Winter 2014

Single Electron Devices and Circuit Architectures: Modeling Techniques, Dynamic Characteristics, and Reliability Analysis

Ran Xiao
University of Windsor

Single Electron Devices and Circuit Architectures: Modeling Techniques, Dynamic Characteristics, and Reliability Analysis

By

Ran Xiao

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2013

Single Electron Devices and Circuit Architectures: Modeling Techniques, Dynamic Characteristics, and Reliability Analysis

by

Ran Xiao

APPROVED BY:

_____

Dr. X. Xu
Department of Civil & Environmental Engineering

_____

Dr. H. K. Kwan
Department of Electrical and Computer Engineering

_____

Dr. C. Chen, Advisor
Department of Electrical & Computer Engineering

2013. 12. 9

DECLARATION OF CO-AUTHORSHIP/PREVIOUS PUBLICATION

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:
This thesis also incorporates the outcome of a joint research in collaboration with, and under the supervision of, Professor Chunhong Chen. In Chapters 2 and 3, the key ideas, experimental designs, and data analysis and interpretation were proposed by the author, with the review and revision being provided by Prof. Chen. In Chapter 4, the key ideas and algorithms were presented by Prof. Chen in consultation with the author, and the experimental setup and analysis were completed by the author. Prof. Chen also provided the final review for Chapter 4.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes one original papers that have been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter 2: Section 2.2 and 2.3.1-2.3.4	Single-Electron Tunneling Based Turnstile: Modeling and Applications, in Proceedings of 2013 IEEE International Conference on Nanotechnology (IEEE-Nano'13), Beijing, China, August 2013.	in press

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.
I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The Single Electron (SE) technology is an important approach to enabling further feature size reduction and circuit performance improvement. However, new methods are required for device modeling, circuit behavior description, and reliability analysis with this technology due to its unique operation mechanism. In this thesis, a new macro-model of SE turnstile is developed to describe its physical characteristics for large-scale circuit simulation and design. Based on this model, several novel circuit architectures are proposed and implemented to further demonstrate the advantages of SE technique. The dynamic behavior of SE circuits, which is different from their CMOS counterpart, is also investigated using a statistical method. With the unreliable feature of SE devices in mind, a fast and recursive algorithm is developed to evaluate the reliability of SE logic circuits in a more efficient and effective manner.

DEDICATION

To my grandparents, parents, aunty Dan and Yan.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Chunhong Chen, for his support and encouragement. I have constantly benefited from discussions with him during my study and research work. I appreciate his creativeness and innovation within several papers we have co-authored, which cover almost all topics presented in this thesis. I would also like to thank Dr. H. K. Kwan and Dr. X. Xu for their valuable comments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS/SYMBOLS

Abbreviations:

Abbrev.	Description
ADC/DAC	Analog-Digital/Digital-Analog Converter
BDD	Binary Decision Diagram
BDEC	Boolean Difference-based Error Calculator
CCM	Correlation Coefficient Method
CDF	Cumulative Distribution Function
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Cellular Neural Network
ER	Equivalent Reliability
FD	Frequency Divider
FSB	Frequency Synthesizer Block
FSCL	Frequency Sensitive Competitive Learning
MC	Monte-Carlo simulation
ME	Master Equation
MOSFET	Metal-Oxide-Semiconductor Field-Effect-Transistor
MV	Multiple-Valued
NDR	Negative Differential Conductance
PDF	Probability Distribution Function
PFD	Phase-Frequency Detector
PGM	Probabilistic Gate Models
PTM	Probability Transfer Matrices
SE	Single Electron
SED	Single Electron Device
SET	Single Electron Tunneling
SN	Storage Node
VCO	Voltage-Control Oscillator

Symbols:

Constants	Definition	values
e	Elementary charge	1.6×10^{-19} C
k_B	Boltzmann's constant	1.38×10^{-23} J/K

Chapter 1	
C	Capacitance of the SE island
E_c	Charging energy
T	Temperature

Chapter 2	
C_{Σ}	Total capacitance on the SE island
C_{ext}	Equivalent capacitance of the reminder of the circuit viewed from the tunnel junction
C_G	Gate capacitance of SET turnstiles
C_{SN}	Capacitance of SN in SET turnstiles
C_T	Capacitance of tunnel junctions
d	Euclidean distance
K	Gain factor of the conscience mechanism
N_W	Number of wins in the current neuron
$P_{thermal}(t)$	Probability that a thermal error happens before time t
q_i	Charges at the nodes in SET turnstiles
R_T	Resistance of tunnel junctions
V_C	Threshold voltage of electron tunnel
V_{ij}	Voltage across the tunnel junction
V_{SN}	Potential at SN in SET turnstiles

ΔE	Electrostatic energy difference before and after the tunnel event
Γ	Tunnel rate

Chapter 3	
C_{load}	Load capacitance at output node
Gaussian (μ, σ^2)	Gaussian distribution with the expected value μ and variance σ^2
I_{DS}	Current went through the SET transistor
N	Total number of states during the logic transition
N_e	Total number of tunnel events that need to be considered within each tunnel junction
N_{tot}	Total electron tunnels happened within each tunnel junction
$P_{error,\,N}(T_N)$	Error probability at time T_N
$P_i(t)$	Time-dependent occupation probability of state i
$T_{accurate}$	Delay results from the exact solution of ME
$T_{approximate}$	Delay results from the statistical model
T_{avg}	Sample average delay of MC simulation
T_{eq}	Delay calculated by step estimation method
t_i	Arrival time of state i
T_{MIB}	Delay calculated by MIB macro-model
V_{sw}	Output voltage swing
Δt_i	Time interval between state i and state $i-1$
$\Phi^{-1}(\cdot)$	Inverse function of the CDF of a Gaussian variable
Γ_i	Tunnel rate from state $i-1$ to state i

Chapter 4	
$\{r^0, r^1\}$	Reliability pair of a signal, depending on the error-free value ("0" or "1") of the signal

