circuits have not been adequately addressed. Based on the proposed stochastic models and approaches, several low-overhead techniques are developed for investigating the functional variability and how it impacts a circuit's reliable operation in advanced CMOS technologies.

ACKNOWLEDGEMENTS

I would first like to express my sincerest gratitude to my supervisor, Dr. Jie Han for his great support and guidance in my research work. His knowledge and experience in many areas in all aspects of research including conceptualization, the conduct of research, technical writing and presentation has inspired me and helped me develop my research. I would like to thank Dr. Fabrizio Lombardi for his insightful advices into this research. I would also like to thank Dr. Bruce F. Cockburn and Dr. Ming J. Zuo for serving on my dissertation committee.

I would like to thank Jinghang Liang, with whom I collaborated extensively. He always managed to spend considerable time discussing problems with me and provided a great deal of insights on my project. I would also like to thank many other people for their support and discussions: Zhixi Yang, Zhiyin Zhou, and Gautam Kamath.

I wish to thank my entire family, who supported me and taught me to always pursue my goals.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

CAD	Computer Aided Design
CDF	Cumulative Distribution Function
CMOS	Complementary Metal Oxide Semiconductor
EDA	Electronic Design Automation
FIT	Failures in Time
HCI	Hot-Carrier Injection
I/O	Input/Output
IC	Integrated Circuit
LEF	Line-Edge Roughness
MCS	Monte Carlo Simulation
MOSFET	Metallic Oxide Semiconductor Field Effect Transistor
NBTI	Negative-Bias Temperature Instability
PDD	Probabilistic Decision Diagram
PDF	Probability Density Function
PGM	Probabilistic Gate Model
PI	Primary Input
PO	Primary Output

PSN	Power Supply Noise
PTM	Probabilistic Transfer Matrices
RDF	Random Dopant Fluctuations
SCM	Stochastic Computational Model
SER	Soft Error Rate
SEU	Single Event Upsets
SPICE	Simulation Program with Integrated Circuit Emphasis
STM	Stochastic Transistor Model
TDDB	Time-Dependent Dielectric Breakdown
TSA	Temporary Single Stuck-At
TUT	Transistor under Test
ViER	Variation-induced Error Rate
VLSI	Very-Large-Scale Integration

CHAPTER 1

Introduction

The nanometric scaling of CMOS technology has introduced substantial challenges in circuit design; the higher integration density and lower voltage/current thresholds have increased the likelihood of soft errors. Process variations have prominently emerged to impact the performance and degrade the reliability of electronic circuits [1]. Process variations are due to random dopant fluctuation or manufacturing imprecision in the CMOS fabrication process. These physical-level characteristics have subsequently resulted in probabilistic device and circuit behavior. Novel nanoelectronic devices (such as carbon nanotubes, silicon nanowires, graphene and molecular electronics) have non-deterministic characteristics due to the uncertainty inherent in their operational behavior, so emerging technologies have significant limitations for reliable operation. Reliability has, therefore, become a major concern and probabilistic design methodologies are needed for assembling reliable circuits and systems out of unreliable devices [2, 3].

In the remainder of this chapter, we describe technology trends, soft errors and process variations that lead to uncertainty in circuit behavior in Section 1.1. Previous work on circuit reliability evaluation and soft error analysis is provided in Section 1.2. The main contribution of this research is stated in Section 1.3 and Section 1.4 outlines the remaining chapters.

1.1. Background and Motivation

1.1.1. Technology Trends

The advent of CMOS VLSIs has been accomplished by the downscaling of almost every device parameter such as feature size, power supply voltage, threshold voltage, etc. The need for more performance and integration drives the scaling down to the nanometer regime. Before these parameters approach their fundamental or physical limits, downscaling is still the most important and effective way for achieving high performance digital CMOS VLSI circuits operating with low power. The limit is expected to be reached when the feature size approaching 5nm since the off-state leakage current becomes too huge that will render the circuits ineffective in the sense that transistor will behave indefinitely. Until then we will still have probably 6 more generations taking about 20~30 years [4]. As CMOS technology enters the nanometer regime, shrinking device dimensions, lower design tolerances and fabrication variability have negative impacts on reliability and result in increased device failure rates [1, 5, 6]. The effects of process variations, due to random dopant fluctuations or sub-wavelength lithography, are expected to reduce transistor reliability as technology further scales [7]. Permanent faults can be caused by time-dependent dielectric breakdown of materials, hot carrier injection effects and negative bias temperature instability in transistors [8]. Electromigration becomes a major concern for interconnect reliability and can lead to faults due to connection shorts and opens [9]. Furthermore, transient (soft) errors may result from temporary environmental

influences [10]. Higher integration densities and lower voltage/ current thresholds have increased soft error rates in VLSI circuits.

Non-conventional nanotechnologies, currently being investigated as potential alternatives to CMOS, are expected to have lower reliability than current CMOS technology. This is a result of manufacturing processes and sensitivity to environmental factors – nondeterministic behaviors will be present due to quantum effects, environmental noise and, in some cases, inexpensive but inaccurate chemical self-assembly [4]. The imprecision and randomness inherent to the stochastic nature of chemical self-assembly will inevitably raise the density of defects in molecular devices, which subsequently cause malfunctions of logic gates and interconnects in circuits. The reliability limitations of nanoscale devices have become first-order issues and probabilistic designs, rather than deterministic ones, will be necessary to account for the stochastic behavior of nanoscale circuits and systems [11].

The design of ‘‘probabilistic logics” has been of interest since the early days of electronic computers when von Neumann proposed to synthesize reliable systems from unreliable components [12]. In his study, errors are treated probabilistically and a system is considered reliable if the probability of its correct output is greater than a threshold. As von Neumann stated, when the probability of output error reaches this threshold, the results from computation become irrelevant to the inputs and restoration of the outputs to their correct signal values is not possible. von Neumann’s work has motivated many efforts to characterize the reliability of fault-tolerant architectures in both conventional [13] and nanotechnology [14] systems. In light of the continuous scaling of CMOS and the emergence of new nanoscale technologies, reliability has increasingly been a concern and is expected to become a major design metric as performance and power are for today. This increasing demand on reliability design calls for accurate and efficient evaluation tools for the analysis of circuit reliability. Reliability evaluation through analysis and/or simulation also serves as the first step towards

understanding when fault-tolerance needs to be added to a system and what the resulting reliability gains are.

1.1.2. Process Variations

Process variability becomes the dominant factor impacting the design of high yield integrated circuits in nanometric CMOS technologies. The impact of process variations have been examined at different levels of abstraction: from device, circuit to micro-architectures. The most important sources of process variations include random dopant fluctuations [15] and line-edge roughness [16]. Random dopant fluctuations (RDFs) result from the discreteness of dopant atoms in the channel of a transistor. The dopant atoms control the switching threshold voltage of the transistor V_{th}. RDFs become more evident since the dopant concentration decreases exponentially as technology advances, which subsequently leads to great V_{th} variations. The impact of RDF-induced variations leads to a Gaussian distribution of V_{th} and the standard deviation is given by [15]

$$\sigma_{V_{th,RDF}} = \left(\frac{\sqrt[4]{4q^3 \varepsilon_{si} \varphi_B}}{2} \cdot \frac{T_{ox}}{\varepsilon_{ox}} \cdot \frac{\sqrt[4]{N}}{\sqrt{W_{eff} L_{eff}}} \right) \tag{1.1}$$

where W_{eff} and L_{eff} are the effective channel width and length, T_{ox} is the gate oxide thickness, N is the channel dopant concentration. $\varphi_B = 2k_B T \cdot \ln(N/n_i)$ (with k_B Boltzmann's constant, T the absolute temperature, n_i the intrinsic carrier concentration, q the elementary charge), and ε_{si} and ε_{ox} are the permittivity of the silicon and oxide, respectively. In a 16-nm technology, there are only tens of dopants left in the channel, therefore the RDF effect becomes dominant [1].

Line-edge roughness (LER) stems from the process of sub-wavelength lithography, which causes variations in the critical dimensions of the feature size. As technology scales, more severe roughness will result from the increased gap between the wavelength of light and the patterning width. Experiments have shown that LER is on the order of 5 nm, and it does not scale with the feature size

of devices [17]. LER is therefore expected to be a dominant source of variations, especially for short-channel devices. LER impacts both V_{th} degradation and sub-threshold leakage. Research shows that the V_{th} variation due to LER closely follows a $1/\sqrt{W_{eff}}$ relationship [16] [17]. For model simplicity, σ can also be modeled by:

$$\sigma_{V_{th,LER}} = \frac{\alpha}{\sqrt{W_{eff}}} \tag{1.2}$$

where W_{eff} and is the effective channel width, and α is a fitting parameter calibrated by experimental data [16] [18].

Since it has been shown that different sources of RDF and LER are statistically independent [17], the overall standard deviation for V_{th}, due to the effect of process variations, can be calculated as [18]

$$\sigma_{Vth} = \sqrt{\sigma^2_{Vth,RDF} + \sigma^2_{Vth,LER}} \tag{1.3}$$

In order to estimate process variation effects, SPICE Monte Carlo simulation is usually used to model random mismatch between different components due to process variation. As in equation (1.3), these parameters would ideally have a Gaussian distribution. The effect of V_{th} variation cannot be modeled easily in SPICE because of the very complex relation of threshold voltage for short-channel MOS models. Hense, usually the variation of parameter V_{th0} (long-channel threshold voltage) is considered. In a SPICE simulator the value of V_{th0} given by the manufacturer typically cannot be modified. A DC voltage source may be located in series with the gate terminal of the device which has a Gaussian distribution with zero mean value and standard deviation σ_{Vth} to model the voltage shift, as shown in Figure 1.1. To simulate variation effects, each transistor is replaced by an equivalent degraded transistor in circuit netlists.

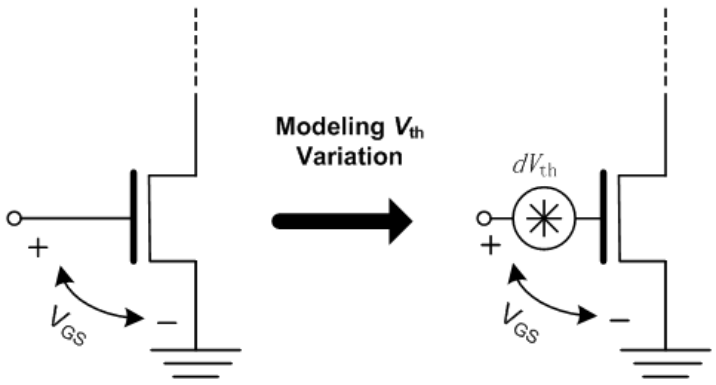


Figure 1.1. (a) MOS transistor model; (b) equivalent degraded transistor model in HSPICE.

1.1.3. Voltage Variations

Power supply noise (PSN) is caused by the non-ideal properties and fluctuations in the power supply network due to the parasitic resistance, capacitance and inductance of the interconnect. In [64], a stochastic approach is proposed to obtain the collective IR and LdI/dt drops and to analyze the power supply integrity. A stochastic method that computes the impulse response at every node is developed to propagate the statistical parameters through the linear model of the power grid to obtain the mean and standard deviation of the voltage drops. It has been observed that the overall voltage drop at any node in the power grid is approximately a Gaussian distribution.

The effect of power supply noise can be modeled by a coupling voltage source [64], as shown in Figure 1.2. An acceptable power noise for today's VLSI circuits is about $\pm 5\%$ [65]. However, as technology scales, PSN becomes a significant source of V_{dd} variations.

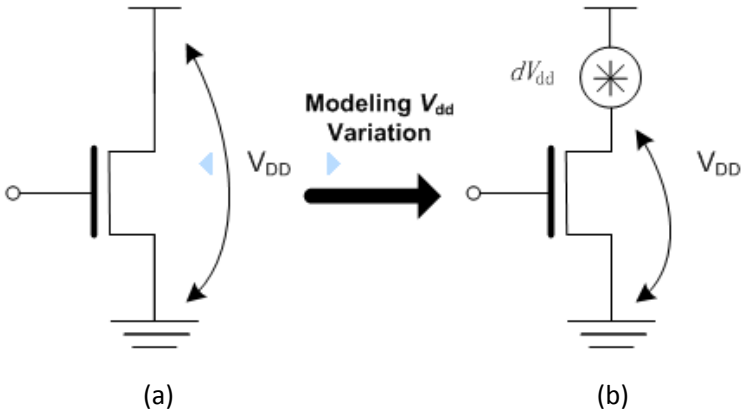


Figure 1.2. (a) MOS transistor model; (b) power supply noise coupled transistor model in HSPICE.

1.1.4. Temperature Variations

Temperature variations result in hot spots and cause leakage currents, which have great impact on the performance and power of a chip [1]. They also have effects on the degradation of devices (by affecting the NBTI of PMOS transistors, for example), and the leakage currents can make the transistor work in the subthreshold region. However, variations in temperature have relatively minor effect on Vth variability.

1.1.5. Input Variations

Due to the asymmetric characteristics of CMOS logic gates, a CMOS gate may experience different delay and power dissipation with respect to different input vectors [6]. Furthermore, it may also suffer a different probability of failure [28], which varies with different input voltages. Ideally, in a digital system, logical 1 is represented by V_{dd} and logical 0 is represented by ground voltage (V_{ss}). CMOS circuits usually restore output voltages to either V_{dd} or V_{ss}. However, noisy digital circuits may propagate degraded logical values caused by supply or ground noise.

The degraded signals have a great impact on a circuit's performance, leakage and reliability.

1.1.6. Soft Errors

This scaling will also result in increased soft error rates (SERs), due to a variety of factors such as an increased number of transistors on a chip, scaled supply voltages and reductions in feature sizes that reduce the node capacitance and thus lower the critical charge (Q_{cri}) required for reliable operation [10]. Soft errors, also called Single Event Upsets (SEU), are intermittent malfunctions of the hardware caused from energetic particles, namely neutrons from cosmic rays and alpha particles from packaging material. Particle strikes are the major reason for soft errors. When a particle strikes a sensitive region on a chip, the charge that accumulates could exceed the minimum charge that is needed to flip the value stored in the capacitance, resulting in a soft error. Based on device-physics models, the electrical effect of a radiation-induced transient is usually modeled by a double exponential current or voltage glitch. The voltage pulse generated by a particle strike on the diffusion region of a semiconductor device is determined by [19]

$$C_L \frac{dV_a(t)}{dt} + I_{DS}^{V_a} = i_{SEU}(t) \tag{1.4}$$

where C_L is the load capacitance, $I_{DS}^{V_a}$ is the drain current of the NMOS transistor and $V_a(t)$ is the transient output voltage [19]. For the node a in Fig. 1.2(a), the calculated voltage glitch $V_a(t)$ is shown in Fig. 1.2(b). Fig. 1.2(b) indicates that a radiation-induced transient could cause a momentary bit-flip at logic and system levels.

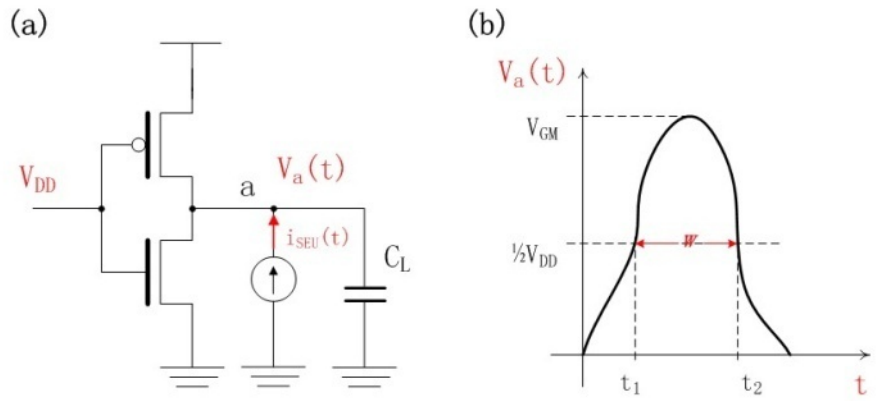


Figure 1.3. Soft error models: (a) a current model; (b) a radiation-induced voltage glitch.

1.2. Related Work

In response to this increasing demand on reliable design, several analytical approaches have been proposed for the reliability evaluation [20-28] and soft error rate (SER) analysis of logic circuits [10, 29-38]. In contrast to the general definition of reliability, i.e., the probability of the correct functioning of a circuit, SER has been used as a measure at the vulnerability of a circuit under the influence of soft errors. While reliability evaluation techniques are essential at the core of an SER analysis, an SER analyzer often considers various technology-dependent factors such as the electrical and timing effects of soft errors. Table 1 summarizes the features of some recent approaches for both reliability and SER analysis.

Table 1.1. Summary of different features of recent approaches in the technical literature for reliability evaluation and SER analysis.

Type		Accurate Analysis	Runtime	Memory Requirement	Scalability	Approach
Reliability evaluation techniques	**Analytical/ Symbolic**	Yes	Short	High	Yes, with reduced accuracy	[20], [22], [26], [28], [36]
		No	Short	Low	Good	[21], [23-25]
	Simulation-based	Yes	Long	High	Medium	Monte Carlo
	Simulation-based analytical	Yes, with limited precision	Medium	Medium	Good	[27] and this work
SER analyzers	**Analytical/ Symbolic**	Technology dependent	Short	High	Yes, with reduced accuracy	[29-33]
		Technology independent	Short	High	Yes, with reduced accuracy	[34-36]
	Simulation-based	Technology dependent	Long	Medium	Medium	[38]
	Simulation-based analytical	Technology independent	Medium	Medium	Good	[37] and this work

An analytical evaluation can be readily accomplished for small circuits with no loss of accuracy. As a circuit becomes large, it becomes difficult, if not impossible, to implement an exact analysis of its reliability; usually, a compromise is made on the accuracy of the evaluation. This is caused by the increased computational complexity in signal correlation due to reconvergent fanouts in combinational circuits and/or feedback loops in sequential circuits. Therefore, simulation has emerged as a possible solution. In a simulation-based approach, experimental data are gathered to characterize the behavior of a circuit by randomly sampling its activity. As an example, Monte Carlo (MC) simulation has been widely used when an analytical approach is not available or easy to implement. A disadvantage of simulation using random vectors is that numerous pseudo-random numbers need to be generated and a large number of simulation runs must be executed to reach a stable output, so evaluating large circuits a very time-consuming process.

For evaluating circuit reliability, several analytical approaches have been proposed, including those using probabilistic transfer matrices (PTMs) [20, 36], probabilistic gate models (PGMs) [26], Bayesian networks [21], probabilistic decision diagrams (PDDs) [22], Boolean difference calculus [23], circuit transformations [24] and several scalable methods based on single-pass analysis [25]. In these approaches, a probabilistic error model is considered; the probability that a circuit produces correct outputs is then obtained as the circuit reliability by a probabilistic analysis. For example, the PTM framework describes the error behavior of a gate (or a circuit) by a probabilistic truth table that contains the circuit information. Circuit reliability is then calculated based on matrix manipulation and arithmetic operations. An accurate analysis generally incurs a complexity exponential in circuit size; it is therefore practically infeasible for large circuit analysis. A design automation tool that considers reliability at the transistor level has recently been proposed for estimating the reliability of CMOS logic gates, as well as that of some small circuits such as full adders [28]. However, it has not been applied to the analysis of large circuits since it still incurs an excessive complexity in computation. So, a tradeoff between accuracy and complexity is usually sought by introducing either constraints on error behavior or simplified conditions on signal correlation. These approaches are referred to as reliability evaluation techniques in Table 1.

Recent research has also focused on analyzing the soft error rate (SER). SER has been used as a metric for measuring the likelihood of a circuit's malfunction when it is affected by soft errors. Several tools have been developed for SER analysis of combinational circuits, including SERA [29], FASER [30], SERD [31], and MARS-C [32], as well as its extension to sequential circuits MARS-S [33]. These tools estimate the SER of a circuit by considering three masking mechanisms: 1) logic masking, 2) electrical masking, and 3) latching-window masking [10]. While providing a rather detailed SER analysis, a technology-dependent method is rather complex; moreover, timing and electrical information are usually not

available at an early design phase. In Table 1 these SER analytical approaches are referred to as SER analyzers.

For logic masking, a statistical analysis has recently been developed as a framework for modeling the effect of soft errors [34, 35]. PTMs have been utilized to analyze the SER of sequential circuits [36]. A signature-based analysis provides an efficient and accurate estimation of SER in logic circuits [37]. This signature-based approach uses random input vectors to generate the signatures, or partial truth tables, through the parallel bit-wise simulation of a circuit; the signatures for all nodes in the circuit are then utilized for the calculation of the SER at the primary outputs. A Quasi-Monte Carlo method has been proposed in [38] for a statistical SER analysis; this method considers the effect of process variations and uses deterministic (so quasi-random) sequences to achieve a faster convergence and a shorter runtime than conventional Monte Carlo methods.

1.3. Contribution of this Work

The design of nanometric integrated circuits requires tools that accurately and efficiently compute reliability; this is a stringent requirement in mission critical applications. For space systems, the SER is often concerned. Hence, there is an urgent need to develop a unifying technical framework by which reliable design can be assessed with respect to different metrics (such as reliability and SER), while still retaining flexibility (as technology independence). A computational framework should also be applicable to a variety of fault models that can be encountered when designing such systems. Hence, permanent faults (such as stuck-at) and errors (such as of a transient/soft nature) should both be handled. The proposed models and approaches in this thesis work meet these objectives. In this work, a *simulation-based analytical approach* is presented for an accurate and efficient evaluation of the reliability of a circuit. While random binary bit streams are used to encode signal probabilities, this approach originates from the mathematical formulations of stochastic computation [39, 40] and probabilistic

gate models (PGMs) [26]. Stochastic computational models (SCMs) are proposed and constructed to implement the probabilistic analysis performed by PGMs, thus enabling an accurate analysis of circuit reliability. Differently from a traditional application of stochastic computation [41-44], this approach employs and leverages the bit-wise dependencies encoded in the random binary streams to efficiently handle signal correlations caused by recovergent fanouts or feedback loops in logic circuits. Hence, this approach avoids the large complexity typically encountered in a traditional analytical approach.

In contrast to methods based only on the simulation of random vectors, SCMs explicitly carry out the computation of signal probabilities, so they are generic and versatile for use in both algorithmic development and applications. This feature is shown by modeling various fault types and evaluating the joint signal probability. This work partially extends our previous work [27] but it expands its contribution to include novel materials such as the modeling of multiple fault types and joint output reliability. The joint output reliability is obtained as the joint probability that all outputs are correct and therefore, it accounts for signal correlations in the outputs. It is shown that the differences between the joint and individual output reliabilities are quite significant, especially for large circuits. The proposed approach is further applied to an SER analysis by taking into consideration the occurrence of multiple errors as well as their correlation.