$\{r_{eq}^{0}, r_{eq}^{1}\}$	Equivalent reliability pair for certain signal
$\{r_{g}^{0}, r_{g}^{1}\}$	Reliability pair of a gate, depending on the error-free value ("0" or "1") of its output signal
F_j	Primary output of the logic circuit
K_{ab}^{*}	Probability that both signal a and b are "1"
$\mathbf{M}$	Probability transfer matrix
$\mathbf{M}'_{\text{update}}$	Probability transfer matrix for gate reliability pair
$\mathbf{M}_{\text{cor}}$	Probability transfer matrix for correlated case
$\mathbf{M}_{\text{ind}}$	Probability transfer matrix for independent case
$\mathbf{P}$	Actual input probability vector
$\mathbf{P}^{*}$	Error-free input probability vector
P_s	Probability of the signal *s* being logic "1"
$\mathbf{R}$	Output reliability vector
$r_{a'b'}^{(i,j)}$	Probability that both *a'* and *b'* are reliable
s	A specific signal in the combination circuit
s^{*}	Error-free version of signal *s*
ε_i	Error probability of gate *i*
$\theta^{(i,j)}$	Correlation coefficient between two signals a^{*} and b^{*}

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Motivation

It is widely known that the tremendous growth in modern semiconductor industry has been relied on the continuous shrinking of electronic device dimension over decades. In microelectronic industry, making things smaller is the most valuable point, since smaller feature size implies higher computation power per unit area as well as lower cost. However, as the size of traditional MOSFETs has been scaling down to nanometer, circuit designers are facing new challenges that were not much critical in the past, for instance, short channel effects, physical limitations imposed by thermal fluctuations, stochastic nature of quantum effects, large power dissipation due to leakage and ultrahigh integration densities, and the technological limitation in manufacturing methods, etc. There have been reports suggesting that the MOS transistor itself cannot be shrunk beyond certain limits dictated by its operating principle [127]; and according to the International Technology Roadmap for Semiconductors [1], such rapidly improvement will eventually come to end in the near future. Therefore, alternative technologies are desired in order to achieve further scaling and performance improvement. Some novel devices based on new operation concept and technologies with great scaling potential have been fabricated and under investigation such as Single Electron Devices (SEDs), resonant tunneling devices (RTDs), quantum cellular automata (QCA), quantum dots, molecular electronic devices, and carbon nanotubes or nanowires.

As one of those emerging technologies, SEDs are based on the controllable transfer of single electrons (SEs) between small conducting electrodes and have had already several demonstrative scientific experiments as well as enabled fabrication methods. Since electrons were discovered in 1897, people have been trying to build so-called “ultimate circuits” that are able to treat one electron as one bit information. It is considered as an ultimate target for electronic engineers and even for the whole information industry, as electrons are the smallest particles that can be manipulated or controlled with the existing technical conditions. The applicability of single electronic is wide. For instance, in the field of microelectronic, SEDs are considered as one the most promising candidate of next generation nanometer devices beyond MOS transistors. The most predictable and lucrative application is the SE memories, which could store one bit of information with one electron theoretically; and several companies have single-electron memory products in their roadmap for a planned release in the 2010 to 2015 time frame [7]. I also would

like to anticipate that numerous data could be stored in few square nanometers using SE memories; and the processors using SEs will have extremely high density and computational capability. Over the decades, engineers and also physicists and chemists were driven to dedicate themselves to realize these expectations.

1.2 A Brief Introduction of Single Electronics

As the structure size keep scaling into few nanometers, the granularity of charge in terms of the finite number of charge or electrons need to be taken into consideration. The core of single electron manipulation is the phenomenon called Coulomb Blockade. Its basic concept is illustrated in Fig. 1.1 as explained by Likharev in [3], where a conduction island is initially electro neutral before an extra electron being injected from outside; then this uncompensated single electron charge will create an electric field which repulses the addition of following electrons.

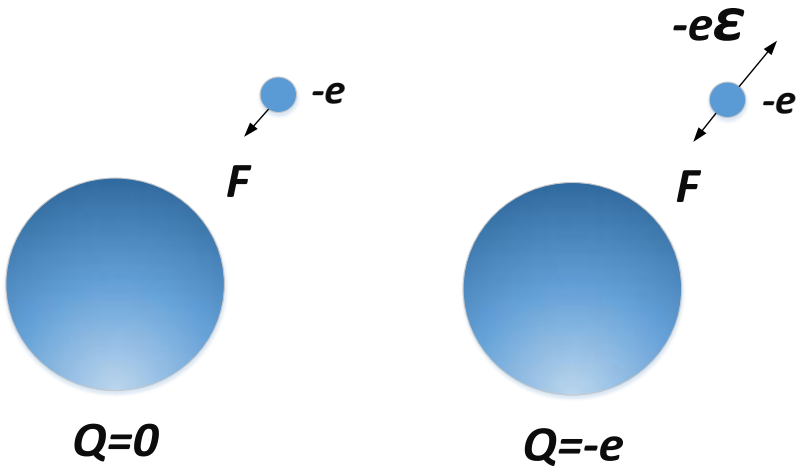


Figure 1.1. The basic concept of single electron control (after [3]).

It should be noted that the square of the island size has to be small enough to generate strong electric field that can ensure the SE control. For example, the field is ~140kV/cm on the surface of a 10-nm sphere in vacuum [3]. More adequate description of this phenomenon is using the charging energy E_c, as:

$$E_c = e^2 / C \tag{1.1}$$

where $e \approx 1.6 \times 10^{-19}$ Coulomb is the elementary charge and C is the capacitance of the island. If this charging energy is large enough, especially compared to thermal energy k_BT, then the single electron control could be realized; and SE effects could be detected as well.

In order to avoid the disturbance from thermal energy, the Coulomb blockade phenomenon is commonly investigated and studied either under very low temperature (e.g., ~10K for 100nm-scale devices) or in extremely small structures (smaller than ~1nm for room temperature operation). This harsh condition is partly the reason for the realization of SE manipulation took until the late 1980s although its concept has been proposed at very beginning of last century. However, the rapid growth in nanofabrication techniques recently has made possible the solid state science field as well as single electronics.

1.3 Advantages and Current State of SE Applications

The advantages of single electronics are obvious, such as low power consumption and inherent small feature size. Unlike the traditional CMOS where power consumption is mainly due to the dynamic capacitors charging and discharging and static leakage, SE technique transports electrons by utilization of electron tunneling phenomenon, which refers to effects manifest in the injection and extraction of individual electrons from nano-scale structures, and the corresponding change in energy of system [2]. As the results, logic switching in SE digital circuits could be completed by only single or several electrons transporting from one electrode to another; and therefore the energy it consumed is extremely low compare to its CMOS counterpart. It is understood that in CMOS technology, to conduct a current between the source and the drain, a voltage in the hundred mV has to be applied to obtain an inversion layer to let the current go through the transistor; while for SEDs, a supply voltage of few millivolts is enough. For example, estimated switching energy of single SE logic element is in the order of tens of meV (or, ~10^{-21}J) [68], comparing with fJ~pJ (depend on the load capacitances) for a typical CMOS gate; and for memory applications, the reported standby power of a hybrid SED/MOS static memory cell is few nW [39]; for embedded system applications, the simulation results in the literature have shown that an AVR microcontroller implemented by hybrid SED/MOS architecture consumes approximately 200uW running at 4MHz, permitting it to run round 20 years on one AA battery [99].

On the other hand, the nano-scale feature size of SEDs means they will get high integration density and therefore high computation power for processors, as well as the high storage density for memory applications. For example, the report from NTT research group showed a multi-valued hybrid memory cell could be made within an area of ~10^{-4}um^2, which is much smaller than

that of CMOS implementation [6]. In electronic industry these advantages gives possibilities to more portable and energy-saving products (e.g., cellphones and tablet computers).

SE technology has also been of much interest for a variety of metrology applications such as current standards or single-photon sources, thanks to its capability of one by one charge transfer. It could provide a quantized current depend on the frequency of the electron transfer. This current quantization has been experimentally demonstrated in Si-wire charge-couple devices [25]. More applications in terms of current measurement can be found in [35-38].

TABLE 1.1. Circuit architectures and applications using SEDs

SE devices	**Applications**
SE Transistor	Logic Gates [70], CNN [92], BDD Logic [80], Embedded system [99]
SE Turnstile	Memory [88], Phase Detector [91], Current Meter [36]
SE Threshold Gate	Flip-flops [68], Majority Gates [71, 100]
SE Trap	Memory [43, 44]
Hybrid CMOS-SED	ADC/DAC [89], Multipliers [85-87], VCO [90], Binary adder [82], Multiple-valued (MV) Logic [94-95]

Furthermore, SEDs have many novel functionalities such as Coulomb Oscillation coming from the electron tunneling effect involved. Due to this, SEDs have the ability to efficiently implement some functions that are inefficient using CMOS logic, threshold logic, or nonlinear functions [99]. Many novel circuit architectures have been proposed and studied in order to exploit such special properties. For some applications where the circuits are complex using CMOS, they may become very concise when SEDs take over, such as multiplexers [99], multipliers [87], or analog neural networks [101]. Other examples are the SE static memory cell using its negative differential conductance (NDR) character [44], or ADC/DACs [89] using its Coulomb oscillation behavior.

Some existing applications using SEDs are listed in Table 1.1, where we can see for each kind of SEDs there are several applications, including both analog and digital ones. A more detail review of SE technology can be found in Section 2.1.

1.4 Challenges

Although it has promising future, several challenges still remain for large-scale integration of SEDs and repeatable fabrication method for small feature size products that can work under room temperature. Certain techniques are also needed for hybrid CMOS and SEDs compatible integration. SE circuits that operate reliably at room temperature require the critical feature size less than 10nm, which is also still challenging for repeatable fabrication using existing lithographic techniques.

On the other hand, SEDs are usually suffered from so-called random background charge effects which can easily destroy the desired device functionality, especially for memory and digital applications. This is due to the charged impurities located close to the SE island generated during the manufacturing process. In addition, the SE effects can be easily influenced by thermal energy, which means its performance would be temperature depended. In order to avoid the big performance variation due to temperature fluctuation, certain temperature compensation mechanism is need (e.g., feedback structures as in [83]); and in terms of SE circuits design, the robustness against random background charge as well as temperature fluctuation is prerequisite. The new circuit architectures that have these kinds of features are highly desired for new generation of SE circuits design.

As a summary, although the SE technology holds great promise, the practical applications using SEDs would require further research, especially for the fault tolerant techniques, temperature independent architecture design, and novel information processing architecture which could take advantages of those SE new functionalities.

1.5 Contributions

In this study, the modeling technique and simulation method have been comprehensively investigated. A new analytical compact model for one of SEDs, the SE turnstile, has been proposed. The model is able to accurately describe the tunneling events involved, which is verified by Monte Carlo simulation with good agreement. Based on the proposed turnstile, hybrid SED/MOS circuit co-simulations are performed in Spectre simulator on Cadence platform by implementing the proposed turnstile model in Verilog-A language. By taking advantages of the frequency characters of SE turnstile, I design several application circuits that could manipulate signal information in frequency domain, which include the new digital frequency divider, the frequency-voltage level converter, a phase-frequency detector as well as a conscience mechanism circuits in competitive learning network. In addition, we emphasized the applications of SE

techniques in neural networks by discussing how their drawbacks can be masked in large-scale parallel computing. A comparative study has been presented to show the efficiency of those new SE circuit architectures. (Chapter 2)

In order to accurately analyze the switching delay of SE circuit, a statistical method has been proposed. The switching process within SE logic gates is comprehensively studied; and its dynamic feature is analyzed theoretically. The stochastic Markov characteristics in electron tunneling-based systems are investigated, which deduced the proposed statistical delay model. The effective accurate delay estimation using proposed model is verified by both theoretical analysis and extensive simulation results. It has been shown that the proposed method has overcome those drawbacks of existing approaches, which have also been analyzed and evaluated for comparison study. (Chapter 3)