As the SCM approach focuses on algorithmic development aiming for both high accuracy and low computational complexity, it relies on a logic-level analysis so that a fast evaluation can be provided at the early stage of a logic design process. However, it exhibits a major shortcoming: a constant probability of gate failure is assumed in a gate-level analysis, which actually has no physical basis for its applicability (as faults and defects usually affect individual devices such as transistors [45] [46]). For example, process variations, due to random dopant fluctuation or manufacturing imperfections in the CMOS fabrication process, have emerged to impact performance and degrade the reliability of electronic

circuits. The physical characteristics of devices have subsequently resulted in probabilistic circuit behaviors that manifest as a switching error of a transistor [45]. Manufacturing defects can also result in stuck-at faults in transistors [46]. The error probability also depends on the topology of a logic gate as well as its input vector. For example, if both the pull-up and pull-down networks are OFF in a CMOS gate, then the gate output is dependent on the previous output state (assuming no leakage).

As comprehensive circuit analysis (dealing with electrical and timing information such as in Monte Carlo SPICE simulation) is thought to increase the computational complexity, thus further complicating the reliability assessment problem, a transistor-level analysis could circumvent these disadvantages and therefore provide the basis for a more accurate analysis. In this work, an elaborated transistor-level SCM analysis is proposed to leverage accuracy and efficiency of reliability evaluation. Stochastic models are initially developed for transistors by extending the probabilistic analysis of gate-level SCMs. Logic gates are then modeled by considering sequential as well as combinational effects, such as timing sequences, gate topology and inputs to transistor operations. Since the probability is encoded into stochastic binary streams and signal correlation is carried on the bit-wise dependencies of the streams, the proposed transistor-level approach is also scalable for use in the analysis of large circuits.

The proposed models and approaches have various applications. As an example, in this thesis work, the transistor-level approach is used for variability and variability-induced error analysis. At device level, Monte Carlo SPICE simulations are utilized to characterize transistor (device) faulty behavior. And at the circuit level the transistor-level stochastic approach is adapted for error propagation and system-level analysis. Excellent model scalability enables efficient mapping between physical process variability and variability at circuit level. Simulation results show that our methodology enable accurate circuit

variability analysis compared with purely Monte Carlo SPICE, while achieve several orders of speed-up.

1.4. Thesis Outline

In this dissertation, we focus on stochastic transistor-level, gate-level and circuit-level reliability analysis. The thesis proceeds as follows. Chapter II presents stochastic computational models (SCMs) and their use in reliability evaluation. Chapter III discusses the transistor-level stochastic models and the approach for circuit reliability evaluation. Chapter IV demonstrates a direct application of the proposed models and approaches on variability analysis. Chapter V concludes the thesis.

CHAPTER 2

A Stochastic Computational Approach

Reliability is fast becoming a major concern due to the nanometric scaling of CMOS technology. Several computational methodologies have been developed for evaluating the reliability of logic circuits. Accurate analytical approaches, however, have a computational complexity that generally increases exponentially with circuit size. This makes it intractable to analyze the reliability of large circuits.

In this chapter initially presents novel computational models based on stochastic computation; in these models, probabilities are encoded in the statistics of random binary bit streams. A computational approach using the stochastic computational models (SCMs) is then proposed for the reliability evaluation of logic circuits. As it takes into account signal correlations and evaluates the joint reliability of multiple outputs, this approach accurately determines the reliability of a circuit; its precision is only limited by the random fluctuations inherent in the representation of the random binary bit streams. It is able to account for various fault models as well as calculating the soft error rate (SER). Since it is based on both simulation and analysis, it takes advantages of both ease in implementation and accuracy in evaluation. The proposed stochastic approach has a linear

computational complexity and is therefore scalable for large circuit analysis. Extensive simulation results and comparisons with existing approaches are presented; they demonstrate the accuracy, scalability and execution simplicity of the proposed approach.

This chapter is organized as follows. Section 2.1 reviews probabilistic gate models (PGMs). Section 2.2 presents stochastic computational models (SCMs) and their use in reliability evaluation. Section 2.3 discusses the stochastic approach for circuit reliability evaluation. Its accuracy and efficiency are assessed in Section 2.4. Extensive simulation results are provided in Section 2.5 together with a detailed comparison with existing approaches. Section 2.6 summarizes this chapter.

2.1. Probabilistic Gate Models

Most faults in nanometric logic circuits either are inherently probabilistic, or can be modeled probabilistically. Therefore, the reliability analysis of logic circuits has been based on the probabilistic treatment of signals [2]. The *signal probability* of an input or output of a logic gate is usually defined as the probability that the signal is logical "1." A *logic function* transforms its inputs to its output probability. The *reliability of an output* is defined as the probability of the output with an expected logic value of "1," or its complement otherwise. Given independent inputs, Boolean functions can be mapped to arithmetic operations of signal probabilities, by the following rules [2, 3]:

Rule I: Boolean "NOT," or $B = \overline{A}$, corresponds to

$$b = 1 - a, \tag{2.1}$$

where $b = P(B = 1)$ and $a = P(A = 1)$.

Rule II: Boolean "AND," or $C = AB$, corresponds to

$$c = a \cdot b, \tag{2.2}$$

where $c = P(C = 1)$, $b = P(B = 1)$ and $a = P(A = 1)$.

Rule III: Boolean "OR," or $C = A + B$, corresponds to

$$c = a + b - a \cdot b, \tag{2.3}$$

where $c = P(C = 1)$, $b = P(B = 1)$ and $a = P(A = 1)$.

However, if the input signals are not mutually independent, then the corresponding probability function may change. For example, the following rule maps the Boolean "AND" with two input signals that are totally dependent.

Rule IV: Boolean "AND" of a signal A with itself, or $C = AA$, corresponds to

$$c = a, \tag{2.4}$$

where $c = P(C = 1)$, and $a = P(A = 1)$.

Proof:

$$c = P(C = 1) = P(A = 1 \& A = 1) = P(A = 1) = a. \qquad \square$$

By applying "AND," "OR" and "NOT," any Boolean logic function can be mapped to an arithmetic equation of signal probabilities.

Example 1: The Boolean "XOR," or $C = A\overline{B} + \overline{A}B$ corresponds to

$$c = a \cdot (1 - b) + (1 - a) \cdot b, \tag{2.5}$$

where $c = P(C = 1)$, $b = P(B = 1)$ and $a = P(A = 1)$, provided that A and B are independent.

Further, a generic combinational network can be mapped to an arithmetic equation of signal probabilities, provided that correlations among signals are taken into consideration.

A probabilistic gate model (PGM) relates the output probability of a gate to its input and error probabilities; this is accomplished according to the function and malfunction (such as in the presence of an error) of the gate [26]. In general, the output probability of a gate can be calculated by the following equation,

Z = P(output "1"|gate faulty)• P(gate faulty) + P(output "1"|gate not faulty)• P(gate not faulty)

(2.6)

Consider a von Neumann fault, i.e., a fault that flips the correct output of a gate and resembles the behavior of a soft error. Let ε denote the error rate, i.e., $\varepsilon = \mathrm{P}(\text{gate faulty})$, and p, the fault-free output probability, i.e., $\mathrm{p} = \mathrm{P}(\text{output "1"}|\text{gate not faulty})$. The following equation is then applicable to any logic gate/function for the calculation of its output probability,

$$\mathrm{Z_v} = (1 - \mathrm{p}) \cdot \varepsilon + \mathrm{p} \cdot (1 - \varepsilon). \tag{2.7}$$

For example, consider a two-input AND gate (where $\mathrm{X_1}$ and $\mathrm{X_2}$ represent the input signal probabilities). Then, the output signal probability is given by $\mathrm{X_1\,X_2}$ for a fault-free gate. Given a probabilistic von Neumann fault, the output probability of an AND gate is then given by $\mathrm{Z_v} = (1 - \mathrm{X_1X_2})\varepsilon + \mathrm{X_1X_2}(1 - \varepsilon)$.

Stuck-at faults can also be modeled in a PGM. For a stuck-at-1 fault, (2.7) becomes

$$\mathrm{Z_{SA1}} = \varepsilon + \mathrm{p} \cdot (1 - \varepsilon). \tag{2.8}$$

For a stuck-at-0 fault, this is given by

$$Z_{SA0} = p \cdot (1 - \varepsilon). \tag{2.9}$$

A simple algorithm can be obtained by the iterative execution of a gate PGM according to the specific structure of a circuit. The execution of PGMs from the primary inputs to the outputs of a circuit produces the signal probability of each output. In this simple algorithm, it is assumed that all signals are mutually independent, so the correlation due to reconvergent fanouts is not considered. Hence, the basic PGM algorithm, albeit simple to implement, performs only an approximate evaluation.

An accurate algorithm accounts for signal dependencies in a circuit [26]. With no feedback, if all inputs are mutually independent, reconvergent fanouts are the only topological structures that introduce signal dependencies in a circuit. Fig. 2.1 (a) shows a simple reconvergent fanout. The fanout originates at point B and reconverges at point D. If the input to a fanout has a deterministic value (with probability 1 or 0), the statistical dependence of the two fanout branches is effectively eliminated. As per definition of statistical independence, P(B1=1, B2=1) = P(B1=1) P(B2=1) if and only if P(B=1) is equal to 1 or 0. As shown in Fig. 2.1 (b), the fanout is decomposed into two equivalent circuits with deterministic inputs “1” and “0”, containing no fanouts. Hence, signal dependencies are eliminated by fanout decomposition. The simple PGM algorithm can then be used to calculate the output probabilities of the two circuits. The found output probabilities are then utilized to evaluate the signal probability at point D, as

$$Z = Z_1 P + Z_0 (1 - P), \tag{2.10}$$

where Z_1 and Z_0 are the output probabilities when the fanout input is set to "1" and "0" and P is the signal probability at point B.

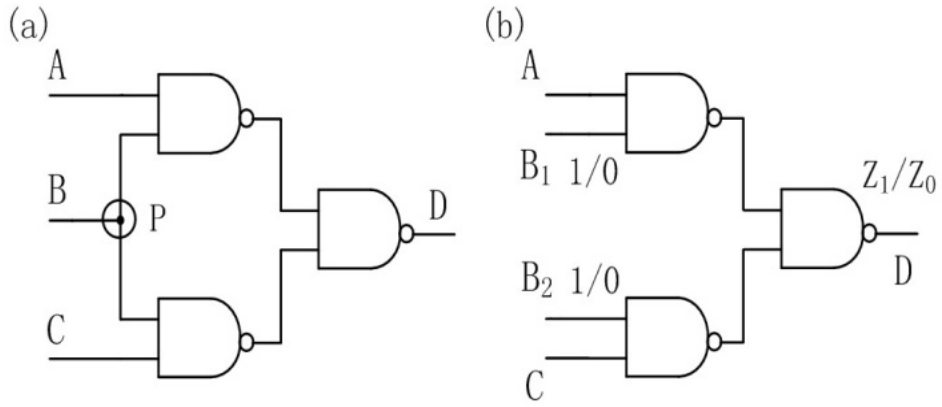


Figure 2.1. (a) A reconvergent fanout; (b) A fanout decomposition; its output probability is given by (2.10).

For a given circuit, signals are traced back from each primary output and all reconvergent fanouts on the paths leading to that output, are identified. For each reconvergent fanout, the circuit is decomposed into two sub-circuits. This process is repeated for every reconvergent fanout until all reconvergent fanouts are eliminated in the sub-circuits. The gate PGMs are then applied to obtain the output probabilities of each sub-circuit and the input probability of each reconvergent fanout. Finally, these are used to find the reliability of the original circuit.

An algorithm using PGMs allows for an accurate reliability evaluation by identifying reconvergent fanouts and then decomposing a circuit into sub-circuits for each reconvergent fanout. As the required computation doubles for each reconvergent fanout, the PGM algorithm has a computational complexity that increases exponentially with the number of dependent reconvergent fanouts [26]. As applicable to any analytical approach, the accurate analysis of large circuits is therefore likely to be intractable due to its very large computational overhead.

2.2. Stochastic Computational Models

Stochastic computation was first introduced in the 1960s for logic circuit design [39, 40], but its origin can be traced back to von Neumann's seminal work on probabilistic logic [47]. In stochastic computation, real numbers are represented by random binary bit streams that are usually implemented in series and in time. Information is carried on the statistics of the binary streams. Von Neumann's gate multiplexing is a special type of stochastic computational structure, in which redundant binary signals are implemented in parallel and in space. Both forms of stochastic computation have been the focus of investigation for fault-tolerant design of computational architectures [41-44, 51]. Stochastic computation offers advantages such as computational simplicity, fault tolerance and high speed [41, 52]. Its promise in data processing has been shown in several applications including stochastic decoding [53], neural computation [41] and fault-tolerant computing [43, 44].

In stochastic computation, signal probabilities are encoded into binary bit streams, i.e., serially in the time domain. Uniformly distributed random bit streams are used in this chapter to encode signal probabilities. A specific probability is represented by a number of bits set to a value that is usually in proportion to the mean number of 1's in a bit stream. Fig. 2.2 shows a stochastic encoding and an inverter. As Boolean operations can be mapped to arithmetic operations, the inverter probabilistically implements the complement operation of *Rule I*. Note that in Fig. 2.2, a sequence length of 10 bits is used for illustration purposes; a larger sequence length is usually needed in practice.

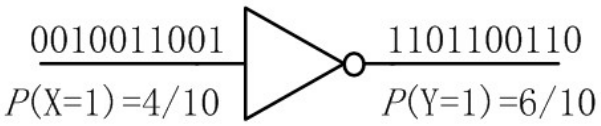


Figure 2.2. An inverter and a stochastic encoding.

Stochastic computation transforms Boolean logic operations into probabilistic computations in the real domain. Although each binary bit is processed by a Boolean gate, signal operations are no longer Boolean in nature, but they are arithmetic computations by stochastic logic. Complex arithmetic operations can be implemented by simple stochastic logic. According to *Rule II*, for instance, multiplication can be implemented by an AND gate, as shown in Fig. 2.3(b). Conventionally, the random distributions of bits in the binary streams are required to be statistically independent for correct computation, as shown in the example of Fig. 2.3(b) for multiplication. However, the bit-wise dependencies of random binary streams can be used to yield new stochastic logic models that account for the statistical correlation in input signals. This is shown in Fig. 2.3(a) as a general stochastic model of AND in which the two input signals may be correlated.

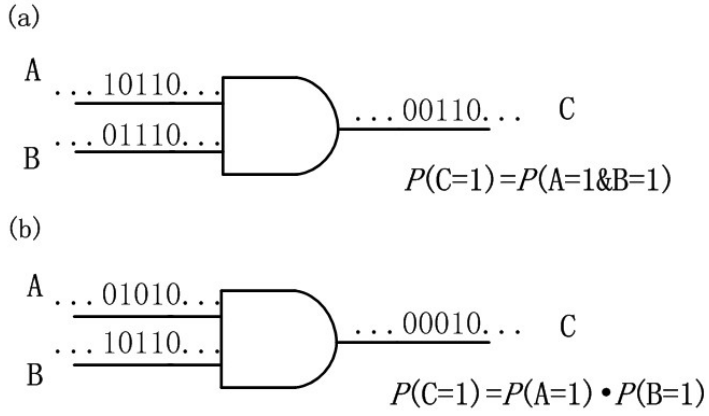


Figure 2.3. Stochastic AND logic: (a) the general model; (b) a special case of multiplication, when the two inputs are statistically independent.

Signal correlations are accounted for in the stochastic logic as follows. If an AND gate has two independent random bit streams X_1 and X_2 as inputs, then its output

will be a sequence encoding $Z = X_1X_2 = 0.81$ for $X_1 = X_2 = 0.9$. If the inputs are not independent, the output will depend on the correlation of the two input signals. If the two inputs are totally dependent, as shown in Fig. 2.4, then it results in $Z = X_1 = X_2 = 0.9$. This complies with *Rule IV*.

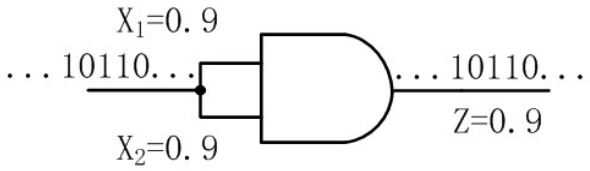


Figure 2.4. Signal correlations maintained in stochastic logic processing: totally correlated inputs X1 and X2.

A more general example can be given by a reconvergent fanout, as shown in Fig. 2.5. Inputs A, B and C are mutually independent, however, random bit streams X_1 and X_2 both originates at point B therefore they are correlated. When reconverging at output D, signal correlations are accounted for and can be calculated dictated by equation (2.10) as

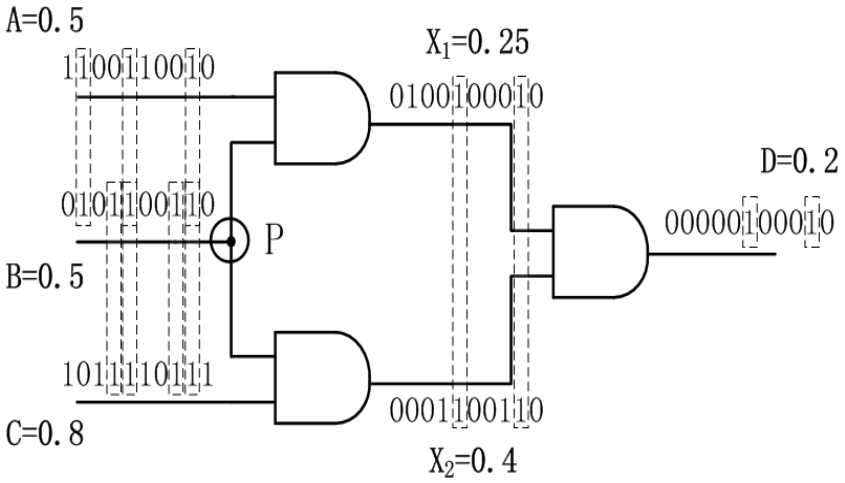


Figure 2.5. Signal correlations maintained in stochastic logic processing: correlated inputs X1 and X2.

$D = (0.5 \times 0.8) \times 0.5 + 0 \times (1 - 0.5) = 0.2$. If X_1 and X_2 are independent, $D = X_1X_2 = 0.1$.

This feature of stochastic computation is applicable to any logic function. Fig. 2.5 shows the stochastic operations performed by an XOR gate. The general model of Fig. 2.5(a) computes $\mathrm{P(C=1)=P(A=1\&}B=0|A=0\&B=1)$ while for independent inputs, a special case in Fig. 2.5(b) computes $\mathrm{P(C=1)=P(A=1)\cdot\big(1-P(B=1)\big)+(1-P(A=1))\cdot P(B=1)}$. Hence, a stochastic logic implements a corresponding probabilistic operation as dictated by a mapping rule or a combination of rules; at the same time, it maintains the signal correlations present in the random binary bit streams.

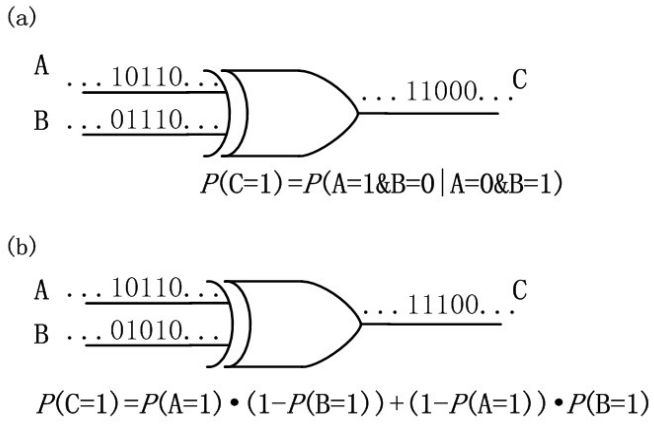


Figure 2.6. Stochastic XOR logic: (a) the general model; (b) a special case for statistically independent inputs.

This computational capability of stochastic logic allows the numerical evaluation of circuit reliability using stochastic computational models. Stochastic computational models (SCMs) are based on the operations of stochastic logic and the notions of PGMs. As discussed previously, any gate affected by a von Neumann fault can be modeled by (2.7). Moreover, (2.7) can be implemented by the stochastic logic of an XOR gate [27], as follows:

$$\mathrm{XOR_{sto}(p,\varepsilon)=p(1-\varepsilon)+(1-p)\varepsilon}, \tag{2.11}$$

where p is the fault-free output probability and ε is the gate error rate. The special case of a stochastic XOR is used to compute (2.7) because gate errors are assumed to occur independently. The general model of Fig. 2.6(a) must be used if

there is a correlation between the gate error and the input signals. (2.11) indicates that the PGM equation (2.7) can be implemented by a stochastic XOR logic regardless of the type of logic gate modeled by PGM. Therefore, an SCM can be obtained by adding an XOR gate to an unreliable gate and using an input of XOR to implement the gate error rate. This is shown in Fig. 2.6, in which an unreliable AND gate (Fig. 2.7(a)) is implemented by a general stochastic structure (Fig. 2.7(b)) and an SCM with an XOR gate (Fig. 2.7(c)). In this case,

$$\mathrm{p} = \mathrm{P}(\mathrm{X}_1 = 1 \& \mathrm{X}_2 = 1), \tag{2.12}$$

$$\mathrm{XOR}_{\mathrm{sto}}(\mathrm{p}, \varepsilon) = \mathrm{p}(1 - \varepsilon) + (1 - \mathrm{p})\varepsilon. \tag{2.13}$$

In addition to the von Neumann fault that was originally modeled in [27], the stuck-at faults can also be modeled by SCMs. For (2.8) considering a stuck-at-1 fault, an SCM can be constructed by adding an OR gate to the unreliable gate and using an input of the OR to implement the gate error rate, as

$$\mathrm{OR}_{\mathrm{sto}}(\mathrm{p}, \varepsilon) = \mathrm{p} + \varepsilon - \mathrm{p} \cdot \varepsilon = \varepsilon + \mathrm{p} \cdot (1 - \varepsilon). \tag{2.14}$$

For a stuck-at-0 fault, AND and NOT gates are used to implement the function of (2.9):

$$\mathrm{AND}_{\mathrm{sto}}(\mathrm{p}, \bar{\varepsilon}) = \mathrm{p} \cdot (1 - \varepsilon). \tag{2.15}$$

The SCMs for an unreliable AND gate affected by stuck-at-1 and stuck-at-0 faults are shown in Fig. 2.7 (d) and (e) respectively.

As indicated in (2.13) – (2.15), an SCM is universal, because it can be constructed for an arbitrary logic gate. The use of SCMs significantly reduces the computational complexity of a probabilistic analysis by using redundancy in the time domain and stochastic logic for processing the gate error rate.