A novel reliability analysis method, based on concept of equivalent reliability (ER), is proposed and implemented to evaluate the reliability performance of combinational circuits consist of unreliable logic elements. The proposed method provides better results than the state of the art methods in terms of either efficiency or accuracy. Simulation results on several small circuits and also large scale benchmark circuits show that out approach obtains a significant speedup over simulation-based methods and more accurate results than other existing analytical approaches. Based on this new ER model the error propagation in combinational circuits is studied; and also discussed are its potential applications, including reliability improvement and fault-tolerant design. (Chapter 4)

CHAPTER 2

MODELING TECHNIQUE FOR SIGNLE ELECTRON DEVICES

2.1 Background

In this section, I give some background knowledge regarding to SED such as its basic components, short history and current frication methods, as well as existing simulation methods. I start with a brief introduction of the SEDs, followed by a quick review of the history for SEDs and some recent experimental studies and fabrication method; then I move to an introduction of the existing modeling and simulation methods for SEDs.

2.1.1 Single electron devices

In order to take advantages of those single electron effects such as the Coulomb blockade, electron tunneling, and Coulomb staircase and oscillations, many nanoscale systems have been proposed and studied in the literature. The basic building block of SEDs and circuits is the tunnel junction, whose circuit symbol is shown in Fig. 2.1, as below.

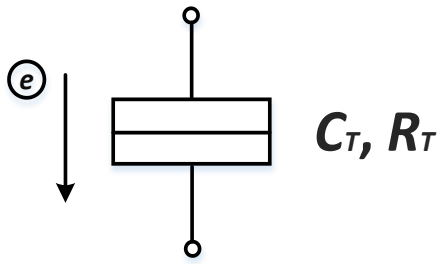


Figure 2.1. Circuit symbol of a single electron tunnel junction with capacitance C_T and resistance R_T; an electron can tunnel through the junction if certain tunnel criteria (which will be discussed in detail in Section 2.2.1) is satisfied.

The tunnel junction is usually recognized as an energy barrier between two electrodes, whose height and width corresponding to its capacitance C_T and tunnel resistance R_T, which are the phenomenological quantities based on the orthodox theory. It should be noted that the tunnel junction is a representative model for a board range of permeable nanoscale structures fabricated by different methods on different materials. The electron can tunnel through the junction one by one if the capacitances of the junction as well as capacitances at two terminals are small enough,

In this Chapter, Section 2.2 and 2.3.1-2.3.4 are the outcome of a joint research under the supervision of professor Chunhong Chen.

which means the voltage across the junction can be changed significantly due to few electron tunnels. After an electron tunnel happened, the voltage across the junction would decrease then suppressing the following tunnel events. This is the basic principle how the electron manipulation can be realized. Such phenomenon is known as so-called Single Electron Tunneling (SET); and those devices and circuits that utilize SET effects are usually recognized as SET-based devices and circuits.

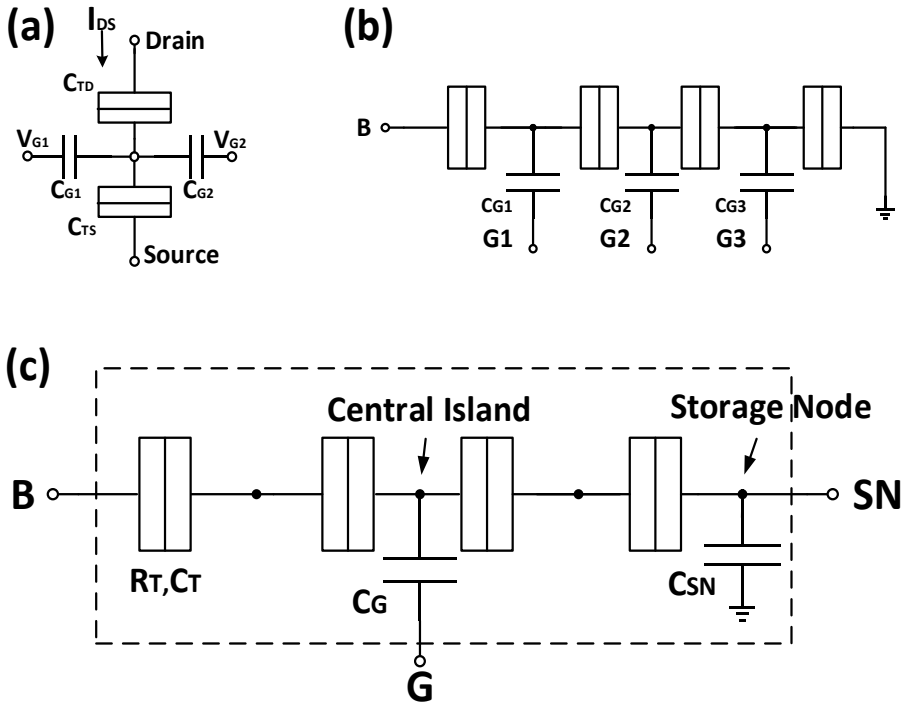


Figure 2.2. Circuit schematic of (a) SET transistor, (b) SET charge pump and (c) SET turnstile.

Those SEDs composed by tunnel junctions are SET transistors, SET pumps and SET turnstiles, as shown in Fig. 2.2(a-c), respectively; where those are not shown including resonant tunneling devices (RTDs) and quantum dots (QDs). The most popular one among them might be SET transistors consists of two tunnel junctions separated by a nanoscale island to ensure the occurrence of Coulomb blockade. It has three terminals (viz., drain, source and gate terminals), as the traditional CMOS transistor; however, its Coulomb oscillation character is unique and useful for circuit designers.