A distinguishing feature of the SCM approach is that it efficiently handles reconvergent fanouts. As shown previously, signal correlations are maintained and preserved by the general model of stochastic logic. Since the statistical dependence of the fanout branches is eliminated if and only if the input to the fanout is 1 or 0, when signals are processed in the form of binary bit streams (consisting of 1's and 0's), logic operations do not need to consider the correlations caused by reconvergent fanouts. Signal dependencies are therefore inherently maintained in the distribution patterns of the random binary bit streams and are propagated to the next logic level.

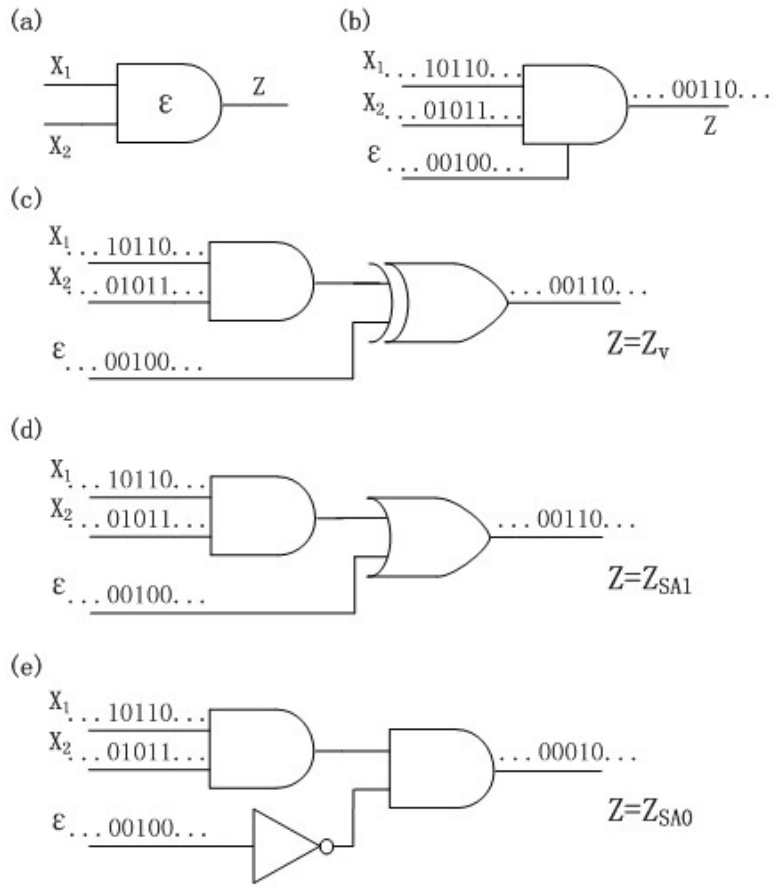


Figure 2.7. (a) An unreliable AND gate; (b) A stochastic logic implementation; (c) A stochastic computational model (SCM) for the von Neumann fault; (d) An SCM for the stuck-at-1 fault; (e) An SCM for the stuck-at-0 fault.

2.3. A Stochastic Approach for Circuit Reliability Evaluation

A stochastic computational network can be constructed using the SCMs of gates for the reliability evaluation of a circuit. The output probabilities are obtained by using stochastic sequences as inputs and propogating them from the primary inputs to the outputs.

A logic circuit may contain more than one primary output. Individual output reliabilities have been considered in [2.17]; in this chapter, we consider the joint output reliability. As output signal probabilities are encoded by the proportion of 1's in the output stochastic sequences, signal correlation is preserved in the distribution pattern. Let A, B, C, D be the output signals that may be correlated, then the output of a stochastic AND logic is given by

$$\mathrm{AND}_{\mathrm{sto}}(\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D}) = \mathrm{P}(\mathrm{A}=1,\mathrm{B}=1,\mathrm{C}=1,\mathrm{D}=1). \tag{2.16}$$

In fact, (2.16) evaluates the joint output probability of A, B, C and D, i.e., the probability that all outputs are "1." A joint probability of the outputs can thus be calculated by applying a stochastic AND, which takes into account the signal correlations among output signals.

As the output signal probability is the probability of the output being "1," the output reliability is the output signal probability if the fault-free output is expected to be 1 (or the complement of the output signal probability otherwise). Hence, a stochastic XOR gate with one inverted input is used to convert the output probability into reliability by either keeping or flipping the output sequence from each output of an unreliable circuit according to the correct output value (as produced by the equivalent fault-free circuit). This results in a stochastic computational architecture as shown in Fig. 2.8. The joint output reliability can

then be obtained by the output sequence of the AND gate that takes the outputs of the XOR gates as (correlated) input probabilities. The joint reliability provides a more accurate estimation, especially for large circuits.

We demonstrate the proposed SCM approach by taking the benchmark circuit C17 as an example. The von Neumann fault is used for illustration. Initially, a stochastic computational circuit is obtained by adding an XOR gate to each of the gates in C17, as shown in sub-circuit 1 in Fig. 2.9. At the same time, the original C17 is used to obtain the fault-free outputs, as shown in sub-circuit 2 in Fig. 2.9. Then the input signals as well as the gate error rate (that is now an input to the XOR gates) are initialized by generating random bit streams. The streams are propagated through the stochastic computational circuit and the original fault-free circuits. The output sequences are then processed by the XOR and AND gates to obtain the joint output reliability of the circuit.

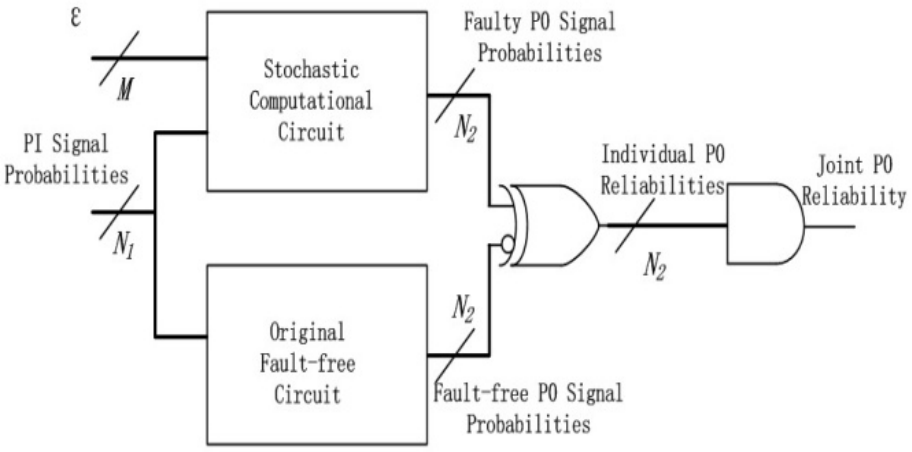


Figure 2.8. A stochastic computational architecture for the evaluation of the joint output reliability of a circuit.

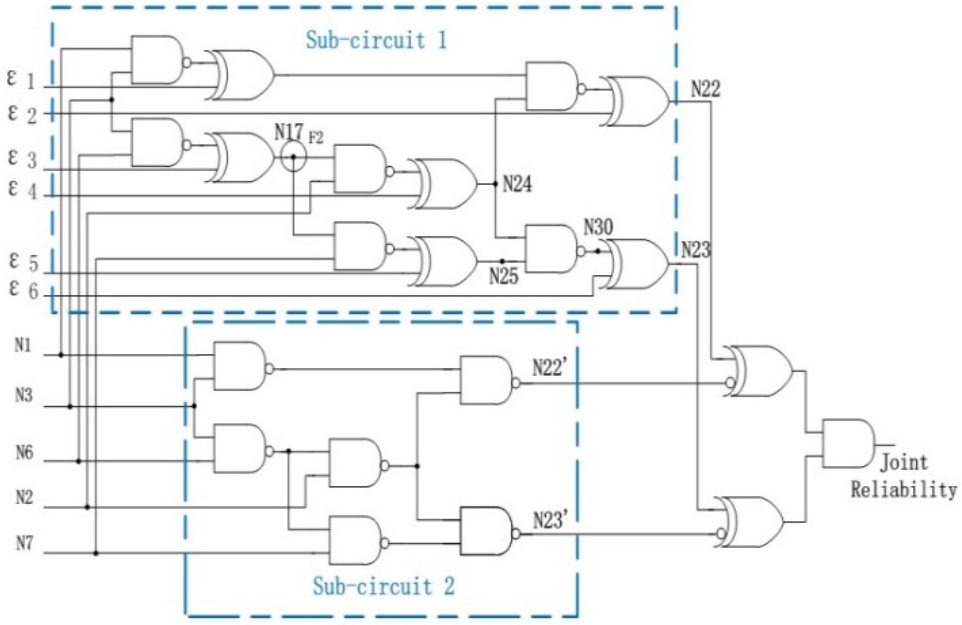


Figure 2.9. A stochastic architecture using SCMs for the evaluation of circuit reliability (for C17). Sub-circuit 1: the stochastic computational circuit; sub-circuit 2: the original fault-free circuit.

The evaluation procedure using the SCM approach as proposed in this manuscript, is as follows:

1. Construct the stochastic computational architecture by adding stochastic logic gate(s) according to a specific type of fault, as well as XOR and AND gates for obtaining the joint output reliability (Fig. 2.8);
2. Generate initial random bit streams encoding signal probabilities of the primary inputs and the gate error rates in the circuit;
3. Propagate the binary streams from the primary inputs to the outputs and obtain the stochastic bit streams at the outputs;
4. Decode the signal probability as the joint output reliability of the circuit from the obtained random bit streams.

In SCMs, signal probabilities are carried in the random binary bit streams and signal dependencies are preserved in the stochastic logic network. Hence, the reliability obtained using the SCM approach is accurate. The precision of the obtained result is only limited by features such as the resolution in the representation of bit streams and the random permutation and fluctuation of stochastic sequences. This occurs as in stochastic computation, probabilistic rather than deterministic values are propagated, which results in inevitable random fluctuations in the representation of probabilities, as detailed next.

2.4. Accuracy and Efficiency

Sequence length is an important parameter since it determines the resolution of the results. As an example for a sequence length of 10, the resolution is 0.1; this means that any probability with a precision less than 0.1 cannot be represented. An error due to a limited resolution is illustrated in Fig. 2.10. Fig. 2.10 (a) shows a scenario in which there are two independent inputs: X1=0.2 and X2=1. As Z= X1 X2, it can be found that Z=0.2 from the output binary stream. This is an accurate result. Fig. 2.10 (b) shows a scenario in which X1=0.8 and X2=0.8. The correct output should be 0.64. However, due to the limited resolution, it is found that Z=0.6 by this computation. In our experiments, a sequence of 1000 is mostly used to give a resolution of 0.001. A result is rounded to its nearest available representation, so the maximum error due to this resolution is 0.0005. This indicates that the result obtained in a single experiment will have a precision error of up to $1/2L$ for a sequence length of L.

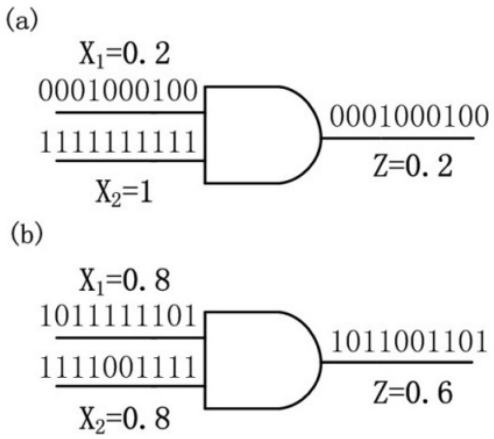


Figure 2.10. Resolutions in stochastic computation: (a) The desired output; (b) An imprecise output due to limited resolution.

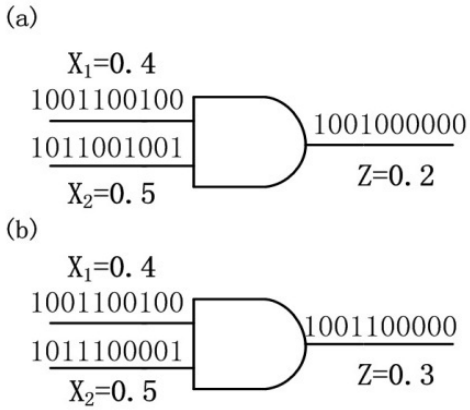


Figure 2.11. Random permutations in stochastic computation: (a) The desired permutation; (b) A permutation resulting in an error.

Errors can also be caused by the random permutation of bits in a sequence. Fig. 2.11 illustrates an example of a randomized permutation; the logic operation in Fig. 2.11 (a) gives the desired output value, while the operation in Fig. 2.11 (b) gives an output that is considered to be in error. In general, longer sequences tend to be better randomized; however, random permutations are probabilistic in nature and therefore, they do not always provide the desired results. The error due to a

random permutation is considered as “noise” and contributes to the notion that in a stochastic network, the output values are probabilistic rather than deterministic.

Random fluctuation is an inherent feature of stochastic computation [39, 40]. Simulation of C17 has shown that the result of each experiment fluctuates around the expected mean value, as shown in Fig. 2.12. The result of an experiment is an output sequence obtained for a given input combination. This fluctuation can be analyzed quantitatively by investigating the mean and variance of the output distribution. The law of large numbers states that the average result obtained from a large number of experiments is close to their mean value. Assuming that each experiment X_k is a random variable with the same mean μ and variance σ^2, then for n experiments

$$E\left[\frac{1}{n}\sum_{k=1}^{n} X_k\right] = \frac{1}{n}\sum_{k=1}^{n} E[X_k] = \frac{1}{n} n\mu = \mu, \tag{2.17}$$

$$\mathrm{Var}\left(\frac{1}{n}\sum_{k=1}^{n} X_k\right) = \frac{\sigma^2}{n}. \tag{2.18}$$

The error can be measured by the standard deviation, as

$$e = |x - \mu| \approx \frac{\sigma}{\sqrt{n}}, \tag{2.19}$$

where n is the number of experiments performed. (2.19) shows that the error is proportional to $1/\sqrt{n}$. Therefore, a higher accuracy can be obtained by increasing the number of experiments. However, as precision is limited by resolution, an increase in the number of experiments does not always result in a better accuracy. Fig. 2.13 shows the distribution of the results from 1,000,000 experiments for C17. According to the Central Limit Theorem, the distribution of a large number of samples approaches a Gaussian distribution (as the sample size increases). In this case, a Gaussian distribution (with mean 778.218 and variance 27.5601 calculated from experimental data) fits very well the distribution of the data. Hence, a Gaussian distribution results since the deterministic input signals are affected by noise and become probabilistic in the stochastic processing network.

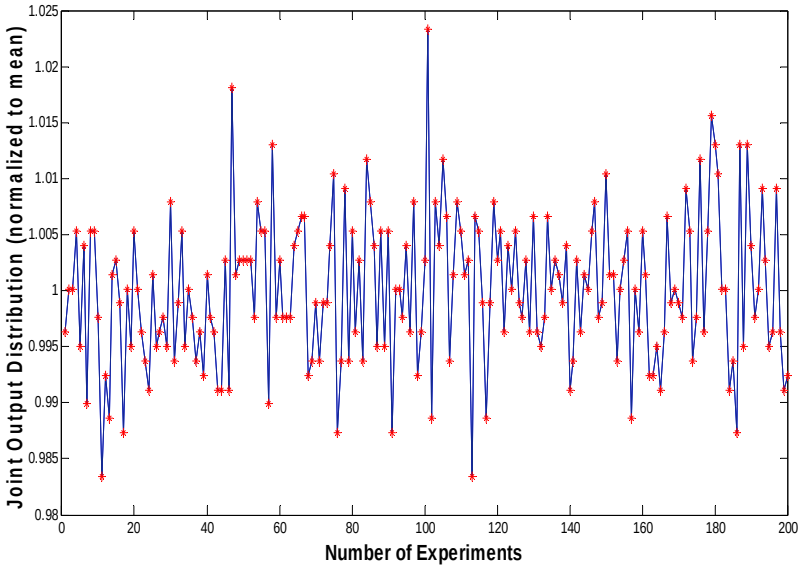


Figure 2.12. Random fluctuations in stochastic computation for C17.

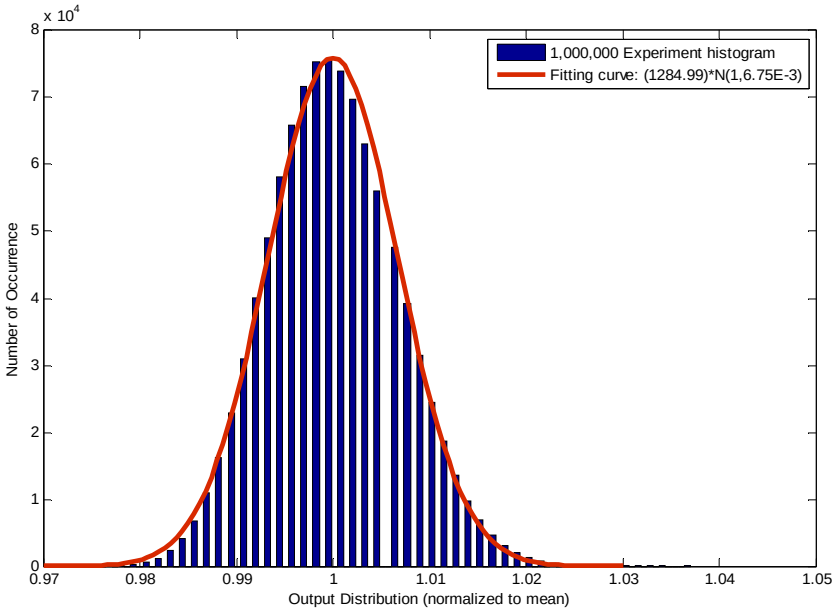


Figure 2.13. Output distributions are approximately Gaussian for C17.

There are two major steps in the stochastic computation of circuit reliability: (1) the generation of the input random bit streams (this usually accounts for over 90% of the total run time); (2) the propagation of the bit sequences through the circuit.

For random sequence generation, the SCM approach has a linear complexity with the number of 1's in the sequence to be generated. It is thus proportional to the sequence length for a fixed error rate and therefore, it has often a linear complexity with sequence length. For sequence propagation, the SCM approach has a complexity that increases linearly with the number of gates in a circuit and the length of the random bit sequences.

The presentation of this chapter has dealt only with combinational circuits; even though not explicitly presented, the proposed SCM approach can also be applied to the analysis of sequential circuits.

2.5. Validation by Simulations

In this section, the proposed SCM approach is compared to the following reliability evaluation techniques: the accurate PGM algorithm, the PTM approach, the signature-based SER analyzer and the Monte Carlo (MC) simulation. Simulations have been performed on a 2.66-GHz Pentium microprocessor with 2 GB of memory.

To validate the model, the proposed SCM approach is initially compared with the accurate PGM and PTM approaches on some small circuits. While individual output reliabilities are reported in [27], only joint output reliabilities are reported in this section. Table 2.1 shows the evaluated circuit reliability and the relative error for a von Neumann fault with ε=0.05. The relative error is defined as the ratio of the difference between an approximate value and the accurate value over the accurate value. A maximum of 1000 inputs are used in the simulations. As shown in Table 2.1, the SCM approach yields highly accurate results; the maximum relative error is around 0.2% by using a sequence length of 1000. Table 2.2 shows the results for stuck-at faults. It can be seen that the maximum relative error is around 0.1% for both stuck-at-1 and stuck-at-0 faults.

Accuracy can be further improved by increasing either the sequence length, or the number of experiments for each input combination. However, a significant increase in runtime is encountered at a rather marginal improvement in accuracy (as indicated by (2.19)). Although the runtime of the SCM approach appears mostly longer than that of the PGM and PTM approaches, the longer time required for identifying and decomposing reconvergent fanouts in PGM, as well as the time required for analyzing the circuit structure in PTM, is not included in the runtime reported in the tables. In the SCM approach, the runtime is dominated by the procedure for generating the random bit sequences. If fault-free deterministic inputs are used, then the only sequences to be randomized as initial inputs are the gate error rates. As these are independent processes, it would be possible to further reduce the runtime for the random bit generation by a parallelized procedure.

Table 2.1. Accuracy comparison of the SCM, PGM and PTM approaches for the von Neumann fault.

Circuit	*Characteristics*			*SCM* $\varepsilon = 0.05\ L = 1000$			*Accurate PGM* $\varepsilon = 0.05$		*PTM* $\varepsilon = 0.05$	
	Gates	*PIs*	*POs*	*R*	*Time (s)*	*Rel. error*	*R*	*Time (s)*	*R*	*Time (s)*
C17	***6***	***5***	***2***	0.7830	0.06	0.11%	0.7839	0.002	0.7839	0.001
Majority	***10***	***5***	***1***	0.8634	0.15	0.13%	0.8623	0.002	0.8623	0.05
Full adder (Majority)	***8***	***3***	***2***	0.7896	0.02	0.1%	0.7904	0.0007	0.7904	0.18
Full adder (XOR/NAND)	***6***	***3***	***2***	0.8016	0.01	0.2%	0.8000	0.001	0.8000	0.001
Full adder (NAND)	***12***	***3***	***2***	0.6547	0.03	0.21%	0.6533	0.008	0.6533	0.002
Comparator	***4***	***2***	***3***	0.8265	0.008	0.01%	0.8264	0.006	0.8264	0.0009
Decoder2	***6***	***2***	***4***	0.7385	0.01	0.09%	0.7392	0.03	0.7392	0.009
MUX4	***7***	***6***	***1***	0.8228	0.14	0.09%	0.8221	0.001	0.8221	0.52

The SCM approach is further compared with the signature-based approach [37] for the SER analysis of the LGSynth91 benchmarks [54]. For a gate affected by soft errors, the SER is defined as its output error probability while for a circuit, the SER is defined as the complement of its joint output reliability. The simulation results are shown in Table 2.3. The signature-based SER analyzer finds the testability of a circuit by using bit-parallel simulation for SER calculation; this process is similar to the stochastic simulation as used in the SCM approach. Only temporary single stuck-at (TSA) faults are analyzed in [27].

Table 2.2. Accuracy comparison of the SCM, PGM and PTM approaches for stuck-at faults.

Circuit	*Stuck-at-1 (TSA-1) Fault*							*Stuck-at-0 (TSA-0) Fault*						
	SCM $\varepsilon = 0.05$ $L = 1000$			*Accurate PGM* $\varepsilon = 0.05$		*PTM* $\varepsilon = 0.05$		*SCM* $\varepsilon = 0.05$ $L = 1,000$			*Accurate PGM* $\varepsilon = 0.05$		*PTM* $\varepsilon = 0.05$	
	R	***Time (s)***	***Relative error***	***R***	***Time (s)***	***R***	***Time (s)***	***R***	***Time (s)***	***Relative error***	***R***	***Time (s)***	***R***	***Time (s)***
C17	0.915	0.06	0.03%	0.914	0.003	0.914	0.001	0.854	0.068	0.01%	0.855	0.003	0.855	0.001
Majority	0.929	0.15	0.04%	0.929	0.003	0.929	0.047	0.917	0.165	0.04%	0.917	0.003	0.917	0.047
Full adder (Maj)	0.883	0.02	0.01%	0.883	0.0005	0.883	0.175	0.883	0.023	0.09%	0.883	0.0005	0.883	0.189
Full adder (XOR/NAND)	0.905	0.01	0.09%	0.904	0.002	0.904	0.001	0.883	0.015	0.06%	0.883	0.002	0.883	0.001
Full adder (NAND)	0.825	0.03	0.11%	0.824	0.008	0.824	0.002	0.773	0.033	0.12%	0.774	0.008	0.774	0.003
Comparator	0.891	0.00	0.07%	0.891	0.006	0.891	0.001	0.926	0.007	0%	0.926	0.007	0.926	0.001
Decoder2	0.815	0.01	0.08%	0.815	0.028	0.815	0.009	0.904	0.016	0.08%	0.903	0.029	0.903	0.009
MUX4	0.940	0.15	0.02%	0.940	0.001	0.940	0.523	0.868	0.151	0.01%	0.868	0.001	0.868	0.536

In most prevous studies, the SER is expressed in Failures in Time (or *FIT*, usually in the number of failures in 10^9 hours) [37], [29-31, 38, 55]. Since the effect of multiple errors is considered in the SCM approach, probabilities are instead used in this section as SER values. In Table 2.3, the stuck-at faults are assumed to be uniformly distributed with a gate SER $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-2}$. Although an SER

is generally considered small, a rather large value (10^{-2}) is used in the simulations to show the difference between the approaches. The overall circuit SER is given by the sum of the stuck-at-0 and stuck-at-1 SER values. Table 2.3 shows that the two approaches produce very close results for most circuits when $\varepsilon = 10^{-6}$; however, the relative differences of the SER increases at $\varepsilon = 10^{-2}$. A signature-based approach considers the sensitivity of each gate separately and sums over the resulting circuit SER for each gate. This is different for the SCM approach that accounts for multiple error occurrences as well as their correlation by considering the joint effects of multiple errors in the evaluation of a circuit.

Table 2.3. SER analysis using the SCM and signature-based approaches.

Circuit	*No. Gates*	*Signature-based SER analyzer* $Signature\ length = 10,000$		*SCM approach* $Sequence\ length = 1,000,000$, $input = 10,000$			
		SER $\varepsilon = 10^{-6}$	*SER* $\varepsilon = 10^{-2}$	*SER* $\varepsilon = 10^{-6}$	*Relative difference*	*SER* $\varepsilon = 10^{-2}$	*Relative difference*
majority	10	3.4503×10^{-6}	3.4473×10^{-2}	3.4374×10^{-6}	0.38%	3.3185×10^{-2}	3.9%
parity	15	1.5×10^{-5}	1.5×10^{-1}	1.4976×10^{-5}	0.16%	1.3977×10^{-1}	7.3%
decod	22	1.9884×10^{-5}	1.9873×10^{-1}	2.0012×10^{-5}	0.63%	1.8642×10^{-1}	6.6%
x2	38	1.8547×10^{-5}	1.8519×10^{-1}	1.8463×10^{-5}	0.45%	1.7563×10^{-1}	5.4%
pm1	41	1.8923×10^{-5}	1.8871×10^{-1}	1.8897×10^{-5}	0.14%	1.7965×10^{-1}	5.0%
cu	43	1.6577×10^{-5}	1.6578×10^{-1}	1.6631×10^{-5}	0.32%	1.5816×10^{-1}	4.8%
z4ml	45	2.6005×10^{-5}	2.6010×10^{-1}	2.6105×10^{-5}	0.38%	2.4208×10^{-1}	7.4%
mux	50	6.6941×10^{-6}	6.7689×10^{-2}	6.7369×10^{-6}	0.64%	6.4965×10^{-2}	4.0%
pcle	61	2.8963×10^{-5}	2.8896×10^{-1}	2.9061×10^{-5}	0.34%	2.6632×10^{-1}	8.5%

These two approaches model a similar error scenario as long as the node (or gate) SER is small. In this case, the probability that more than one error occur is even smaller and thus negligible. This is confirmed by the simulation results for $\varepsilon = 10^{-6}$. As the circuit size or the SER of each node increases, however, the probability of independent multiple-error occurrence increases. This may result in

large discrepancies using these two evaluation methods, as indicated in the reported simulation results for $\varepsilon = 10^{-2}$. In the scenario of multiple dependent transient faults caused by a single radiation upset [56], the SCM approach can readily be adapted to model the correlated multiple errors by using dependent stochastic sequences. Although the SCM approach needed a longer runtime due to its use of the redundant stochastic sequences, the signature-based approach required a greater effort in execution due to the complicated programming and algorithms involved. The SER analysis by both approaches primarily considers logic masking; however the results obtained can be enhanced by modeling technology-dependent factors such as the electrical and timing effects on SER.

Finally, large benchmarks of ISCAS-85 are simulated to compare the efficiency of the SCM approach with that of the Monte Carlo simulation (MCS). The most straightforward and intuitive algorithm for reliability analysis with certain constant gate error rate is based on fault injections and random pattern simulations in the Monte Carlo framework, circuit reliability can be established through the resulting statistical outcomes. The simulation-based approach MCS generates pseudo-random numbers, usually uniformly distributed, over the probability interval [0,1] for each gate and compares it to the constant gate error rate to determine whether the gate fails or not. The pseudo-random numbers for each gate need to be generated independently to generate a random pattern that mimics erroneous circuit's behavior. Then random pattern simulation with a random input vector is performed to evaluate the effect of a sample of erroneous gates on the final output. The final results are obtained as statistical outcomes using a large number of single-bit random pattern simulations. A major drawback of this purely simulation-based approach is that a very large number of runs with millions of pseudo-random generations are required to achieve convergence of the output results. For instance, suppose that the circuit under evaluation has *Nin* primary inputs, *Nout* primary outputs, and *NG* gates, and a total number of *M* single-bit runs are needed to achieve convergence. The total number of pseudo-number generations $KMCS = (NG + Nin) \times M$. In contrast, the proposed simulation-

based analytical approach, SCM, takes the advantage of statistical randomness but has been developed as a general computational framework to efficiently implement analytical algorithms. From an analytical perspective, signal and error probabilities are encoded into binary bit streams, and analytical algorithms are explicitly performed in terms of stochastic computations. Stochastic bit sequences can be generated using pseudo-random generators. Alternatively, they can be generated directly from available physical sources of randomness [44]. From a simulation perspective, computation overhead is greatly reduced though parallelization of single-bit simulations. For a certain number of repetitions M with required convergence constraint, the SCM approach process data in parallelized sequence thus reduce the number of repetitions. For the SCM approach, given a sequence length of L bits were used, only M/L repetitions are needed to be equivalent to the MCS with M runs. Moreover, pseudo-random numbers are only generated for an extremely small portion of erroneous bits per stochastic sequence therefore the SCM approach is more efficient in that it requires less pseudo-random number generations in the stochastic computing process. For example, suppose the sequence length is L, and the gate error rate is ε. The total number of pseudo-number generations $KSCM = NG \times \varepsilon \times M + Nin \times M/L$. Therefore SCM is orders of magnitude faster than the MCS. This is confirmed by the simulation results shown in Table 2.4. In the MC simulation, a total of one million simulations were run for each circuit in order to ensure a relatively stable output reliability. For the SCM approach, a sequence length of 1,000 bits were used for 1,000 random inputs, which is equivalent to the number of MCS runs.

Table 2.4. Simulation results of ISCAS-85 benchmarks by the SCM approach and Monte Carlo simulation.

Circuit	Characteristics			Monte Carlo Simulation $\varepsilon = 10^{-3}$ sample $= 1,000,000$			SCM $\varepsilon = 10^{-3}$ L $= 1,000$, input $= 1,000$		
	gates	inputs	outputs	Average Reliability	Joint Reliability	Runtime	Average Reliability	Joint Reliability	Runtime
C432	250	36	7	0.9841	0.9476	31.3m	0.9838	0.9494	10.77s
C499	202	41	32	0.9967	0.9132	24.7m	0.9968	0.9135	9.18s
C880	383	60	26	0.9911	0.8056	58.1m	0.9910	0.8049	17.35s
C1355	546	41	32	0.9924	0.7969	83.6m	0.9926	0.7984	28.01s
C1908	880	33	25	0.9786	0.6761	139.3m	0.9796	0.6801	39.68s
C2670	1193	157	64	0.9896	0.6464	227.6m	0.9895	0.6424	60.85s
C3540	1669	50	22	0.9459	0.5614	350.7m	0.9471	0.5625	81.72s
C5315	2307	178	123	0.9903	0.4623	508.6m	0.9902	0.4587	135.31s
C6288	2416	32	32	0.8934	0.1189	466.2m	0.8936	0.1211	110.63s
C7552	3512	207	108	0.9830	0.2556	778.6m	0.9830	0.2532	160.06s

It can be seen that both approaches produce accurate evaluations of circuit reliability, while the SCM approach requires a significantly smaller runtime compared to the MC simulation. Our further evaluation showed that the differences in the simulation results were mainly due to the different input samples used in these two approaches, as MCS uses 1,000,000 randomized input vectors while SCM samples 1,000 input vectors. The runtime of the SCM approach could be further reduced through re-using pseudo-random numbers with reduced randomness in the stochastic sequences. Also shown in Table 2.4 is that the obtained joint output reliabilities deviate significantly from the average reliabilities of individual outputs for circuits of such size, and thus represent a

more accurate measure of circuit reliability. These results demonstrate the accuracy and efficiency of the SCM approach, especially when it is applied to the evaluation of large circuits.

2.6. Summary

Advances of VLSI circuits and systems into the nanometric regimes require accurate and efficient reliability evaluation techniques. In this chapter, a novel stochastic approach is proposed as a computational framework for the reliability evaluation of logic circuits. This approach uses stochastic computational models (SCMs); it accurately evaluates the reliability of a circuit with a precision limited by the inherent randomness of the binary bit streams used in stochastic computation. Compared to accurate analytical approaches found in the technical literature, the proposed SCM approach efficiently handles signal correlations introduced by reconvergent fanouts and thus significantly reduces the computational complexity. Specifically, it has a complexity that increases linearly with the length of the random bit sequences and the number of gates in a circuit.

Compared to the simple simulation of random vectors, the proposed approach has the following distinguishing features: 1) Versatility. The SCM is flexible due to its pronounced arithmetic nature. 2) Generality. The SCM approach has been developed as a general computational framework to efficiently implement analytical algorithms. 3) Scalability. Compared to Monte Carlo simulation, the SCM approach is scalable as it benefits from the use of a reduced number of pseudo-random numbers. The proposed stochastic approach is therefore potentially useful in the design and test of reliable VLSI circuits and systems.

CHAPTER 3

A Transistor-Level Stochastic Reliability Analysis

Over the last few decades, most quantitative measures of VLSI performance have improved by many orders of magnitude; this has been achieved by the unabated scaling of the size of the MOSFET in CMOS technology. However, scaling also exacerbates noise and reliability issues, thus posing new challenges in circuit design. Reliability becomes a major concern due to many and often correlated factors, such as parameter variations and soft errors. Existing reliability evaluation tools focus on algorithmic development at the logic level [20-37]; a constant error rate for gate failure is usually employed, thus leading to approximations in the assessment of a VLSI circuit. This chapter proposes a more accurate and scalable approach that utilizes a transistor-level stochastic analysis for digital fault modeling. This approach accounts for very detailed measures, including the probability of failure of individual transistors, the topology of logic gates, disparate fault models, timing sequences and the applied input vectors. Stochastic transistor models (STMs) are initially developed and then used to construct transistor-level gate models. Three types of mappings are considered, i.e., from

the gate inputs to the operations of the transistors, from the transistors to a pull up/down network and from the pull up and pull down networks to the gate output. Finally, a circuit-level evaluation approach is utilized to assess a circuit's reliability using the proposed STMs. Since signal correlation is accounted for in the distribution pattern of the stochastic binary bit streams, this approach is scalable for use in the evaluation of large circuits. Simulation results are provided to demonstrate both the accuracy and the efficiency of the proposed approach.

This chapter is organized as follows [67]. Section 3.1 presents the stochastic transistor models (STMs) and the logic gate models. Section 3.2 outlines the circuit analysis approach and Section 3.3 reports the simulation results. Section 3.4 concludes this chapter.

3.1. Stochastic Models of Transistors and Logic Gates

The CMOS transistor is a voltage-controlled current source. In digital design, the transistor is usually considered to operate as a switch (Figure 3.1). As a switch, the transistor gets its source and drain conducted, if the gate voltage is "high" (for NMOS) or "low" (for PMOS). Thus, the ON/OFF state of a transistor is determined by the applied gate voltage. When transistors are used in a gate and the gate voltage falls off the noise margins, the transistor operates in an indefinite manner, so its state is referred to as "indefinite" or "IND." Hence, there are three operational states in the transistor model used in this thesis: ON, OFF and IND (Figure 3.1). These states are determined by three different gate inputs, i.e., voltage as high, low and outside of the noise margins (corresponding to g as logic "1," logic "0," and "X," respectively, in Figure 3). Mapping between the gate input and the operation of the NMOS and PMOS transistors is summarized in Table 3.1 and it is given as follows:

- For the NMOS transistor, the input 0 results in OFF; the input 1 results in ON; and the input X results in IND.

- For the PMOS transistor, the input 0 results in ON; the input 1 results in OFF; and the input X results in IND.

Table 3.1. Mapping between the gate input and the transistor operations.

Transistor type / Gate input	NMOS State (St)	PMOS State (St)
$g = 0$	OFF	ON
$g = 1$	ON	OFF
$g = X$	IND	IND

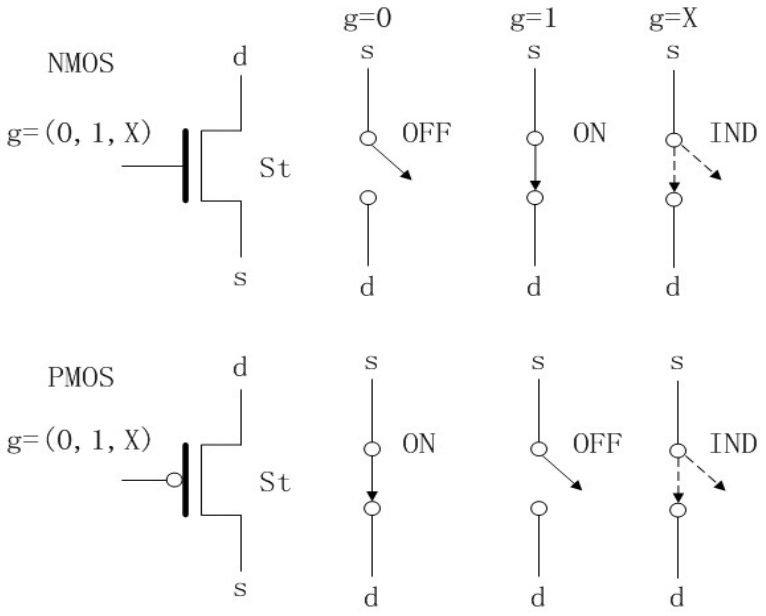


Figure 3.1. Transistor model as a probabilistic switch. g : gate terminal; d: drain terminal; s: source terminal; St: state (ON/OFF/IND) of the transistor.

Since a transistor may be affected by a transient error, its state could be erroneous. A transistor can therefore be modeled as a probabilistic switch such that the probability distribution of the state (ON, OFF and IND) of the transistor is determined by the input signal probability. Here, this probabilistic switching of the transistor is modeled using the stochastic computational models (SCMs) presented in the previous section. As shown in Figure 3.2, the gate input is represented by a random bit stream, and so is the switching error probability of the transistor. If the transistor is affected by a flipping error with an error rate $\varepsilon = \mathrm{P}(\text{transistor faulty})$, then the gate input can be considered to be changed by a stochastic XOR as

$$\mathrm{g}' = \mathrm{XOR}_{\mathrm{sto}}(\mathrm{g}, \varepsilon) = \mathrm{g} \cdot (1 - \varepsilon) + (1 - \mathrm{g}) \cdot \varepsilon \tag{3.1}$$

The newly-generated gate input is then used to determine the state of the transistor (considered now as reliable). This results in a stochastic model for an unreliable NMOS or PMOS transistor as shown in Figure 3.2. A similar stochastic model can be used to estimate the transistor's behavior when affected by a different type of error. For example, the stuck-ON/OFF fault can be modeled using the following equations:

$$\mathrm{g}'_{\mathrm{Stuck-ON}} = \mathrm{OR}_{\mathrm{sto}}(\mathrm{g}, \varepsilon) = \mathrm{g} + \varepsilon - \mathrm{g} \cdot \varepsilon = \varepsilon + \mathrm{g} \cdot (1 - \varepsilon) \tag{3.2}$$

$$\mathrm{g}'_{\mathrm{Stuck-OFF}} = \mathrm{AND}_{\mathrm{sto}}(\mathrm{g}, \bar{\varepsilon}) = \mathrm{g} \cdot (1 - \varepsilon) \tag{3.3}$$

The stuck-ON/OFF transistor fault models are shown in Figure 3.3 and Figure 3.4, respectively. So, a stochastic transistor model can be constructed as follows:

- If the transistor is affected by a flipping error, then the stochastic XOR is used;
- If the transistor is affected by a stuck-ON error, the stochastic OR is used;
- If the transistor is affected by a stuck-OFF error, the stochastic inverter

and AND is used.

In (3.1), (3.2) and (3.3), an input X is considered to always produce the same output (i.e., X), regardless of the stochastic logic being performed.

Differently from the logic-level SCM approach in [27], in which the random bits in the binary streams are considered equivalent and with no order, the stochastic sequences used in this thesis match the operation of the transistor in multiple clock cycles. This allows to account for errors that occur within a single and multiple clock cycles. Additionally, the temporal sequences in the binary bits ensure the correct modeling of the floating state that could result from the pull-up and pull-down operations of the transistors, as explained in more detail in the next section.

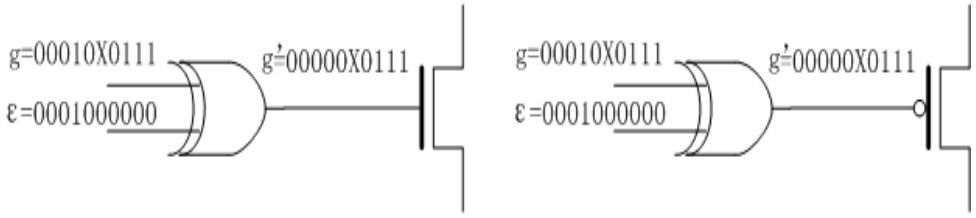


Figure 3.2. Stochastic transistor models for the flipping error: (a) NMOS and (b) PMOS.

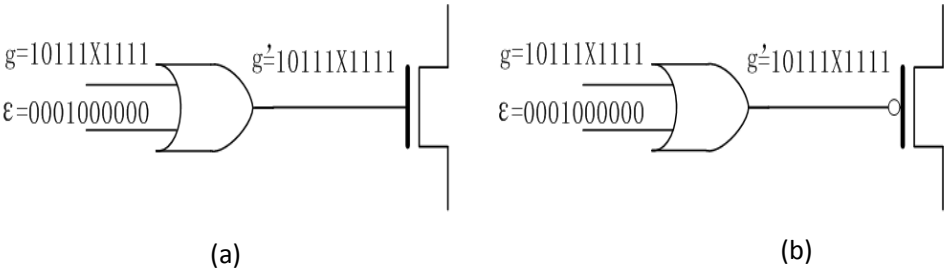


Figure 3.3. Stochastic transistor models for the stuck-ON error: (a) NMOS and (b) PMOS.

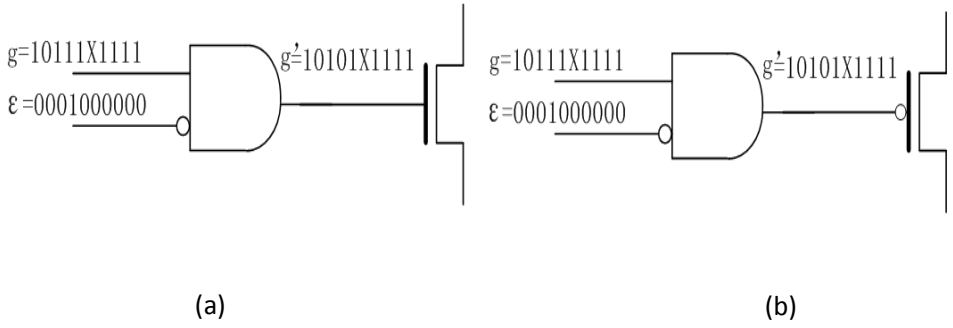


Figure 3.4 Stochastic transistor models for the stuck-OFF error: (a) NMOS and (b) PMOS.

Faults and defects are likely to affect the correct operation of individual transistors in the logic gates of a combinational circuit; so for an accurate and realistic reliability analysis, the gate error rate should be derived in terms of the transistor error probability, while considering also the gate topology as well as the input vectors.

Similar to the logic-level SCM approach in [27], discussed in Chapter II, the transistor-level stochastic approach uses stochastic random sequences to represent both signal and error probabilities. However, the traditional SCM approach is static in the sense that circuit reliability is evaluated without considering signal sequences and timing information, thus it may not always be directly applicable to the temporal operation of the transistors and of the sequential elements such as the flip-flops. Therefore the stochastic streams in the new model are defined differently to account for signal sequencing. Initially, consider the CMOS inverter as an example (Figure 3.5). Given an input sequence N_{in} and the flipping error rate sequences ε_p and ε_n (the stuck-ON/OFF can be considered similarly), the ON/OFF states of the PMOSFET P and the NMOSFET N are obtained via functional bit-parallel simulation of the input sequences. Then the output node sequence N_{out} is found as a function of the state of each transistor, i.e., $N_{out} = f(St_P, St_N)$; this can generally be estimated according to the functionality of the

gate. For the inverter it is given as follows: (1) when P (pull-up network) is ON and N (pull-down network) is OFF, the output is logic 1; (2) when P (pull-up network) is OFF and N (pull-down network) is ON, the output is logic 0; (3) when P and N are simultaneously OFF, the output is floating, or Z (i.e., it depends on its previous value); and (4) when P and N are simultaneously ON or any of P and N is indefinite or IND, the output is defined as unknown, or X. This is also shown in Table 3.2.