The SET turnstile is yet another nanometer device that offers a promising way to develop ultimate circuits. The first SET turnstile was demonstrated by a French-Dutch corporation in 1990

[29] and, since then, various modeling and fabrication techniques have been reported, including the multilevel memory cell in 2004 [40] and a room temperature data processing turnstile circuit in 2006 [19]. In the rest of this Section, the SET turnstile is also the main subject of our research work. For more detail review of SEDs I refer our reader to [4].

2.1.2 Experimental studies and fabrication methods

In 1920s, Robert Milikan showed his famous oil drop experiment which gives possibility to localize single electrons on oil drop and measure their elementary charge. In 1951, C. Gorter explained correctly a current suppression at low bias voltage as the cause of Coulomb repulsion [10]. The electron manipulation of electron in solid material is realized until 1980s. In 1985 D. Averin and K. Likharev [129] formulated so-called "Orthodox theory" of single electron tunneling, which quantitatively describes important charging effects such as the Coulomb blockade and single-electron tunneling oscillation [10]; and the first SET transistor was built by Fulton and Dolan [128] in 1987. The standard repeatable procedure to build single-electron device with a double angle evaporation method in the Al/AlO2 material system was also developed by Dolan [130]. This technique and its variations are still today the most prevalent ones to manufacture SEDs in metallic material systems [10]. After that, several SEDs and their simple application circuits with variety of structures or architectures have been proposed and fabricated based on different procedures and materials [17-38], thanks to the growing studies of production techniques for small structures.

The recent progress in SEDs manufacturing using electron beam lithography (EBL) is reported in 2009 [18]. The device dimension of less than 10nm needed for their room temperature operation is achieved in this work. In 2011, the silicon SET transistor fabrication method based on optical lithography is proposed by Sun [17] in Singapore. The fabricated SET transistors exhibit Coulomb oscillation at room temperature due to their extreme small feature size of 4nm. The optical lithography approach also looks like promising since it is compatible with the conventional CMOS fabrication process.

2.1.3 Current modeling and simulation methods

The growing studies on single electronics have been made the revolution of microelectronic industry possible. Since that, the SEDs are more and more involved in the new generation circuit architecture design. Therefore, the modeling techniques and simulation frameworks of these SET-based devices are increasingly desired. In terms of SEDs modeling, recently the research on this area has been very active. To describe the behavior of single electron circuits, three main

approaches are currently used for simulation: Monte Carlo (MC) method, Master Equation (ME), and macro-modeling method. Here, I give a quick review of these three methods; some detail discussion will also be presented in next Chapter Section 3.4.

The MC method (or, more accurately, kinetic Monte Carlo method) is the most popular technique. MOSES by Chen et al. [22] and SIMON by Wasshuber [52] are two commonly-used simulators based on MC method. This approach simulates the electron tunneling events in order to directly emulate the actual behavior of electrons in SET circuits. In transient simulation, it is assumed that tunneling events occur instantaneously as a competing Poisson process. In stationary simulation, a transient simulation is run for a certain number of events per time step. State probabilities, voltages, and steady currents are calculated as averages over these tunnel events. The main drawback of MC simulation is that it usually needs large number of simulation trials to converge into stable and accurate results. Since that, it is only suitable for small scale SE circuits simulation. Some other simulators based on this method are [57-60].

The ME is the direct description for the underlying Markov process of electron tunneling; and it is the exact mathematical model for the electron tunneling events involved, which is able to describe both SEDs static/dynamic characteristics comprehensively. The pioneering simulators applying this method include SENECA by Fonseca et al. [54] and SETTRAN by Korotkov [3]. The ME simulators are also suffered from the unrealistic long simulation time for even medium size SE circuits.

On the other hand, the macro-modeling method looks at the macro-characteristics of electron tunneling, which enable SET circuit simulation using CAD tools (e.g., SPICE) with significant increase in efficiency for large circuits. For the past few years, this approach is increasingly employed when simulating SET circuits. While different SEDs need their specific model (e.g., single-electron transistors [47-51, 61-64] and turnstiles [53, 91-94]), the macro-modeling method is very efficient, thanks to the powerful CAD tools. Some existing models can also deal with certain secondary effects such as background charge and temperature effect. In summary, each of the above three methods has its own advantages and limitations for static and dynamic characteristics of SEDs simulations, which inspires us to analyze and evaluate them based on different performance requirements under different cases.

2.2 Method

In this section, I present a compact model for SET based turnstiles. This macro model is based on behavior description using Verilog language, making it different with those existing turnstile

models (most of them are based on SPICE). This proposed model accurately captures the tunnel process within the SET turnstile. The device characteristics produced are verified by MC simulation. The hybrid SET/MOS con-simulations are successfully performed using Spectre simulator in Cadence platform. In addition, based on the unique functionalities offered by SET turnstile, some application circuits are designed; and their advantages compared to CMOS implementations are showed by extensive simulation results.

2.2.1 Working principle of SE turnstiles

SET-based turnstile is a promising device that can accurately transfer single electrons at high frequency and low power dissipation. Its schematic is shown in Fig. 2.3(a), which consists of four single-electron tunnel junctions with a bias terminal (B), gate terminal (G), storage node (SN), and two coupling capacitances C_G and C_{SN}.

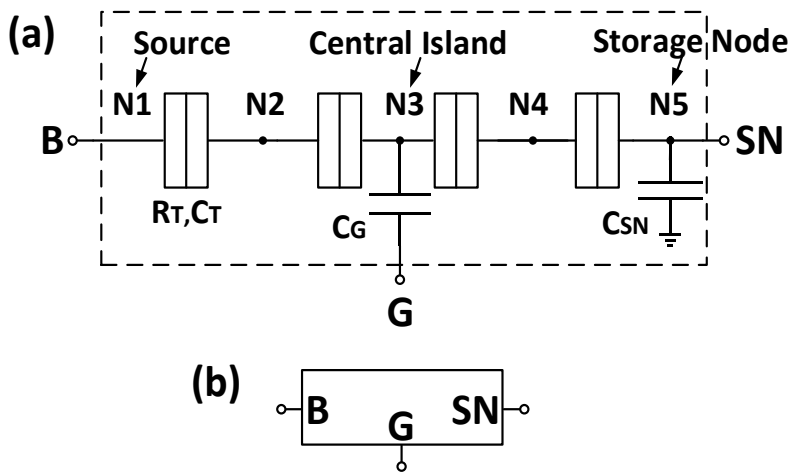


Figure 2.3. (a) Circuit schematic of a SET turnstile, and (b) its circuit symbol.