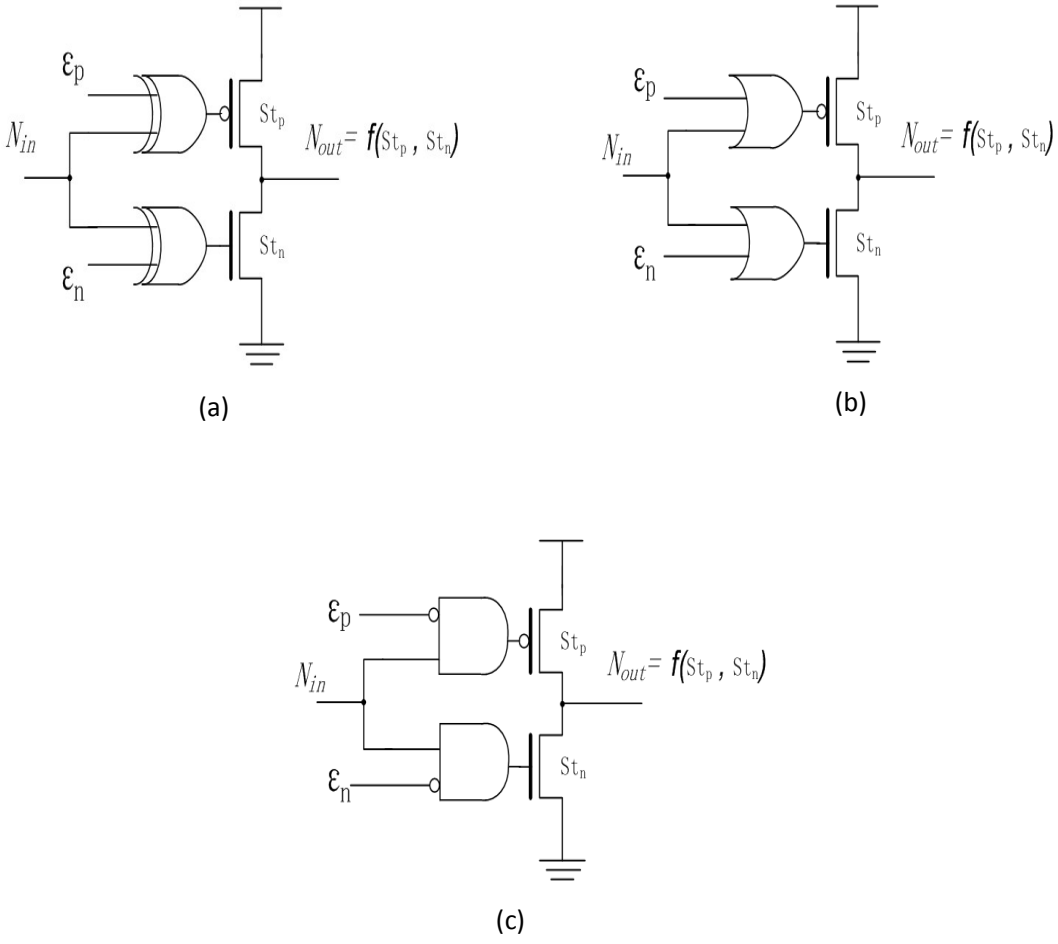


Figure 3.5. Proposed stochastic model for the inverter for (a) flipping, (b) stuck-ON, and (c) stuck-OFF errors

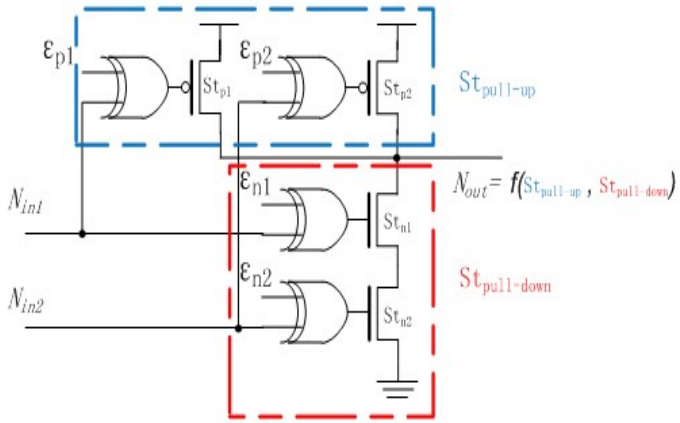
Table 3.2. The gate output as determined by the pull up and pull-down networks.

	Pull-up Network ***(P)***	**Pull-down Network** ***(N)***	**Gate Output** (N_{out})
Network state or gate output	OFF	ON	**0**
	ON	OFF	**1**
	OFF	OFF	**Z**
	ON	ON	**X**
	IND	Don't care	
	Don't care	IND	

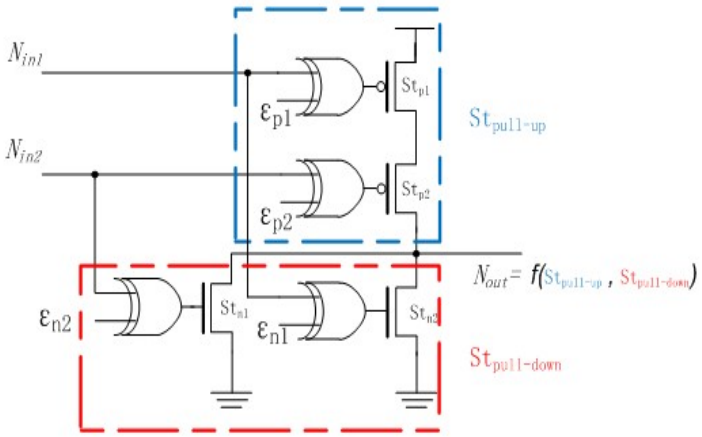
Since random binary bit streams are used for each circuit node by functional/fault simulation (i.e., serially in the time domain), the sequencing property of the bits are defined in such a way that each bit in the sequences represents a logic value at a certain node in one clock cycle. Therefore, it is possible to define and calculate the floating output Z as the previous bit value in a sequence. This is based on the assumptions that the node leakage is negligible and that the node charge will remain at the same level until a refresh operation occurs. Since the unknown (leakage) output X usually falls into the undefined voltage region, it is assumed that as the worst case, it is a faulty output (it usually cannot be immediately restored by the gates). The proposed method is amenable to a parallel-bit simulation for combinational circuits, while for storage elements (such as flip-flops), sequential simulation may still be required.

To further understand the proposed approach, more general cases can be illustrated using NAND2 and NOR2 as in these gates the pull-down and pull-up networks consist of multiple transistors. Let the input sequences N_{in1} and N_{in2} have a length of L and each bit represents the signal value during one clock cycle;

therefore, a sequence represents the sequential states in L clock cycles. The error rate of each transistor is then encoded into the stochastic sequences.

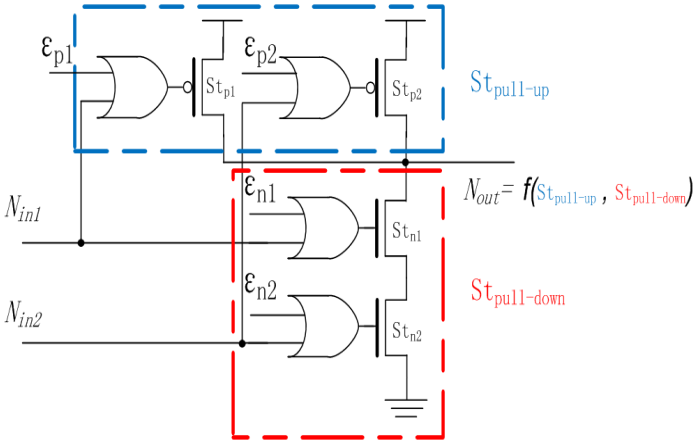


(a)

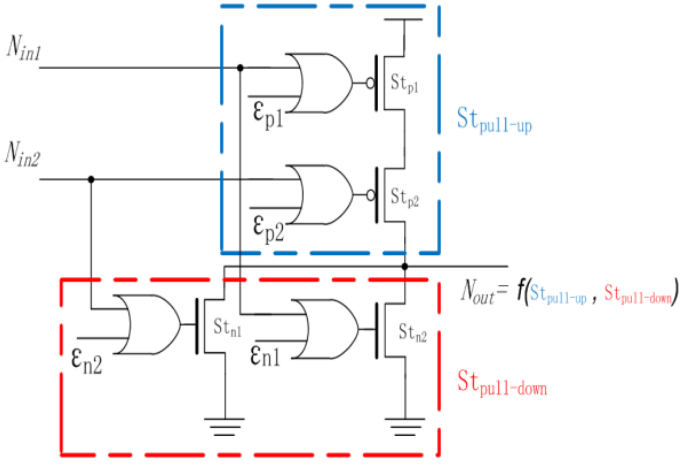


(b)

Figure 3.6. Transistor-level stochastic models for flipping errors for logic gates: (a) NAND2 and (b) NOR2.

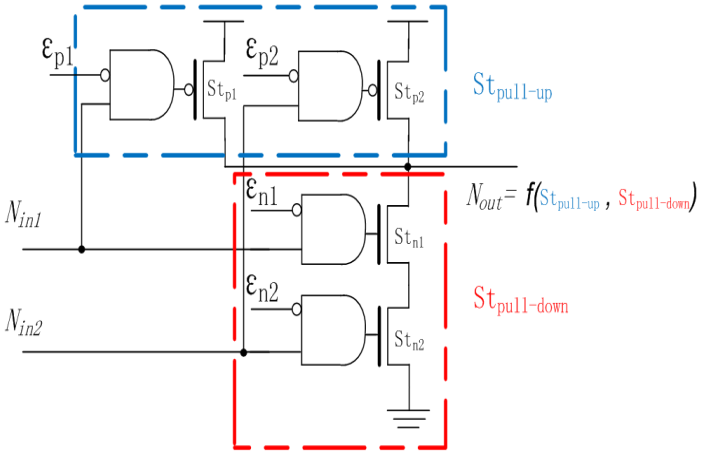


(a)

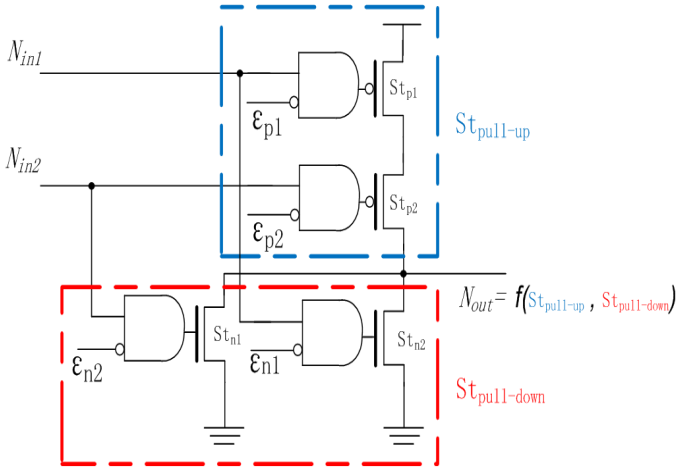


(b)

Figure 3.7. Transistor-level stochastic models for stuck-ON errors for logic gates: (a) NAND2 and (b) NOR2.



(a)



(b)

Figure 3.8. Transistor-level stochastic models for stuck-OFF errors for logic gates: (a) NAND2 and (b) NOR2.

As shown in Figure 3.6, Figure 3.7, Figure 3.8, the operation of each transistor is characterized by an error rate ε and a gate input for two sequences of length L.

Assuming that the transistors are independent, the newly-generated input sequences by the stochastic gates determine the ON/OFF state of each transistor. The output of the pull-up/down network can be computed based on the states of the individual transistors. For a pull-up/down network with multiple transistors, its operational status is determined by the topology of the pull-up/down network as follows:

Table 3.3. Mappings between transistors and networks: (a) series network; and (b) parallel network.

(a)

Transistor state / Network state	Transistor #1	Transistor #2
ON	ON	ON
OFF	OFF	Don't care
	Don't care	OFF
IND	IND	ON
	ON	IND
	IND	IND

(b)

Transistor state / Network state	Transistor #1	Transistor #2
ON	ON	Don't care
	Don't care	ON
OFF	OFF	OFF
IND	IND	OFF
	OFF	IND
	IND	IND

For the transistors connected in series in a pull-up/down network, the network is "ON" when all transistors are ON; this is equivalent to applying a stochastic AND gate to the states of the transistors. As shown in Figure 3.6(a) for example, $\mathrm{St}_{\mathrm{pull-down}} = \mathrm{AND}(\mathrm{St}_{\mathrm{n1}}, \mathrm{St}_{\mathrm{n2}})$. The detailed mapping relationships for two transistors connected in series are shown in Table 3.3(a), and they can readily be extended to any number of transistors connected in series.

For the transistors connected in parallel in a pull-up/down network, the network is "ON" when any of the transistors is ON; this is equivalent to applying a stochastic OR gate to the states of the transistors. As shown in Figure 3.6(a) for example, $\mathrm{St}_{\mathrm{pull-up}} = \mathrm{OR}(\mathrm{St}_{\mathrm{p1}}, \mathrm{St}_{\mathrm{p2}})$. The detailed mapping relationships for two

transistors connected in parallel are shown in Table 3.3(b), and they can similarly be extended to any number of transistors connected in parallel.

The output of the gate N_{out} is established by considering the states of the pull-up and pull-down networks as follows (also shown in Table 3.2):

- If the pull-up network is ON and the pull-down network is OFF, then the output is 1.
- If the pull-up network is OFF and the pull-down network is ON, then the output is 0.
- If the pull-up network is OFF and the pull-down network is OFF, then the output is Z, which depends on the previous value.
- If the pull-up network is ON and the pull-down network is ON, or any of the networks is IND, then the output is X.

The gate error rate/reliability can then be calculated by comparing the faulty and fault-free output sequences. Hence, the proposed approach to modeling a logic gate consists of three types of mapping: 1) mappings from the gate inputs to the operations of the transistors, as illustrated in Figure 3.1, Table 3.1; 2) mappings from the transistors to a pull up/down network, as shown in Table 3.3; and 3) mappings from the pull up and pull down networks to the gate outputs, as shown in Table 3.2.

3.2. A Circuit-Level Evaluation Approach

For a circuit made of unreliable transistors, its reliability can be estimated by evaluating the stochastic bit streams following propagation from the primary inputs to the primary outputs. Practically, this can be done by comparing the obtained output sequences for the unreliable and reliable circuit case; such a procedure can be implemented for the benchmark circuit C17 as follows. Initially, the stochastic (unreliable) circuit is obtained by adding a stochastic gate to each of

the transistors in C17: using the stochastic XOR for the flipping error, adding the stochastic OR for a stuck-ON error or adding the stochastic inverter and AND for a stuck-OFF error. Then, the input signals as well as the transistor error probability (that is now an input to the stochastic gate) are initialized by generating random bit streams. The streams are propagated through the stochastic circuit and the original fault-free circuit, as shown in Figure 3.9. Subsequently, XOR gates are used to detect the mismatch of the stochastic sequences from the unreliable and the reliable circuits. Since the C17 has more than one primary output, the joint circuit error probability can be obtained by using a stochastic OR gate to detect any error present in the multiple stochastic output sequences. The final structure is shown in Figure 3.9.

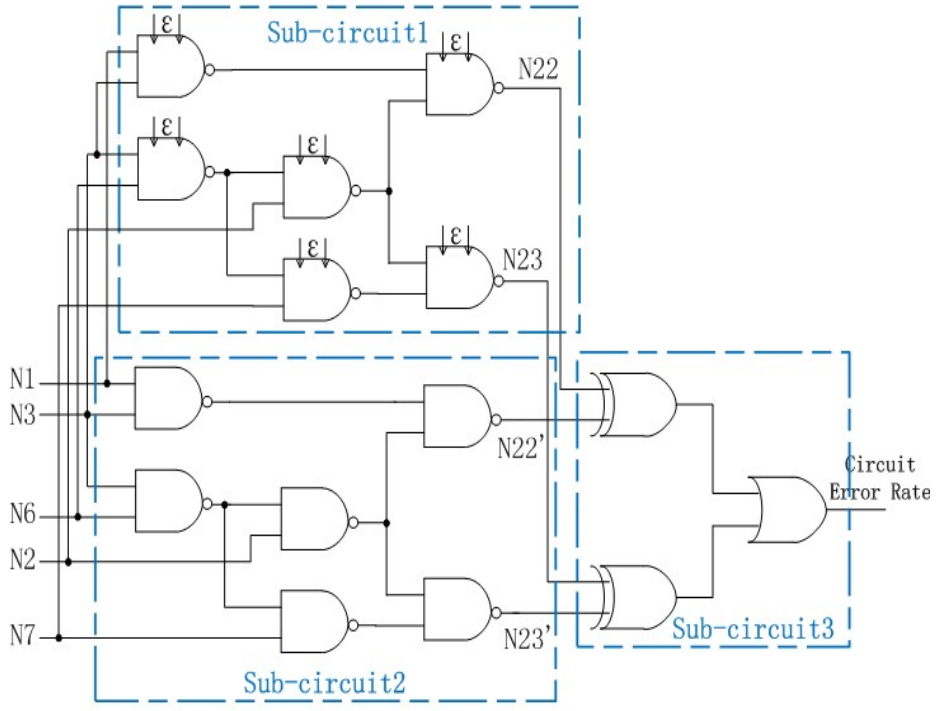


Figure 3.9. Computational structure for the reliability evaluation of C17. Sub-circuit1: the stochastic circuit, implemented using the NAND gate of Figure 9(a); Sub-circuit2: the original fault-free circuit, implemented using regular NAND gates; Sub-circuit3: XOR and OR gates for obtaining the joint error probability from the output stochastic sequences.

The evaluation procedure using the transistor-level stochastic approach as proposed in this thesis is given as follows:

1. Construct the stochastic circuit by adding a stochastic gate to each of the transistors in the circuit (for flipping or stuck-ON/OFF errors);
2. Generate the initial random bit streams for the signal probabilities of the primary inputs and the error probabilities for the transistors;
3. Propagate the stochastic streams from the primary inputs to the primary outputs in both the reliable and unreliable circuits;
4. Use XOR and OR gates to decode the joint error probability of the circuit from the obtained stochastic bit streams.

3.3. Validation by Simulations

For validating the applicability and the accuracy of the transistor-level stochastic models, the proposed approach is compared with the transistor-level Monte Carlo (MC) simulation and the gate-level SCM approach for the ISCAS-85 [57] benchmarks. Simulations were performed on a 2.60-GHz Intel microprocessor with 4 GB memory. In the MC simulation, random input vectors are applied and faults are randomly injected into the circuits. The circuit reliability is then obtained by the statistical outcomes using a large number of simulation runs. Compared to the MC simulation, the proposed stochastic approach is more efficient as it requires a significantly smaller number of pseudo-random generations in the stochastic computing process. This is confirmed by the simulation results shown in Table 3.4. In the MC simulation, a total number of one million simulations were run for each circuit to ensure a relatively stable output reliability, while in the stochastic approach, a sequence length of 10,000 bits were used and produced a relatively stable output reliability. It can be seen that while both approaches provide an accurate evaluation of circuit reliability, the proposed approach requires a significantly smaller runtime compared to the MC simulation.

Table 3.4. Simulation results for ISCAS-85 benchmarks by Monte Carlo simulation, the gate-level SCM approach and the proposed transistor-level approach.

Circuit	*Characteristics*			*Monte Carlo Simulation* $\varepsilon = 10^{-3}$ $sample = 1,000,000$		*SCM* $\varepsilon_{gate} = 1 - (1 - 10^{-3})^{n}$ $L = 1,000$, $input\ (max) = 1,000$		*Proposed stochastic approach* $\varepsilon = 10^{-3}$ $L = 10,000$	
	gates	**PIs**	**POs**	***PF***	***Time (m)***	***PF***	***Time (s)***	***PF***	***Time (s)***
C17	6	5	2	9.1×10^{-3}	1.5	1.94×10^{-2}	0.03	8.9×10^{-3}	0.26
C432	250	36	7	5.5×10^{-2}	52	2.31×10^{-1}	13.1	5.6×10^{-2}	7.2
C499	202	41	32	6.05×10^{-2}	93	2.98×10^{-1}	11.5	6.01×10^{-2}	11.7
C880	383	60	26	7.01×10^{-2}	105	3.67×10^{-1}	21.6	7.27×10^{-2}	16.2
C1355	546	41	32	8.47×10^{-2}	148	4.73×10^{-1}	29.8	8.62×10^{-2}	23.6
C1908	880	33	25	1.44×10^{-1}	243	6.59×10^{-1}	40.0	1.43×10^{-1}	36.4
C2670	1193	157	64	1.73×10^{-1}	401	7.69×10^{-1}	67.2	1.71×10^{-1}	52.2
C3540	1669	50	22	2.21×10^{-1}	612	8.22×10^{-1}	88.1	2.19×10^{-1}	81.0
C5315	2307	178	123	3.05×10^{-1}	870	8.89×10^{-1}	132.7	2.97×10^{-1}	104

In the gate-level SCM approach, it is assumed that the correct functioning of a gate requires the correct functioning of all its transistors. Thus, a simple equation is used to relate the reliability of the transistors to that of a gate, i.e., $\varepsilon_{\text{gate}} = 1 - (1 - \varepsilon_{\text{transistor}})^{\text{n}}$, where n is the number of transistors in the gate. The proposed transistor-level approach considers the gate topology and the applied input vectors, so it produces different error rates for different types of gates. Therefore, in Table 3.4 the gate-level approach results in a difference as large as 400% compared to the transistor-level approach. A lower circuit reliability is generated due to the use of a conservative gate error rate in the gate-level approach. Table 3.5 shows the results for stuck-at faults. It can be seen that accuracy and efficiency are achieved for both stuck-at-1 and stuck-at-0 faults. Albeit beyond the scope of this

manuscript, the bit-parallel nature of the proposed approach can be further explored to reduce its computational complexity through the potential parallelization of the stochastic simulation.

Table 3.5. Simulation results for ISCAS-85 benchmarks for stuck-ON/OFF errors

Circuit	*Stuck-ON error*						*Stuck-OFF error*					
	MC simulation $\varepsilon = 10^{-3}$ $sample = 1,000,000$		*SCM* $\varepsilon_{gate} = 1-(1-10^{-3})^{n}$ $L = 1,000$ $input\ (max) = 1,000$		*Proposed approach* $\varepsilon = 10^{-3}$ $L = 10,000$		*MC simulation* $\varepsilon = 10^{-3}$ $sample = 1,000,000$		*SCM* $\varepsilon_{gate} = 1-(1-10^{-3})^{n}$ $L = 1,000$ $input\ (max) = 1,000$		*Proposed approach* $\varepsilon = 10^{-3}$ $L = 10,000$	
	PF	*Time (m)*	*PF*	*Time (s)*	*R*	*Time (s)*	*PF*	*Time (m)*	*R*	*Time (s)*	*R*	*Time (s)*
C17	5.1×10^{-3}	1.5	9.5×10^{-2}	0.03	4.8×10^{-3}	0.26	4.4×10^{-3}	1.5	9.1×10^{-2}	0.03	4.5×10^{-3}	0.26
C432	4.3×10^{-2}	52	1.4×10^{-1}	13.1	3.9×10^{-2}	7.2	3.2×10^{-2}	52	1.2×10^{-1}	13.1	3.3×10^{-2}	7.2
C499	5.6×10^{-2}	93	1.8×10^{-1}	11.5	5.7×10^{-2}	11.7	5.1×10^{-2}	93	1.7×10^{-1}	11.5	5.2×10^{-2}	11.7
C880	6.1×10^{-2}	105	2.8×10^{-1}	21.6	6.3×10^{-2}	16.2	6.0×10^{-2}	105	2.6×10^{-1}	21.6	5.8×10^{-2}	16.2
C1355	6.9×10^{-2}	148	3.5×10^{-1}	29.8	7.2×10^{-2}	23.6	6.2×10^{-2}	148	3.1×10^{-1}	29.8	6.2×10^{-2}	23.6
C1908	1.2×10^{-1}	243	4.1×10^{-1}	40.0	1.2×10^{-1}	36.4	1.1×10^{-1}	243	3.9×10^{-1}	40.0	1.1×10^{-1}	36.4
C2670	1.6×10^{-1}	401	5.5×10^{-1}	67.2	1.6×10^{-1}	52.2	1.7×10^{-1}	401	5.8×10^{-1}	67.2	1.7×10^{-1}	52.2
C3540	2.0×10^{-1}	612	6.8×10^{-1}	88.1	1.9×10^{-1}	81.0	1.9×10^{-1}	612	6.9×10^{-1}	88.1	2.0×10^{-1}	81.0
C5315	2.7×10^{-1}	870	7.4×10^{-1}	132.7	2.7×10^{-1}	104	2.6×10^{-1}	870	7.6×10^{-1}	132.7	2.5×10^{-1}	104

3.4. Summary

Accurate and efficient reliability evaluation techniques are very important as CMOS technology continues to scale in the nanometric regime. Faults and defects are likely to affect the correct operation of individual transistors in the logic gates of a combinational circuit; so for an accurate and realistic reliability analysis, the gate error rate should be derived from the transistor error probability, while also considering the effects of different transistor errors. A transistor-level analysis accounts for features such as temporal signal sequences, logic gate topology and

different input vectors to determine the reliable operation of circuits; hence, it is more accurate than existing gate-level evaluation methodologies.

This chapter has presented such an approach using stochastic transistor models (STMs) for the evaluation of nanometric CMOS circuits. In the proposed model, a transistor has been modeled as a probabilistic switch such that the probabilistic distribution of its states (ON, OFF and IND) has been determined by the input signal probability and its probabilistic input has been modeled using stochastic computational models (SCMs). Logic gates have been modeled using STMs and at the circuit level, a new evaluation approach has been proposed to assess the reliability of a circuit.

Simulation results have shown the accuracy and efficiency of the proposed approach. Since signal correlation is accounted for in the distribution pattern of the stochastic binary streams, the proposed approach requires a significantly smaller runtime compared to simulation-based approaches (such as the Monte Carlo method), while providing a more accurate result compared to gate-level evaluation methodologies. The proposed approach is scalable for the evaluation of large circuits and can further be improved by considering more accurate physical models of the transistor.

CHAPTER 4

Variability and Variation-Induced Error Analysis

Over the last few decades, As CMOS technology scales into the nanometer regime, random parameter variations become a prominent feature and start to dominate the behaviors of CMOS logic circuits [1] [5] [6] [58]. Among various sources, process variations are caused by the randomness or imprecision introduced in the CMOS fabrication process [7] [17]. These are mainly due to random dopant fluctuations and line-edge and line-width roughness [1]. Statistical models are developed in [15] for the random dopant fluctuations in MOS transistors. Variation-induced parameter fluctuations in MOSFETs have been studied using simulations [18] and variation-tolerant techniques [17]. Voltage variations have increasingly been a concern as the supply voltage (V_{DD}) scales to reduce power dissipation [59]. Variations also exist in the lifetime of devices [24], due to the threshold voltage shifts caused by negative-bias temperature instability (NBTI) and hot-carrier injection (HCI), as well as gate current shifts caused by time-dependent dielectric breakdown (TDDB) [60].

As process, voltage and temperature (PVT) variations become severe in scaled CMOS technologies, extensive research has been devoted to the modeling of delay and power variability [61] [62]. In contrast, there was inadequate effort toward the understanding of the functional variability of CMOS circuits. While many approaches have been developed for the evaluation of circuit reliability and soft error rates (SERs), most are focused on the error propagations at the gate level and thus the errors are considered technology-independent. The authors in [28] investigated the reliability of CMOS logic gates under threshold voltage variations due to random dopant fluctuations. It is shown that the reliability drastically varies with respect to different technology generations, supply voltages and input vectors. Work has also been done to model the long-term reliability of circuits affected by aging-induced variations [8] [60] [63].

In this chapter, the impacts of variations on the functions of CMOS circuits are investigated by the functional variability, which is the probability of the functional output of a circuit falling off the noise margins. Initially, analytical models are developed for the modeling of transistor variability under parameter variations. These models are then extended to consider the variability of logic gates consisting of several transistors. In this approach, transistors are modeled as probabilistic switches and their operation is affected by static and dynamic variations. This model accounts for the effects of process and voltage variations and are thus more realistic and accurate compared to previous models. As the functional variability propagate through a logic circuit, variation-induced error rate (ViER) increases quickly and begin to hamper gate's functionality robustness as technology scales beyond 16nm. The proposed stochastic transistor and gate models can further be used in the evaluation of circuit variability and variation-induced error rate. This enables us to gain insights into the impacts of variations in advanced CMOS processes such as those of 22nm and 16nm.

The rest of this chapter is organized as follows. Section 4.1 presents the analytical variability models for transistors and gates. In Section 4.2, the proposed stochastic

models in previous section are elaborated for variability and variation-induced error rate analysis. Section 4.3 presents ViER circuit analysis. Simulation results are presented in Section 4.4. Section 4.5 gives conclusion.

4.1. Variability and Related Models

Parameter variations have been a major concern in circuit design due to their impacts on the performance, power and robustness of CMOS circuits. Section 1.1 reviews several sources of variation in CMOS circuits. As shown in Fig. 3.1, the transistor is considered to work as a switch in digital design. The transistor's ON/OFF state is determined by an overdrive voltage (V_{OD}). An NMOS transistor is ON if $V_{OD,N} = V_{gs} - V_{th,N} > 0$ and it is OFF otherwise; a PMOS transistor is ON if $V_{OD,P} = V_{sg} - |V_{th,P}| > 0$ and it is OFF otherwise. Due to process variations, as discussed in Section 2, the V_{th} becomes a probabilistic variable, which follows a Gaussian distribution $N(V_{th}, \sigma_{Vth})$. Since various sources of RDFs and LER are statistically independent [17], the standard deviation for V_{th}, due to the effect of process variations, is given by [18]

$$\sigma_{Vth} = \sqrt{\sigma^2_{Vth,RDF} + \sigma^2_{Vth,LER}} \tag{4.1}$$

where $\sigma_{V_{th,RDF}}$ and $\sigma_{V_{th,LER}}$ are given by (1.1) and (1.2) respectively. Then the probability density function (PDF) and the cumulative distribution function (CDF) can be obtained as follows:

$$pdf_{th,N}(v) = \frac{1}{\sigma_{Vth,N}\sqrt{2\pi}} \exp\left[\frac{-(v-V_{th,N})^2}{2\sigma_{Vth,N}{}^2}\right] \tag{4.2}$$

$$cdf_{th,N}(v) = \frac{1}{2} + \frac{1}{2} \cdot erf\left[\frac{(v-V_{th,N})}{\sqrt{2}\sigma_{Vth,N}}\right], \tag{4.3}$$

for NMOS transistors, and

$$\mathrm{pdf}_{\mathrm{th,P}}(\mathrm{v}) = \frac{1}{\sigma_{\mathrm{Vth,P}}\sqrt{2\pi}} \exp\left[\frac{-(\mathrm{v}-|\mathrm{V}_{\mathrm{th,P}}|)^2}{2{\sigma_{\mathrm{Vth,P}}}^2}\right], \tag{4.4}$$

$$\mathrm{cdf}_{\mathrm{th,P}}(\mathrm{v}) = \frac{1}{2} + \frac{1}{2} \cdot \mathrm{erf}\left[\frac{(\mathrm{v}-|\mathrm{V}_{\mathrm{th,P}}|)}{\sqrt{2}\sigma_{\mathrm{Vth,P}}}\right], \tag{4.5}$$

for PMOS transistors, where erf [ ] is the Gauss error function.

Due to the effect of power supply noise, the supply voltage $\mathrm{V_{dd}}$ also becomes probabilistic and follows a Gaussian distribution $\mathrm{N}(\mathrm{V_{dd}}, \sigma_{\mathrm{p}})$. The PDF and CDF are given by:

$$\mathrm{pdf}_{\mathrm{p}}(\mathrm{v}) = \frac{1}{\sigma_{\mathrm{p}}\sqrt{2\pi}} \exp\left[\frac{-(\mathrm{v}-\mathrm{V_{dd}})^2}{2{\sigma_{\mathrm{p}}}^2}\right], \tag{4.6}$$

$$\mathrm{cdf}_{\mathrm{p}}(\mathrm{v}) = \frac{1}{2} + \frac{1}{2} \cdot \mathrm{erf}\left[\frac{(\mathrm{v}-\mathrm{V_{dd}})}{\sqrt{2}\sigma_{\mathrm{p}}}\right]. \tag{4.7}$$

Although the interference with ground voltage ($\mathrm{V_{ss}}$) can be similarly modeled, it is usually neglected during a transient analysis [65].

Based on (4.2) – (4.7), the probability that a transistor is ON ($\mathrm{P_{ON}}$) or OFF ($\mathrm{P_{OFF}}$) can be obtained as a function of the input voltage, as follows:

$$\mathrm{P}_{\mathrm{ON,NMOS}}(\mathrm{v_{in}}) = \mathrm{cdf}_{\mathrm{th,N}}(\mathrm{v_{in}}), \tag{4.8}$$

$$\mathrm{P}_{\mathrm{OFF,NMOS}}(\mathrm{v_{in}}) = 1 - \mathrm{P}_{\mathrm{ON,NMOS}}(\mathrm{v_{in}}), \tag{4.9}$$

$$\mathrm{P}_{\mathrm{ON,PMOS}}(\mathrm{v_{in}}) = \mathrm{cdf}_{\mathrm{th,P}}(\overline{\mathrm{V_{dd}}} - \mathrm{v_{in}}) = \Pr\left[(\overline{\mathrm{V_{dd}}} - \mathrm{v_{in}}) - \left|\overline{\mathrm{V_{th,P}}}\right| > 0\right] = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\mathrm{v}} \mathrm{pdf}_{\mathrm{th,P}}(\mathrm{u})\mathrm{du}\right]\mathrm{pdf}_1(\mathrm{v})\mathrm{dv}\,, \tag{4.10}$$

$$\mathrm{P}_{\mathrm{OFF,PMOS}}(\mathrm{v_{in}}) = 1 - \mathrm{P}_{\mathrm{ON,PMOS}}(\mathrm{v_{in}}), \tag{4.11}$$

where $\overline{V_{dd}}$ means a variable and pdf_1 is for the Gaussian distribution $N\left((V_{dd} - v_{in}),\ \sigma_p\right)$.

The computation of (4.10) is illustrated in Fig. 4.1. The cumulative probability that $(\overline{V_{dd}} - v_{in})$ is larger than $\left|\overline{V_{th,P}}\right|$ (i.e., $V_{OD,P} > 0$), varies with respect to σ_p, $\sigma_{Vth,P}$ and v_{in}.

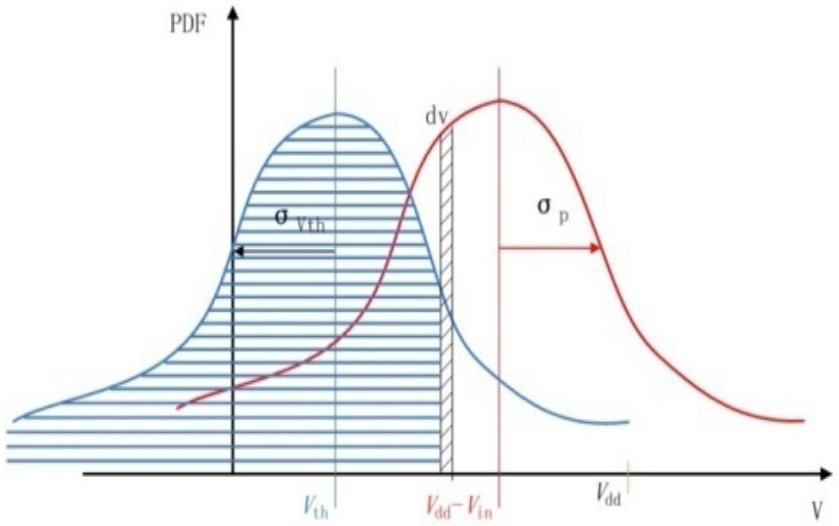


Fig. 4.1. Evaluation of probability that the PMOS transistor switches ON for a given v_{in}.

4.1.1. Inverter

Given the variability models for the NMOS and PMOS transistors, the variability model for a CMOS inverter, shown in Fig. 4.2, can be derived. The inverter consists of an NMOS and a POMS transistor in its pull-down and pull-up network. Given an input voltage v_{in}, the switching probability of the PMOS transistor T_1 is:

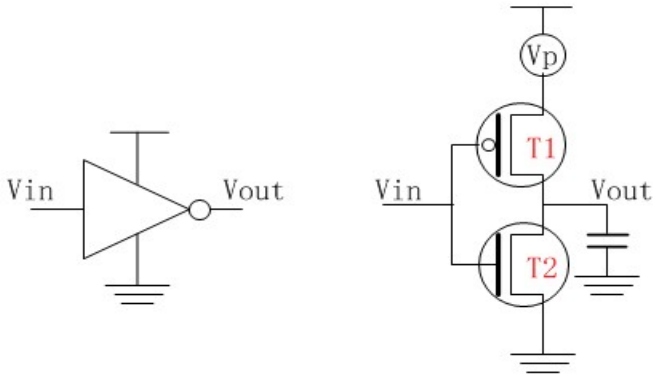


Fig. 4.2. The schematic and structure of a CMOS inverter.

$$P_{ON,T1} = P_{ON,PMOS}(v_{in}), \tag{4.12}$$

$$P_{OFF,T1} = 1 - P_{ON,T1}, \tag{4.13}$$

where $P_{ON,PMOS}(v_{in})$ is given by (4.10).

For the NMOS transistor T_2:

$$P_{ON,T2} = P_{ON,NMOS}(v_{in}), \tag{4.14}$$

$$P_{OFF,T2} = 1 - P_{ON,T2}, \tag{4.15}$$

where $P_{ON,NMOS}(v_{in})$ is given by (4.8).

Assuming each transistor work independently, the probabilities that the inverter outputs a “0” and “1” are given by:

$$P_{INV}(\text{output} = \text{"0"}) = P_{OFF,T1} \cdot P_{ON,T2}, \tag{4.16}$$

$$P_{INV}(\text{output} = \text{"1"}) = P_{ON,T1} \cdot P_{OFF,T2}. \tag{4.17}$$

Note that $P_{INV}(\text{output} = \text{"0"}) + P_{INV}(\text{output} = \text{"1"}) \neq 1$ as it is possible that both the pull-up and pull-down networks are ON or OFF. Assuming further that the inverter only produces output signals within the noise margin when both transistors function correctly; the functional variability of the inverter is then defined as the probability that the output falls off the noise margin, i.e.,

$$V_{INV}(\text{output} = \text{"0"}) = 1 - P_{INV}(\text{output} = \text{"0"}), \quad (4.18)$$

$$V_{INV}(\text{output} = \text{"1"}) = 1 - P_{INV}(\text{output} = \text{"1"}). \quad (4.19)$$

The output voltage of the inverter v_{out} is affected by the power supply noise. If the noise to the ground voltage V_{ss} is negligible, the voltage variation ΔV_{dd} will be the only source that degrades the gate output. The impact of intrinsic noises such as thermal noise is not considered as it is not as severe to affect a digital circuit as technology scales. This imperfect output signal is then propagated to the next gate as the input voltage, v_{in}, which subsequently affects the reliable operation of the transistors in this gate.

4.1.2. NAND Gate

The two-input NAND gate is used as an example to illustrate the variability model of a logic gate with multiple inputs (as shown in Fig. 4.3). Given the two inputs v_{in1} and v_{in2} for the PMOS transistors T_1 and T_2, we obtain

$$P_{ON,T1} = P_{ON,PMOS}(v_{in1}), \quad (4.20)$$

$$P_{OFF,T1} = 1 - P_{ON,T1}, \quad (4.21)$$

$$P_{ON,T2} = P_{ON,PMOS}(v_{in2}), \quad (4.22)$$

$$P_{OFF,T2} = 1 - P_{ON,T2}. \quad (4.23)$$

For the NMOS transistors T_3 and T_4:

$$P_{ON,T3} \approx cdf_{th,N}(v_{in1}) = P_{ON,NMOS}(v_{in1}). \quad (4.24)$$

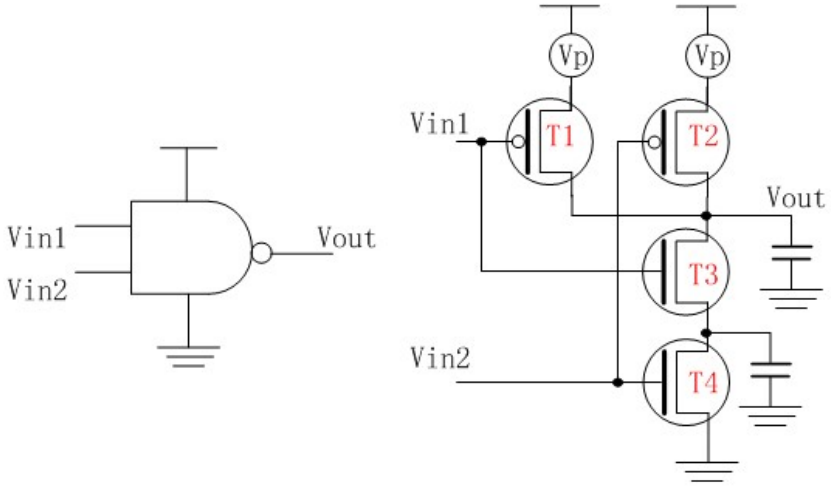


Fig. 4.3 The schematic and structure of a NAND2 gate.

Although each transistor is assumed to work independently, the operations of transistors connected in series are dependent; therefore

$$P_{OFF,T3} = 1 - P_{ON,T3}, \quad (4.25)$$

$$P_{OFF,T3} = P_{ON,NMOS}(v_{in2}), \quad (4.26)$$

$$P_{OFF,T4} = 1 - P_{ON,T4}. \quad (4.27)$$

The output probability of the NAND can be derived based on the gate topology and operating mechanism, i.e.,

$$P_{NAND2}(\text{output} = "0") = P_{OFF,T1} \cdot P_{OFF,T2} \cdot P_{ON,T3} \cdot P_{ON,T4} \quad (4.28)$$

and the variability of NAND for an output of "0" is given by

$$V_{NAND2}(\text{output} = \text{"0"}) = 1 - P_{NAND2}(\text{output} = \text{"0"}). \quad (4.29)$$

For NAND2, the input vectors "00," "01" and "10," are expected to produce a "1" at the output, so it fails to produce a "1" if T_1 and T_2 are both OFF or T_3 and $T_4$are both ON. This indicates that the variability of NAND for an output of "1" is given by

$$V_{NAND2}(\text{output} = \text{"1"}) = P_{OFF,T1} \cdot P_{OFF,T2} + P_{ON,T3} \cdot P_{ON,T4} - P_{OFF,T1} \cdot P_{OFF,T2} \cdot P_{ON,T3} \cdot P_{ON,T4}, \quad (4.30)$$

as well as

$$P_{NAND2}(\text{output} = \text{"1"}) = 1 - PF_{NAND2}(\text{output} = \text{"1"}). \quad (4.31)$$

Using a similar procedure, the variability model for a different type of logic gates can be derived.

4.2. Variation-Induced Errors

As discussed in Section 4.1, we can characterize a transistor's state as ON, OFF, or partially ON according to its conducting current. A degraded transistor's state could be flipped in the sense that the parameter variations alter its conducting current, which could further drive a gate output voltage falling off the noise margins. For a static CMOS gate, the value of output logic is established by considering the states of the pull-up and pull-down networks as follows:

- If the pull-up network is ON and the pull-down network is OFF, then the output is 1.
- If the pull-up network is OFF and the pull-down network is ON, then the output is 0.
- If the pull-up network is OFF and the pull-down network is OFF, then the

output is Z, which depends on the previous value.

- If the pull-up network is ON and the pull-down network is ON, or any of the networks is IND, then the output is X.

For digital logic circuits, the concept of noise margin is introduced for characterizing a technology or design's noise immunity. It defines the legal region for logical '1' and '0', while the indeterminate region in is forbidden for reliable computing. The noise margin of a logic family can be related to the DC voltage characteristics, as shown in Figure 4.4. The slope -1 (unit gain) points are typically taken as defining points for legal region. Therefore, for an output voltage, it has three logic levels: "1", "0" and "X".