The operation principle of the SET turnstile is described briefly as follows. With a proper value of the bias voltage V_B, one or several electrons are pulled into the central island (i.e., node N_3) from the source node N_1 (or the storage node SN) by increasing the gate voltage V_G beyond a certain threshold, and then pushed out to the SN (or N_1) by decreasing V_G. If V_G is a periodic signal, a number of electrons are transferred between the source and the SN over each period. The direction of electron transfer and the number of electrons transferred during each period depend on the V_B and V_G. The circuit symbol for the SET turnstile is shown in Fig. 2.3(b).

To model the above turnstile, we assume: 1) the "orthodox theory" of single electron tunneling is use, 2) the interconnect capacitances are much larger than device capacitances. This ensures that the device characteristics only depend upon the node voltages, and 3) the initial charges on all islands in the turnstile are zero, and all tunnel junctions have capacitance C_T and resistance R_T. From the turnstile architecture, the charge on node i can be expressed as:

$$q_i = \sum_{j=1}^{N} C_{ij} V_j \tag{2.1}$$

where C_{ij} denotes the elements of the capacitance matrix. The voltages and charges on all nodes (islands) can be derived by solving a set of equations from (2.1) with i = 1, 2, … 5.

The orthodox theory explains how and when the electron will tunnel through the junction, which is actually due to quantum effect. However, the orthodox theory gives good agreements with experimental results by using some classical mechanic concepts, certainly by holding some assumptions and approximations (this is also where its name comes from). In this theory, tunneling events are modeled as stochastic events, with a certain tunnel rate Γ (probability per unit time), which depends solely on the reduction of the free energy ΔE of the system as a result of this tunneling event as given by:

$$\Gamma = \frac{\Delta E}{e^2 R_T [1 - \exp(-\Delta E / k_B T)]} \tag{2.2}$$

where k_BT is the thermal energy, e is the charge of an electron, and ΔE is the electrostatic energy difference before and after the tunnel event, and is given by $\Delta E = -e(|V_{ij}| - V_C)$ [68], where V_{ij} is the voltage across the tunnel junction and V_C is the threshold voltage of the junction.

Note that $V_{12} = V_{23}$ and $V_{34} = V_{45}$ due to the symmetrical structure of Fig. 2.3(a). The above threshold voltage for an electron to tunnel through a junction is given by:

$$V_C = \frac{e}{2(C_{ext} + C_T)} \tag{2.3}$$

where C_{ext} is the equivalent capacitance of the reminder of the circuit viewed from the tunnel junction. When $|V_{ij}| > V_c$, an electron will tunnel (even at very low temperature) from node j to i through the junction. This tunnel event will change q_i and q_j, as well as related node voltages.

The operation of a SET-based turnstile can be viewed as a cycle of electron transfers. When V_G increases from zero to a certain positive value (assume a positive value of V_B), an electron tunnels from SN to N_3 though two tunnel junctions (in practice the two tunnel events happen one

after another). When V_G returns to a smaller value, the electron starts tunneling from N_3 to N_1. The transfer cycle is completed as V_G returns to zero. As a result, the voltage on the storage node (i.e., V_{SN}) changes due to the electron charges accumulated at the SN. Here, the V_{SN} is calculated as:

$$V_{SN} = [V_B + \frac{2C_G}{C_T}V_G + \frac{4(C_T + C_G)}{C_T^{\,2}}q_5 + \frac{2}{C_T}q_3] \cdot \frac{C_T^{\,2}}{C_T^{\,2} + 2C_TC_G + 4C_{SN}(C_T + C_G)} \quad (2.4)$$

where q_3 and q_5 (i.e., q_{SN}) are charges on the central island and storage node, respectively.

2.2.2 Verilog-A model and its simulation results

The aforementioned characteristic was described using Verilog-A modeling language, and simulated under Cadence environment in comparison with SIMON simulator [52]. The parameters used for simulation are: C_T = 1aF, C_G = 0.5aF, C_{SN} = 10aF, R_T = 1MΩ, V_B = 0.1V, and V_G is chosen to be a triangular pulse (it can also be a rectangular pulse without affecting the tunneling events, as will be seen later in the paper).

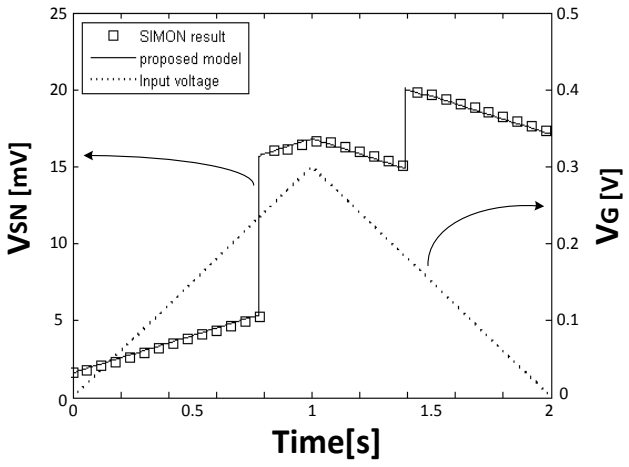


Figure 2.4. Voltage characteristics of the proposed turnstile model at T = 0K.

The results are shown in Fig. 2.4, which reveals the accuracy of our model in predicting the characteristic of the turnstile. I also used a SET transistor which is electrically coupled to the SN of the turnstile (as shown in Fig. 2.5) to detect its potential change (as an electrometer). The parameters of the turnstile are same as Fig. 2.4. The input voltage V_{in} is a pulse signal with 300mV amplitude, 2s period and 50% duty cycle. The parameters of SET transistor are: $C_D = C_S$ =

1aF, C_G =0.1aF, C_{G2} = 2aF, V_{DS} = 0.02V and the phase-control voltage V_{PC}= 0.04V. The simulation result is shown in Fig. 2.6. It can be seen that the SET transistor can sense the presence of single electrons stored in SN. Furthermore, a SET transistor/MOS hybrid circuit can be used to measure and amplify the voltage level of the SN, which corresponds to the number of electrons stored in the SN. In this work, the MIB analytical model is used for SET transistors [47].

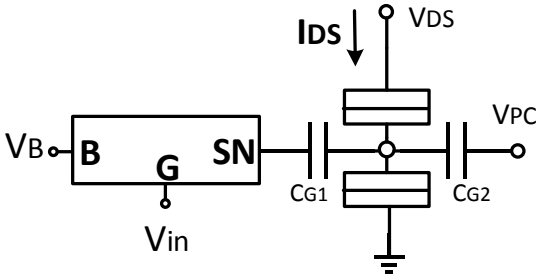


Figure 2.5. Schematic of a single-electron turnstile followed by a SET transistor (used as an electrometer).

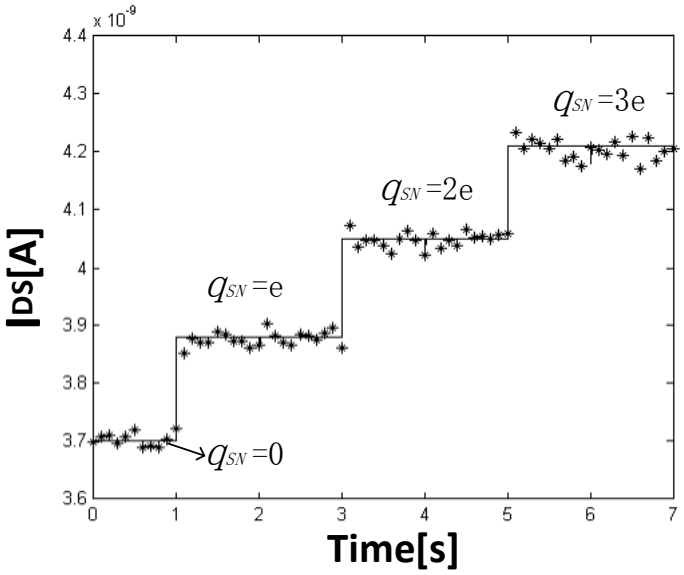


Figure 2.6. Simulation result for I_{DS} characteristics (T=5K) of Fig. 2.5 using SIMON (symbols) and proposed model (solid line), where the gate voltage of the turnstile is a pulse signal with the period of 2s.

It should be noted that the MC simulation is usually time consuming when it involves current-biased SET transistors, high temperature and/or high bias voltages for which the number of states in the circuit during simulation increases. However, when using a Verilog-A model, it is more

efficient to get the same results. One of the key features of Verilog-A model is that, combined with *Spectre*, it provides circuit designers with a comprehensive environment for easily developing proprietary models for specific device behaviors. On the other hand, although the macro-modeling approach can be used alternatively in a SPICE-compatible environment, it provides little physical insight into the operation of devices and may not be easily scalable due to the fact that it is purely empirical in nature [11].

2.2.3 Temperature effects

SET-based devices always suffer from the error tunneling caused by the thermal energy. If T > 0K, the thermal tunnel events take place even if $\Delta E > 0$ (i.e., $|V_{ij}| < V_c$). This is the so-called thermal error. In Monte-Carlo (MC) simulation [52], the error probability is given by the following expression:

$$P_{thermal}(t) = 1 - \exp(-\Gamma t) \tag{2.5}$$

where $P_{thermal}$ denotes the probability that a thermal error happens before time t, Γ is the thermal tunnel rate which can be calculated from (2.2).

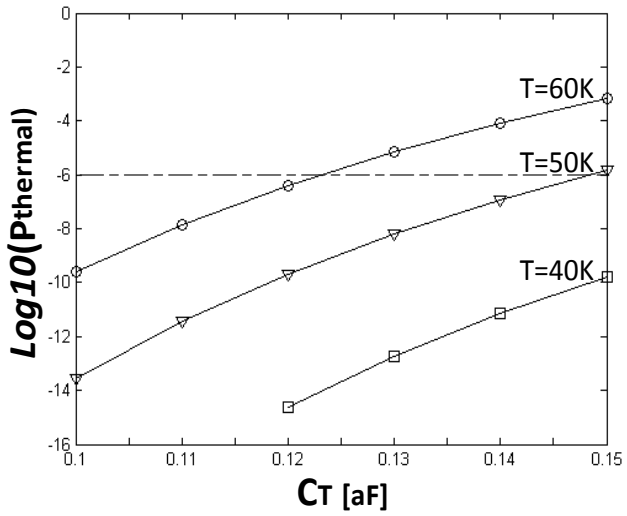


Figure 2.7. The error probability $P_{thermal}$ as a function of junction capacitance C_T and temperature T.

To explore how the temperature T affects the turnstile circuit, I calculated the $P_{thermal}$ using different parameters under different temperature, as shown in Fig. 2.7. We can see that as the T

increases, so does the $P_{thermal}$. The junction capacitance C_T is also a key factor. For instance, at T = 50K, $P_{thermal}$ is about 1e−14 with C_T = 0.1aF, and increases to 1e−6 with C_T = 0.15aF. In our model, for each possible tunneling event, we calculate the tunnel rate using (2.2). By considering the thermal tunnel events as an independent and exponentially distributed process [52], the time interval between two thermal tunnel events is given by: $\Delta t = -\ln(r)/\Gamma$, where r is an evenly distributed random number within the interval [0, 1]. At each iteration, we compare the duration Δt of all tunnel junctions, and take the shortest one (i.e. a thermal tunneling event happens at this junction).