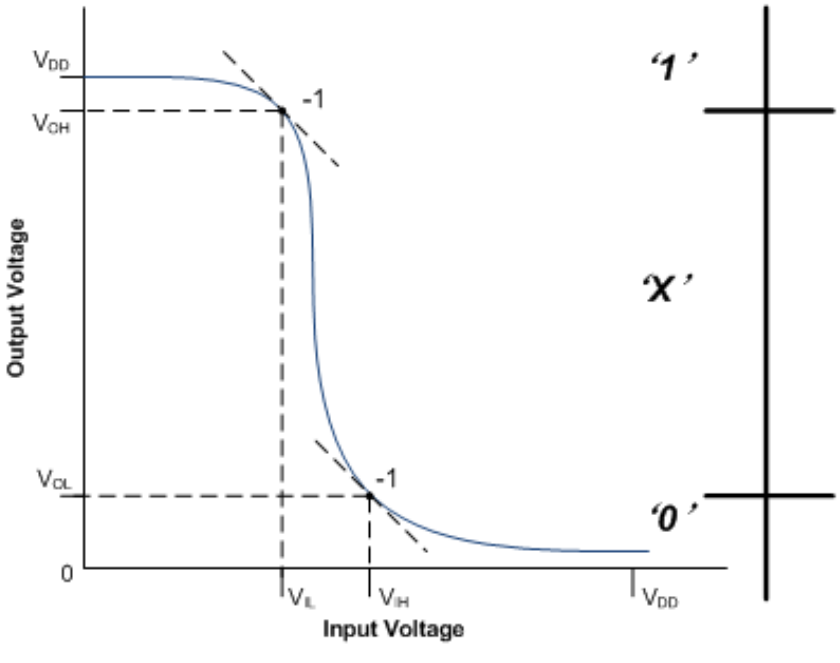


Figure 4.4. Three-level voltage logic

Therefore, *variability* is defined according to the noise margin related three level logic characteristics. When the output is supposed to be "1" for example, the variability of the gate/circuit is calculating as the probability of output having logic "X" and "0". Previous studies show that *functional variability* could induce

leakage and delay errors. However, the effects of *variability* on the reliable functions of CMOS transistors, gates and circuits have not been adequately addressed.

Functional variability does not necessarily lead to functional error, i.e. bit flipping of a gate's output. It can be restored by following CMOS logic gates. Assume that a signal is disturbed by variations and deviates from the nominal voltage level, the CMOS regenerative property will ensure a degraded signal gradually converge back to its nominal voltage level after passing through a number of logic stages. This concept is illustrated in Figure 4.5, by plotting the transient response of a chain of CMOS inverters, the input signal V_0 to the chain is logic "X" with a degraded amplitude. We can observe the voltage will be gradually restored to legal '1' for V_3 within the noise margin.

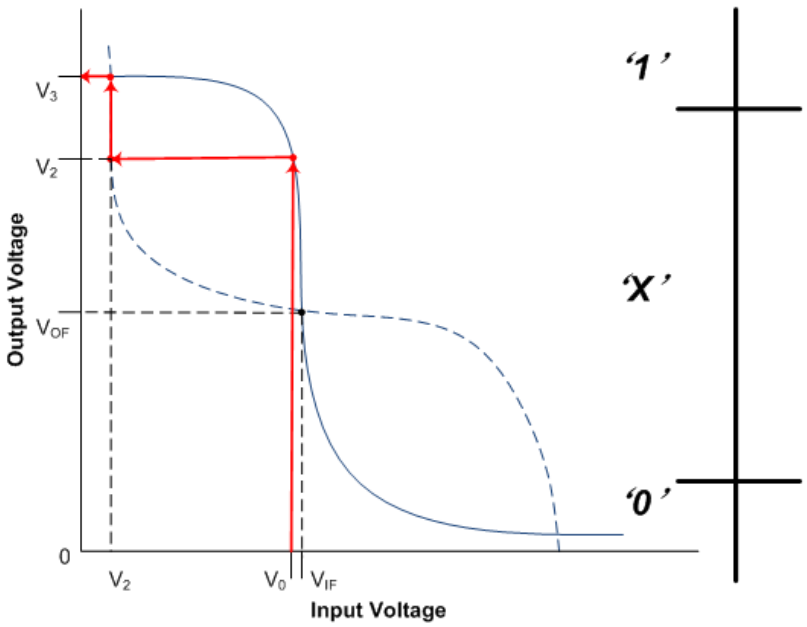


Figure 4.5. Regenerative property of a CMOS gate

Figure 4.5 shows the excepted value for input $V_0 = V_{IF}$ (where the V_{IF} stands for input flipping voltage), input signal at other voltage levels will eventually be restored to either "1" or "0". It means the voltage degradation before V_{IF} can be

restored to its nominal value while the degradation crossing the flipping voltage V_{OF} will be erroneously restored in the opposite direction. Therefore, from the perspective of reliable operation, functional variation-induced error rate (ViER) can be defined as the probability that a functional output of a circuit falls off V_{OF} that leads to flipping logic state. As shown in Figure 4.6, the legal "1" and "0" is still defined by noise margin criteria, while the intermediate region is now devided into two logic levels "+X" and "-X" with the flipping point as boundary. If the nominal voltage of the signal is "1" for example, it could be restored if the degraded signal is "+X" while it will be flipped if it is "-X" or "0". The probability distribution of the signal gradually decreases as it away from its nominal value. Gate/circuit ViER of the flipping error is much less than that of gate/circuit variability. Therefore, the effects of *variability* on the reliable functions of CMOS transistors, gates and circuits can be quantitatively measured in terms of ViER.

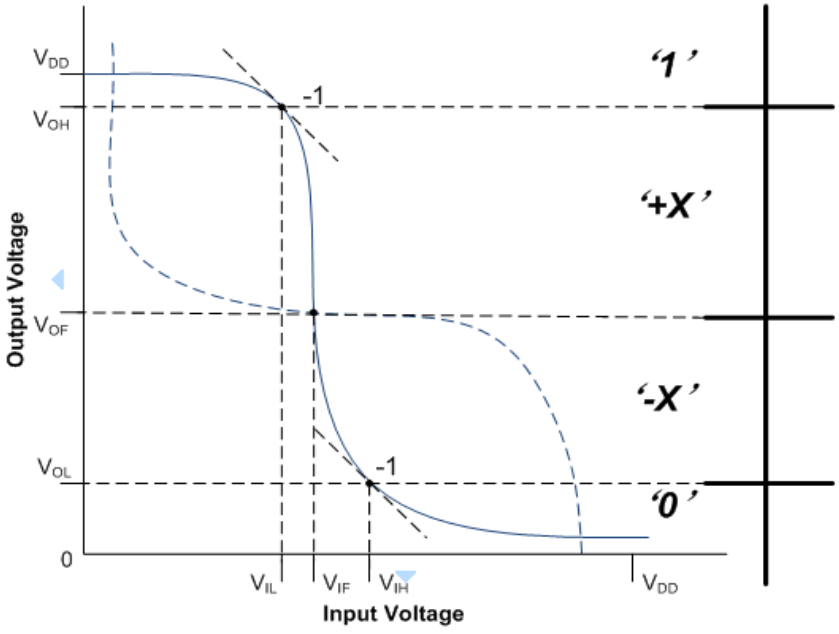
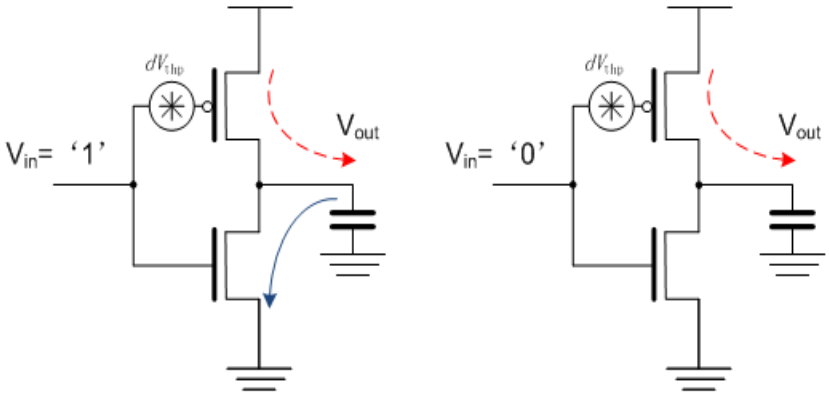


Figure 4.6. Four-level voltage logic

SPICE simulator is the most accurate tool for emulating circuit behaviors. It can perform device, gate and circuit-level analysis. However, its complexity increases with increasing module size. For simulating modules under parameter variations,

the degraded transistor model, shown in Figure 1.1 and Figure 1.2 can be used. Then an erroneous state is detected via observing its output voltage, in terms of variability or variation-induced error rate (ViER). Since transistor is a voltage-controlled current source, usually a transistor is measured according to its conducting current, a transistor can be ON, partially ON, or OFF according to its conducting current. For the testing of variability and ViER, we connect a transistor under test (TUT) with an active load to measure its output voltage. This is illustrated in Figure 4.7. When a PMOS TUT affected by variations is considered, a fault-free NMOS is connected with the TUT to produce a voltage output. When the input is logical "1", shown in Figure 4.7(a), the fault-free NMOS load will be fully conducting. We observe the V_{out}, if the V_{out} goes beyond corresponding noise margin, then it means the TUT becomes ON or partially ON, which is supposed to be OFF. Therefore, the TUT error rate given high input $PF_{pmos,1}$ is defined as the frequency of occurrence that produce erroneous output from Monte Carlo simulations. When the input is logical low, the NMOS switch OFF, we again observe the output, if the TUT become partially ON or even OFF. The output will move outside noise margin. In the same way we define $PF_{pmos,0}$.

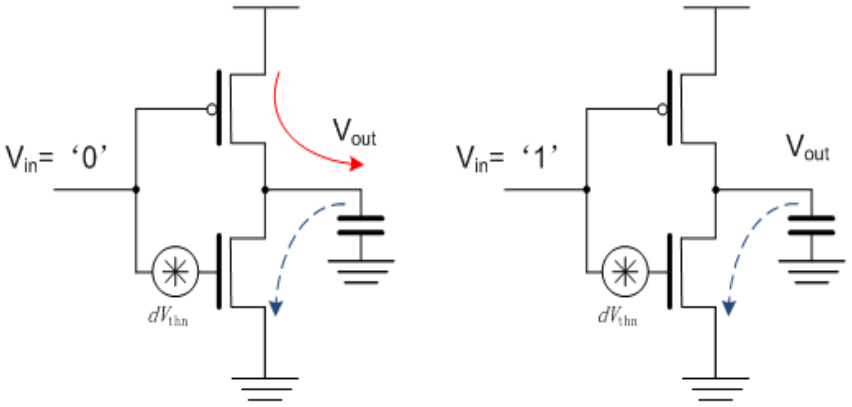


(a) (b)

Figure 4.7. PMOS TUT simulation: (a) input high; (b) input low.

Similarly, we can test a NMOS TUT with a fault-free PMOS load, as shown in Figure 4.8. We can estimate $PF_{nmos,0}$ and $PF_{pmos,1}$. The probability of failure of

a transistor varies in accordance with its input signal, feature size, etc. Therefore for precise analysis, several different PFs need to be generated from SPICE for a specific transistor.



(a) (b)

Figure 4.8. NMOS TUT simulation: (a) input low; (b) input high

CMOS gates and circuits also can be estimated by SPICE Monte Carlo simulations with degraded transistor model. However, it takes a very large computational time and also memory overhead for large scale circuits. This motivates us to develop a fast and accurate error propagation approach.

4.3. Variation-induced Error Rate (ViER) Analysis of Logic Circuits

An accurate and efficient statistical methodology for ViER analysis is crucial for high reliability and yield IC design in nanometric CMOS technology. SPICE model and simulator is the most popular and accurate method used by the VLSI community to emulate circuit's behavior. Monte Carlo (MC) simulation has been widely used for statistical analysis. It is intuitive and easy to implement to estimate statistical circuit activities using SPICE Monte Carlo simulation, and its results are believed to be quite accurate. However, a large number of simulation runs must be executed to reach convergence. SPICE simulation takes extremely

large runtime as well as memory overhead. Therefore reliability estimation using SPICE MC up to circuit level is most time consuming and intractable. Therefore, scalable analytical or simulation-based approaches are needed for circuit-level analysis. Our methodology is to enable accurate estimation using SPICE MC to perform device-level ViER analysis, while perform gate- and circuit-level analysis using enhanced stochastic transistor model (STM), as shown in Figure 4.9.

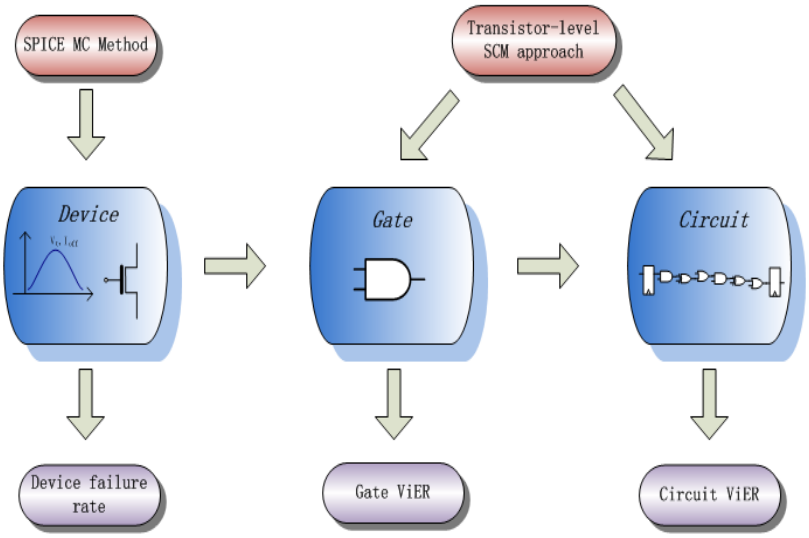


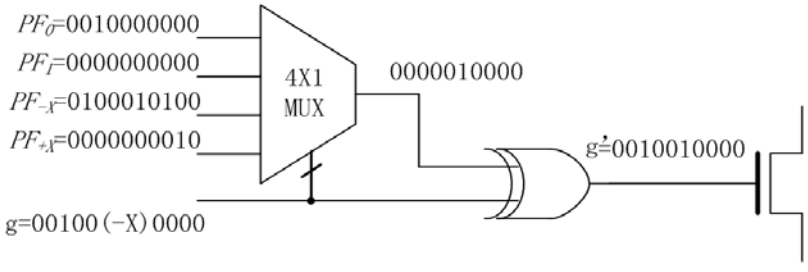
Figure 4.9. Proposed statistical methodology for accurate and efficient ViER analysis

Reliability evaluation can be performed at device, gate and circuit levels. Accuracy should be maintained at each level. At device and gate level, statistical characteristics of CMOS technology are gathered and input into SPICE simulator, by simulation under variation effects, the reliability margin of scaled CMOS devices can be accurately captured. At circuit level, circuit design needs to be analyzed accurately and efficiently under various reliability issues. Therefore a fast and accurate circuit-level approach need to be incorporated, we use stochastic transistor model (STM) for circuit-level analysis. The STM needs to be elaborated

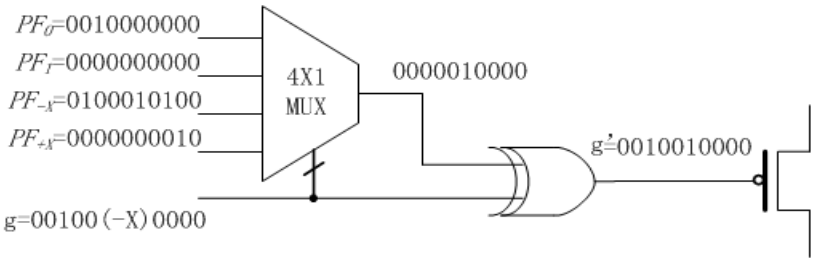
to model plenty of physical details to guarantee accuracy in comparison with SPICE simulator.

As discussed in Chapter 3, three logic transistor-level stochastic models are proposed, they can handle the three-level logic and perform gate as well as circuit variability analysis. However, it has two major shortcomings: 1) variability will only be partially propagated; 2) variations on signal inputs are not modeled, this will lead to a great deviation. Our SPICE simulations show that different inputs result in orders of magnitude difference on probability of failure.

Four-level logic needs to be used for variation-induced error rate analysis. As per the definition of four-level logic, a functional error occurs when a functional output of a gate/circuit falls off the flipping point. For example, if the nominal voltage of the signal is "1", it could be restored if the degraded signal is "+X" while it will be flipped if it is "-X" or "0". Therefore $\mathrm{ViER}(\mathrm{out}=1)=\Pr(\mathrm{out}=0)+\Pr(\mathrm{out}=-\mathrm{X})$. In spite of that, our experimental results also indicate that the degraded logic "+X" results in much higher ViER as it feeding into the following gates (100x~1000x increase). This degradation need to be modeled thus result in four different error rates: PF_0, PF_1, $\mathrm{PF}_{+\mathrm{X}}$, $\mathrm{PF}_{-\mathrm{X}}$. These device parameters need to be extracted from SPICE simulations. We then propose a MUX to configure error rates with respect to different input values. This leads to a new stochastic transistor model shown in Figure 4.10.



(a)



(b)

Figure 4.10. Elaborated Stochastic transistor models: (a) NMOS (b) PMOS.

The mapping between gate input and transistor state is differentiated by different input value, as shown in Table 4.1.

Table 4.1. Mapping between the gate input and the transistor operations: (a) input high ($g = 1$); (b) input low ($g = 0$)

(a)

Transistor type / Gate input	NMOS State (St)	PMOS State (St)
$g = 0/-X$	OFF	ON
$g = 1$	ON	OFF
$g = +X$	Partially ON	Partially OFF

(b)

Transistor type / Gate input	NMOS State (St)	PMOS State (St)
$g = 0$	OFF	ON
$g = 1/+X$	ON	OFF
$g = -X$	Partially OFF	Partially ON

4.4. Simulation Results

The data used in the analysis are based on the 35nm MOSFET model in [18] and adapted to the 32nm, 22nm, 16nm HP Predictive Technology Model (PTM) [66].

Device and design parameters such as T_{ox}, L_{eff}, V_{th} and V_{dd} are adopted from the PTM models.

In Table 4.2, the simulation results for INV, NAND2, NAND3, NOR2 and NOR3 are shown using 32nm, 22nm, 16nm HP PTMs; V_{th} variations are calculated using equations (1) and (2) and calibrated in respect of 35nm MOSFET simulation data. The V_{dd} variation (ΔV_{dd}) is set to $\pm 5\%$ and it is assumed that the voltage drop scales as technology advances. Various CMOS gates are sized for a unit resistance and optimized delay [65].

From Table 4.2, we observe that the variability varies with respect to different parameters (including L_{eff}, V_{th}, V_{dd}, etc.), different input vectors and gate types. As technology scales, the variability increases quickly and begin to hamper gates' functional robustness. For example, the worst-case variability for INV increases from 4.55×10^{-15} at 32nm feature size to 0.00112 at 16nm feature size. Different input vectors can result in different variability that varies by several orders of magnitude, due to the difference in transistor variability.

Table 4.2. Variability evaluation of CMOS logic gates

CMOS Gate Type	*Predictive Technology Models*								
	32nm HP PTM			*22nm HP PTM*			*16nm HP PTM*		
	Average Var	*Worst-case Var*	*Best-case Var*	*Average Var*	*Worst-case Var*	*Best-case Var*	*Average Var*	*Worst-case Var*	*Best-case Var*
INV	2.28×10^{-15}	4.55×10^{-15}	$< 10^{-16}$	8.83×10^{-9}	1.72×10^{-8}	4.14×10^{-10}	5.64×10^{-4}	1.12×10^{-3}	6.67×10^{-6}
NAND 2	2.28×10^{-15}	4.55×10^{-15}	$< 10^{-16}$	2.07×10^{-10}	4.14×10^{-10}	$< 10^{-16}$	7.21×10^{-6}	1.15×10^{-5}	4.45×10^{-11}
NAND 3	1.71×10^{-15}	4.55×10^{-15}	$< 10^{-16}$	1.55×10^{-10}	4.14×10^{-10}	$< 10^{-16}$	2.53×10^{-6}	6.67×10^{-6}	3.33×10^{-16}
NOR2	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	8.62×10^{-9}	1.72×10^{-8}	3.33×10^{-16}	5.61×10^{-4}	1.12×10^{-3}	8.42×10^{-8}
NOR3	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	6.46×10^{-9}	1.724×10^{-8}	$< 10^{-16}$	4.21×10^{-4}	1.12×10^{-3}	1.41×10^{-9}

In CMOS logic gates, pull-up networks (of PMOS transistors) are affected by V_{dd} variations, while pull-down networks (of NMOS transistors) are not. Therefore, the variability of NMOS and PMOS transistors tends to have different trends as

technology scales, which will in turn impact gate variability. For example, the input "1" produces the worst-case variability for INV at 16nm and 22nm feature sizes, while it is not the case at 32nm. This is because V_{dd} scales as technology scales while V_{th} almost remain unchanged. The scaled voltage difference between V_{dd} and V_{th} and the increased V_{th} variation degrade the variability of NMOS transistors from almost 0 (at 32nm) to 0.00112 (at 16nm). However, the variability of PMOS transistors degrades more slowly (from 4.55×10^{-15} to 6.67×10^{-6}). Since the PMOS suffers less from V_{th} variations and the power supply variation ΔV_{dd} decreases as technology scales, the variability of logic gates (INV, NAND2, NOR2, etc.) are dominated by PMOS variability at older technologies and by NMOS variability at later technologies.

Different gates tend to have different variability due to different gate structures and transistor sizes. NAND2 has lower variability than the INV due to its error masking characteristics. Larger gates (NAND3, NOR3) tend to have lower variability because they typically use larger transistors. NAND is more reliable than NOR in terms of variability at 22nm, 16nm while it is less reliable at 32 nm feature size, because NAND tends to use larger NMOS transistors, which mitigate the NMOS variability at later technologies (22nm and 16nm).

To validate the proposed variability model, Monte Carlo simulations have been performed using SPICE. The simulations were run on a 2.66-GHz Pentium microprocessor with 2 GB RAM, for the 16nm HP PTM with$V_{dd} = 0.7V$, and $T = 127℃$. By characterizing the voltage characteristic of a standard inverter, a "perfect" logical "1" has a voltage value in $0.655 \sim 0.7V$, and a logical '0' is in $0 \sim 0.0445V$. One million random patterns were simulated and the standard deviations of parameter variations were considered the same as those in the variability models. As can be seen in Table 4.3, the proposed variability evaluation approach produced very accurate results while achieving a speed up by several orders of magnitude.

Table 4.3. Accuracy and runtime comparisons of the proposed approach and Monte Carlo simulations using SPICE.

CMOS Gate	*16nm HP PTM*			
	SPICE MC		*Proposed Approach*	
	Variability	*Runtime*	*Variability*	*Runtime*
INV	4.62×10^{-4}	10391.22s	5.64×10^{-4}	<0.01s
NAND2	13.5×10^{-6}	11739.25s	7.21×10^{-6}	<0.01s
NOR2	3.13×10^{-4}	11331.17s	5.61×10^{-4}	<0.01s
NAND3	0.8×10^{-6}	15404.67s	2.53×10^{-6}	<0.01s
NOR3	2.11×10^{-4}	15221.86s	4.21×10^{-4}	<0.01s

For validating the enhanced stochastic transistor model, the proposed approach is compared with the SPICE Monte Carlo (MC) simulation. Table 4.4 shows the simulation results for typical CMOS logic gates. A total number of one million simulations were run for each device and gates in SPICE to ensure a relatively stable output reliability, while in the stochastic approach, a sequence length of 10,000 bits were used and produced a relatively stable output reliability.

Table 4.4. Variability estimation for logic gates

Method / **Gate type**	**SPICE Monte Carlo**		**Transistor-level SCM**		
	Variability	**Time**	**Variability**	**Time**	**difference**
INV	$\mathbf{432 \times 10^{-6}}$	7hr	$\mathbf{510.5 \times 10^{-6}}$	0.07s	***18.12%***
NAND2	$\mathbf{12.25 \times 10^{-6}}$	9hr	$\mathbf{14.5 \times 10^{-6}}$	0.08s	***18.37%***
NOR2	$\mathbf{281.5 \times 10^{-6}}$	9hr	$\mathbf{239.9 \times 10^{-6}}$	0.08s	***14.76%***

Finally, Circuit ViER analysis has been performed and the results are shown in Table 4.5. Once the voltage at a circuit node is flipped, it can also be restored inversely to erroneous state. Fortunately, under 16nm PTM technology environment, flipping behavior cannot be observed for gates and circuits. However, recalling equations (1.1), (1.2), We find that the variation could be proportional to $1/\lambda$, which means the amount of threshod variation doubles as λ scales to one half. Therefore, for research purpose, we double the variation under 16nm PTM environment, and then flipping error occurs. We then proposed our circuit simulation under this worse environment. Again, we first need SPICE simulator to characterize the device error rate. We ran into a difficulty for extracting transistor flipping error rate using SPICE MC. In order to maintain accuracy, we proposed to derive individual transistor error rate based on SPICE characterized gate library. The transistor error rates prepared for stochastic approach are fitted with a bunch of gates' error response data. Since in the stochastic approach, four logical values "0," "1," "+X,""-X," are considered. Therefore, we also need to characterize the transistor error rate in terms of different input logic values. In our simulation, we found that the degraded logic values, "+X,""-X," induce much higher error rate to the following gates. Therefore we also use SPICE simulator to characterize a factor that takes this effect into account. As per we discuss in Table 3.2, the output will produce a X when both pull-up and down networks are ON or partially ON, and it has a voltage range between nominal logic value and the flipping point, therefore, if we feed this degraded logic values from SPICE Monte Carlo into another gate's input, we can estimate the error rate increase based on a large amount of simulations. Therefore, the accuracy of our approach can be further validated in terms of flipping error. Table 4.5 provides our results on small benchmark circuits. 1,000,000 simulations were run for SPICE Monte Carlo and the sequence length for stochastic approach is 100,000. Logic-level SCM is also considered here, with accurate gate error rate provided by SPICE simulations.

Table 4.5. Flipping ViER estimation for small circuits

Circuit	*Characteristics*			*Enhanced STM* $L = 100,000$			*Logic SCM* $L = 100,000$		*SPICE Monte Carlo*	
	Gates	*PIs*	*POs*	*ViER*	*Time*	*Rel. error*	*ViER*	*Time*	*ViER*	*Time*
C17	***6***	***5***	***2***	$\mathbf{130 \times 10^{-6}}$	6.8s	34%	$\mathbf{81 \times 10^{-6}}$	3.2s	$\mathbf{199 \times 10^{-6}}$	23.2h
Majority	***10***	***5***	***1***	$\mathbf{327 \times 10^{-6}}$	15.3s	25%	$\mathbf{189 \times 10^{-6}}$	6.8s	$\mathbf{435 \times 10^{-6}}$	29.1h
Comparator	***4***	***2***	***3***	$\mathbf{226 \times 10^{-6}}$	1.5s	23%	$\mathbf{104 \times 10^{-6}}$	0.8s	$\mathbf{292 \times 10^{-6}}$	22.6h

4.5. Summary

This chapter presents an analysis framework for the evaluation of circuit variability and variation-induced error rate (ViER). Variability is quantitatively evaluated as the probability that the output of a circuit, when affected by variations, falls off the noise margin. It is shown that while the variability due to process and voltage variations are small and negligible for the current technology, it is increasingly becoming a factor that will have an impact on a circuit's reliability, especially when the CMOS technology advances into the 16nm or smaller feature sizes. The proposed methodology can be used for evaluating the effects of variability on circuit reliable operations, and predicting the ViER of logic circuits.

CHAPTER 5

Conclusions

The Advance of VLSI circuits and systems into the nanometric regime require accurate and efficient reliability evaluation techniques. In this thesis, we have presented several novel stochastic approaches as a computational framework for the reliability evaluation of logic circuits.

The logic-level approach uses stochastic computational models (SCMs); it accurately evaluates the reliability of a circuit with a precision limited by the inherent randomness of the binary bit streams used in stochastic computation. Compared to accurate analytical approaches found in the technical literature, the proposed SCM approach efficiently handles signal correlations introduced by reconvergent fanouts and thus significantly reduces the computational complexity. Specifically, it has a complexity that increases linearly with the length of the random bit sequences and the number of gates in a circuit.

Compared to the simple simulation of random vectors, the proposed approach has the following distinguishing features: 1) *Versatility*. The SCM is flexible due to its pronounced arithmetic nature. 2) *Generality*. The SCM approach has been

developed as a general computational framework to efficiently implement analytical algorithms. 3) *Scalability*. Compared to Monte Carlo simulation, the SCM approach is scalable as it benefits from the use of a reduced number of pseudo-random numbers. The proposed stochastic approach is therefore potentially useful in the design and test of reliable VLSI circuits and systems. It is also applicable to the computational modeling of complex digital systems; this topic will be pursued for future investigation.

This thesis also presents a transistor-level approach using stochastic transistor models for the evaluation of nanometric CMOS circuits. In the proposed model, a transistor has been modeled as a probabilistic switch such that the probability distribution of its state (ON, OFF and IND) is determined by the input signal probability and its probabilistic behavior is modeled using stochastic computational models (SCMs). As the gate input and the switching error probability of the transistor are represented by random binary bit streams; so if the transistor is affected by a flipping error at a given error rate, the gate input can be considered to be changed by a stochastic XOR. Similar to a logic-level SCM approach, the proposed transistor-level stochastic approach uses stochastic random sequences to represent both signal and error probabilities. However, the traditional SCM approach is static in the sense that circuit reliability is evaluated without considering signal sequences and timing information, thus it may not always be directly applicable to the temporal operation of the transistors and of the sequential elements such as the flip-flops. Therefore the stochastic streams in the proposed new model are defined differently to account for signal sequencing.

At the circuit-level a new evaluation approach has been proposed; the stochastic circuit is first constructed by adding a stochastic gate to each of the transistors in the circuit (for flipping or stuck-ON/OFF errors); then, the initial random bit streams are generated for the signal probabilities of the primary inputs and the error probabilities for the transistors. Propagation of the stochastic streams from the primary inputs to the primary outputs in both the reliable and unreliable

circuits is then accomplished. The XOR and OR gates are then used to decode the joint error probability of the circuit from the obtained stochastic bit streams.

Simulation results have shown the accuracy and efficiency of the proposed approach. It can be seen that while approaches such as Monte Carlo (MC) and gate-level SCM provide an accurate evaluation of circuit reliability, the proposed approach requires a significantly smaller runtime compared to MC simulation. The proposed approach is scalable to the evaluation of large circuits and can be further improved by considering more accurate physical models of the transistor.

As an example, the proposed transistor-level model and approach are incorporated into the proposed statistical methodology for variation-induced error rate (ViER) analysis. Simulation results show that ViER increases quickly and begins to hamper the gate's functionality robustness as technology scales beyond 16nm. The ViER of transistors and gates varies vastly with respect to different technology generations, input vectors, sizing and topology factors. Simulation results also show the ViER of circuits varies under different design techniques, indicating that ViERs should be considered jointly during the design optimization process, and predicting that parameter variation will greatly hamper CMOS circuit functionality robustness in advanced technology.

There are several ways to extend our work. For the logic-level stochastic computational model, we propose to continue improving the scalability of SCM simulation through possible parallelization algorithms. Current SCM approach uses a fixed-length sequence to represent signal and error probabilities, therefore the number of random simulation runs is fixed. We believe that since the error probability is relatively small in most cases, the effective bits that carry information should be selectively simulated to improve the scaling of exact SCM-based computation. The SCM approach can be further explored to develop a method for testing circuit for multiple probabilistic faults. By simulating a circuit using the SCM approach, we should be able to identify input vectors that are

mostly affected by errors in the circuit, and then use a repetition of those input vectors to generate tests that have satisfied fault coverage and detection probability. As the SCM approach is used to analyze the SER of logic designs, electrical masking and timing masking effects can also be incorporated: electrical masking effects can be quantitatively evaluated by SPICE simulation for each type of logic gate, and then an electrical masking factor can be derived for each gate and incorporated into the gate model using stochastic logic. The effects of timing masking can be estimated using static timing analysis (STA) algorithms to determine the error latching windows of gates in a circuit. Through bit-parallel simulation, the fraction of the cycle within the latching window can be characterized.

For the stochastic transistor model (STM), a direct extension would be the reliability evaluation of sequential circuits. Flip-flops are the basic elements in sequential modules. And one may be able to model the internal feedback structure in many ways. One way to deal with it is to characterize the input/output relation of a flip-flop through SPICE simulations. Then a look-up table mapping the input/output of a flip-flop can be incorporated into the STM. The issues here are basically the compatibility and scalability of the approach.

For circuit variability analysis, other variation factors can be considered and extensions to include the effects of leakage currents and switching frequency can be performed in future work. While parameter variations have been analyzed for the reliable function of CMOS transistors, they can also be analyzed for the impact of these variations on SER. Changes in V_t impact SER propagation in several ways. Under process variations, for example, circuit behaviors shift from deterministic to probabilistic, which will affect the error masking effect. These topics will be pursued in future investigation.

BIBLIOGRAPHY

[1] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," in IEEE Micro, vol. 25, no. 6, pp. 10-16, Nov. 2005.

[2] K. P. Parker, and E. J. McCluskey, "Probabilistic Treatment of General Combinational Networks", IEEE Trans. on Computers, vol. C, no. 24, pp. 668–670, June 1975.

[3] I. Bahar, J. L. Mundy, and J. Chen, "A probabilistic-based design methodology for nanoscale computation," in Proc. Int. Conf. Comput.-Aided Des., 2003, pp. 480–486.

[4] International Technology Roadmap for Semiconductors (ITRS) 2009, SIA, http://www.itrs.net/reports.html.

[5] Shekhar Borkar et al., "Parameter Variations and Impact on Circuits and Microarchitecture", Proceedings, DAC 2003.

[6] O. S. Unsal, J. Tschanz, K. A. Bowman, V. De, X. Vera, A. Gonzalez and O. Ergin, Impact of Parameter Variations on Circuits and Microarchitecture, IEEE Micro, 2006.

[7] Kelin Kuhn, et al, "Managing Process Variation in Intel's 45nm CMOS Technology," Intel Tech. Journal, Volume 12, No. 02, June 2008.

[8] W.Wang et al. Compact modeling and simulation of circuit reliability for 65-nm CMOS technology. In IEEE Transactions on Device and Materials Reliability, December 2007.

[9] P. Zarkesh-ha, and A. A. M. Shahi, "Logic Gate Failure Characterization for Nanoelectronic EDA Tools" Proc. IEEE Int. Symp. DFT, Albuquerque, NM, USA, Oct. 2010, pp.16–23.

[10] P.Shivakumar, M.Kistler, S.W.Keckler, D.Burger, and L.Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinatorial Logic," Dependable Systems and Networks, 2002.

[11] Shanbhag NR, Mitra S, de Veciana G, Orshansky M, Marculescu R, Roychowdhury J, Jones D, Rabaey JM. The search for alternative computational paradigms. The special issue on "System IC Design Challenges beyond 32 nm". IEEE Des Test Comput 2008;25(4).

[12] Von Neumann J. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon CE, McCarthy J, editors. Automata studies. Princeton (NJ): Princeton University Press; 1956. p. 43–98.

[13] Siewiorek DP, Swarz RS. Reliable computer systems: design and evaluation. Natick (MA, USA): AK Peters; 1998.

[14] Shukla SK, Bahar RI, editors. Nano, quantum and molecular computing: implications to high level design and validation. Boston: Kluwer Academic Publishers; 2004.

[15] Peter A. Stolk, Frans P. Widdershoven, D. B. M. Klaassen, "Modeling Statistical Dopant Fluctuations in MOS Transistors," IEEE Tran. on Electron Devices, Volume45, No.9, 1998, pp.1960-1971.

[16] A. Asenov, S. Kaya, and A. R. Brown, "Intrinsic parameter fluctuations in decananometer MOSFETs introduced by gate line edge roughness," IEEE Trans Electron Devices, vol. 50, pp. 1254–1260, May 2003.

[17] Samar K. Saha, "Modeling Process Variability in Scaled CMOS Technology," IEEE Design&Test of Computers, Volume27, No.2, pp. 8-15 April 2010.

[18] Gareth Roy, Andrew R. Brown, Fikru Adamu-Lema, Scott Roy, Asen Asenov, "Simulation Study of Individual and Combined Sources of Intrinsic Parameter Fluctuations in Conventional Nano-MOSFETs," IEEE Tran. on Electron Devices, Volume53, No.12, 2006, pp.1960-1971.

[19] Rajesh Garg, Sunil P. Khatri, Analysis and Design of Resilient VLSI Circuits, Springer.

[20] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," ACM Trans. Des. Autom. Electron. Syst., vol. 13, no. 1, pp. 1–35, 2008.

[21] T. Rejimon and S. Bhanja, "Scalable probabilistic computing models using Bayesian networks," in Proc. Int. Midwest Symp. Circuits Syst., 2005, pp. 712–715.

[22] A. Abdollahi, "Probabilistic Decision Diagrams for Exact Probabilistic Analysis," Proc. Int'l Conference on Computer Aided Design, 2007.

[23] N. Mohyuddin, E. Pakbaznia, M. Pedram, "Probabilistic Error Propagation in Logic Circuits Using the Boolean Difference Calculus," in IEEE Intl. Conf. on Comp. Des., Lake Tahoe, CA, USA, pp. 7-13 (2008).

[24] S. Sivaswamy, K. Bazargan, M. Riedel, " Estimation and Optimization of Reliability of Noisy Digital Circuits," in International Symposium on Quality Electronic Design, San Jose, CA, USA, pp. 213-219 (2009).

[25] M. R. Choudhury and K. Mohanram, "Reliability analysis of logic circuits," IEEE TCAD, vol. 28, no. 3, pp. 392–405, March 2009.

[26] J. Han, H. Chen, E. Boykin, J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," Microelectronics Reliability, vol. 51, no. 2, 2011, pp. 468-476.

[27] H. Chen, J. Han, "Stochastic Computational Models for Accurate Reliability Evaluation of Logic Circuits", Proc. Great Lakes Symp. VLSI (GLVLSI), Providence, RI, USA, pp. 61-66 (2010).

[28] W. Ibrahim, V. Beiu, and A. Beg, "GREDA: A Fast and More Accurate CMOS Gates Reliability EDA Tool", IEEE Transactions on Computer-Aided Design of Integrated Circuits and System Accepted on Oct. 24, 2011.

[29] M. Zhang and N. R. Shanbhag, "A soft error rate analysis (SERA) methodology," in Proc. ICCAD, 2004, pp. 111–118.

[30] B. Zhang, W. S. Wang, and M. Orshansky, "FASER: Fast analysis of soft error susceptibility for cell-based designs," in Proc. ISQED, 2006, pp. 755–760.

[31] R. Rao, K. Chopra, D. Blaauw, and D. Sylvester, "An efficient static algorithm for computing the soft error rates of combinational circuits," in Proc. DATE, 2006, pp. 164–169.

[32] N. Miskov-Zivanov and D. Marculescu, "MARS-C: Modeling and reduction of soft errors in combinational circuits," in Proc. DAC, 2006, pp. 767–772.

[33] N. Miskov-Zivanov and D. Marculescu, "Soft error rate analysis for sequential circuits," in Proc. DATE, 2007, pp. 1436–1441.

[34] J. P. Hayes, I. Polian, and B. Becker, "An analysis framework for transient-error tolerance," In VLSI Test Symp., pages 249–255, 2007.

[35] I. Polian, J.P. Hayes, S.M. Reddy and B. Becker, "Modeling and mitigating transient errors in logic circuits," IEEE Trans. on Dependable & Secure Computing, 2010.

[36] C. Yu and J. P. Hayes, "Scalable and Accurate Estimation of Probabilistic Behavior in Sequential Circuits," Proc. VTS, pp. 165-170, 2010.

[37] S. Krishnaswamy, S. M. Plaza, I. L. Markov, and J. P. Hayes, "Signature-based SER analysis and design of logic circuits," IEEE Trans. Comput.-Aided Design Integr. Circuits, vol. 28, no. 1, pp. 74–86, Jan. 2009.

[38] Y-H Kuo, H-I Peng, Wen, C.H.-P., "Accurate statistical soft error rate (SSER) analysis using a quasi-Monte Carlo framework with quality cell models," in Proc. ISQED, 2010, pp. 831–838.

[39] B. R. Gaines, "Stochastic Computing", Spring Joint Computer Conf., 1967, Vol. 30, pp. 149-156.

[40] W. J. Poppelbaump, C. Afuso and J. W. Esch, "Stochastic Computing Elements and Systems", Proceedings of the Fall Joint Computing Conf. 1967, pp. 631-644.

[41] B. Brown and H. Card, "Stochastic neural computation I: Computational elements," IEEE Tran. Computers, vol. 50, pp. 891–905, Sept. 2001.

[42] X. Li, W.K. Qian, M. Riedel, K. Bazargan, D. Lilja, "A Reconfigurable Stochastic Architecture for Highly Reliable Computing," Proc. Great Lakes Symp. VLSI (GLVLSI), Boston, MA, USA, pp. 315-320, 2009

[43] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, "An architecture for fault-tolerant computation with stochastic logic," IEEE Tran. Computers, vol. 60, pp. 93–105, Jan. 2011.

[44] N. Shanbhag, R. Abdallah, R. Kumar, and D. Jones, "Stochastic computation," in Proc. ACM/IEEE DAC 2010, 2010.

[45] W. Ibrahim, V. Beiu, "Reliability of NAND-2 CMOS gates from threshold voltage variations," IIT '09, pp. 135-139.

[46] P. Zarkesh-ha, and A. A. M. Shahi, "Logic Gate Failure Characterization for Nanoelectronic EDA Tools" Proc. IEEE Int. Symp. DFT, Albuquerque, NM, USA, Oct. 2010, pp.16–23.

[47] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," Automata Studies, Shannon C.E. & McCarthy J., eds., Princeton University Press, pp. 43-98, 1956.

[48] J. Han and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," IEEE Trans. on Nanotechnology, vol. 1, no. 4, pp. 201–208, December 2002.

[49] S. Roy and V. Beiu, "Majority multiplexing–economical redundant fault-tolerant design for nano architectures." IEEE Trans. On Nano, 4, 4, 2005.

[50] J. Han, J. Gao, Y. Qi, P. Jonker, J.A.B. Fortes. "Toward Hardware- Redundant, Fault-Tolerant Logic for Nanoelectronics," IEEE Design and Test of Computers, July/August 2005, vol. 22, no. 4, 328-339.

[51] G. Roelke, R. Baldwin, D. Bulutoglu, "Analytical Models for the Performance of von Neumann Multiplexing," IEEE Trans. on Nanotechnology, vol. 6, no. 1, pp. 75–89, 2007**.**

[52] S. S. Tehrani, S. Mannor, and W. J. Gross, "Survey of stochastic computation on factor graphs," in Proc. 37th IEEE Int. Symp. Multiple-Valued Logic, Oslo, Norway, May 2007, pp. 54–59.

[53] C. Winstead, V. C. Gaudet, A. Rapley, and C. B. Schlegel, "Stochastic iterative decoders," In Proc. Intl Symp. Info. Theory, pp. 1116-1120, 2005.

[54] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Muigai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis", Technical Report UCB/ERL M92/41, Electronics Research Lab, Univ. of California, Berkeley, CA 94720, May 1992.

[55] P. Shivakumar et al., "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinatorial Logic," Proc. Int"l Conf. Dependable Systems and Networks, IEEE CS Press, 2002, pp. 389-398.

[56] N. Miskov-Zivanov, D. Marculescu, "Multiple Transient Faults in Combinational and Sequential Circuits: A Systematic Approach," IEEE TCAD, vol. 29, no. 10, pp. 1614–1627, 2010.

[57] M. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering," IEEE Design and Test, vol. 16, no. 3, pp. 72-80, July-Sept. 1999.

[58] S. .Bhunia, S. Mukhopadhyay, and K. Roy, "Process Variations and Process-Tolerant Design," in Int'l Conf. on VLSI Design, pp. 699–704, 2007.

[59] M. Tehranipoor, K. M. Butler, "Power Supply Noise: A Survey on Effects and Research," IEEE Design&Test of Computers, Volume27, No.2, April 2010.

[60] M. Alam, K. Kang, B. Paul, and K. Roy, "Reliability- and process variation aware design of VLSI circuits," in Physical and Failure Analysis of Integrated Circuits, 2007. IPFA 2007. 14th International Symposium on the, july 2007, pp. 17 –25.

[61] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, "High-performance CMOS variability in the 65-nm regime and beyond," IBM J. Res. Devel., vol. 50, no. 4/5, pp. 433–449, 2006.

[62] Y. Cao and L. T. Clark, "Mapping Statistical Process Variations towards Circuit Performance Variability: An Analytical Modeling Approach," in DAC, June 2005, pp. 658–663.

[63] Bernstein, J.B., et al., "Electronic circuit reliability modeling", Microelectronics Reliability 46, 1957–1979 (2006).

[64] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, R. Randa, "A Stochastic Approach to Power Grid Analysis," Proc. DAC, San Diego, CA, USA, pp. 171-176, 2004.

[65] Neil H. E. Weste, D. Harris, CMOS VLSI Design, third edition, Addison Wesley.

[66] Predictive Technology Model (PTM). Available at http://www.eas.asu.edu/~ptm/

[67] H. Chen, J. Han and F. Lombardi, "A Transistor-Level Stochastic Approach for Evaluating the Reliabiltiy of Digital Nanometric CMOS Circuits," in IEEE DFT 2011, Vancouver, BC, Canada, pp. 60-67, 2011.