2.3 Application circuits

Circuit-level SET turnstile/MOS co-simulations are successfully performed by implementing the proposed Verilog-A turnstile model in Cadence environment. Analog co-simulations are also carried out for different application circuits, such as digital frequency divider, frequency-voltage level converter and phase detector. Furthermore, focus on the application of self-learning network, I analyzed the impact of using SEDs in its architecture design and discussed the performance of proposed circuit architecture in terms of area and power consumption.

2.3.1 Reset module

As mentioned before, with a periodic pulse signal of V_G in the SET turnstile, one can keep transferring electrons to or from the SN. The charges that are accumulated at the SN increase until no more tunneling is possible, depending on the value of V_B and V_G. For this reason, to ensure the turnstile can work properly in practice, one usually has to clear the charges at the SN regularly, which means that a reset module is needed.

To realize the reset operation, a multiplexer can be used with a feedback structure which controls the select signal. The feedback signal is usually provided by other parts of the circuit (refer to Fig. 2.8). If its voltage is larger than a certain threshold value, the multiplexer will switch from V_B to a specific voltage V_{reset} to realize the reset operation. The value of V_{reset} should be selected in such a way that the following condition is met:

$$\begin{aligned} &if \quad q_{SN} = \pm e : |V_J| > V_C \\ &if \quad q_{SN} = 0 : |V_J| < V_C \end{aligned} \tag{2.6}$$

where V_J is the maximum junction voltage within the turnstile. This implies that electrons can keep tunneling to or from SN until there is no pure charge present on the node. For instance, with

the same circuit parameters as before, let us assume $q_{SN} = -e$ (i.e. an electron stored in SN) in the first equation above. To pull this electron out from SN, it requires:

$$|V_J| = \frac{C_T C_G + C_{SN}(C_T + 2C_G)}{C_T^{\,2} + 2C_T C_G + 4C_{SN}(C_T + C_G)} \cdot [V_{reset} - \frac{4(C_T + C_G)}{C_T^{\,2}} e] + \frac{C_T + 2C_G}{C_T^{\,2}} e > V_C \quad (2.7)$$

i.e., $V_J = (0.33V_{reset} + 0.016e/C_T) > V_C$, or $V_{reset} > 154.5\text{mV}$. After clearing the charges in SN, the second equation in (2.6) must be satisfied to prevent any tunnel event within the turnstile, which means $V_{reset} < 161.9\text{mV}$. Generally we can select the V_{reset} to be (154.5mV+161.9mV)/2 ≈ 158mV.

2.3.2 Digital frequency divider

Digital frequency divider (FD) is a commonly used logic circuit, and can be traditionally implemented by using a series of *n* D flip-flops (for 2^n-division) [107], and each of which consists of 25 or so transistors [15]. When an odd-integer division is required, the circuit could become more complex with extra logic gates. Alternatively, one can implement the *N*-divider using a new architecture based on SET turnstile FD unit (as shown in Fig. 2.8(a)). This unit has a SET turnstile, a parallel SET/MOS, a threshold comparator, and a reset module. The input signal is connected to the gate terminal of the turnstile. A parallel SET/MOS circuit is used to measure the charges stored in the SN. The second gate of the SET transistor is connected to a phase-control voltage V_{PC}. Due to the constant current bias for MOSFET, the voltage oscillation cross the SET transistor is amplified to the drain terminal of the NMOS. This drain voltage is sent to both reset and comparator modules, and the source terminal of NMOS is connected to V_{SS} to bias the transistor at its sub-threshold region.

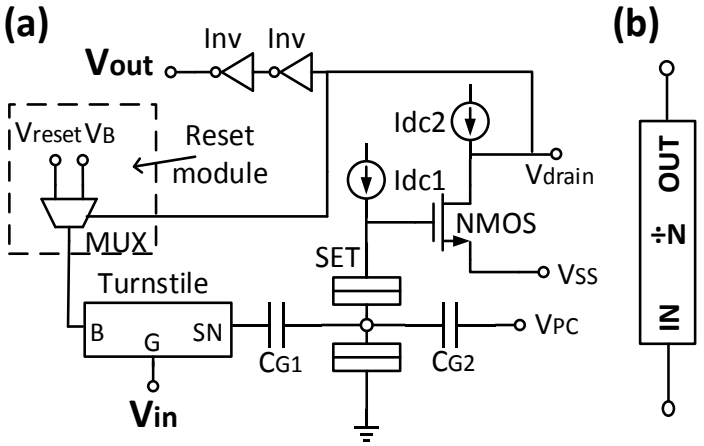


Figure 2.8. (a) Schematic of a SET turnstile digital FD unit, and (b) its circuit symbol.

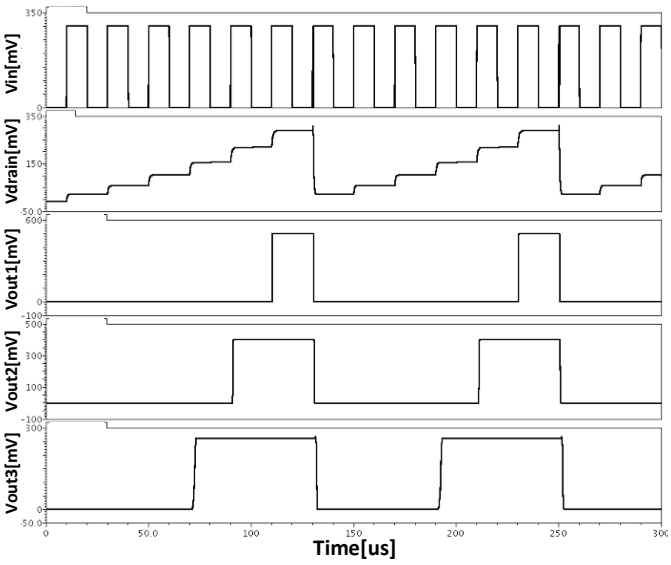


Figure 2.9. Simulation results of a 6-division SET turnstile FD unit in Cadence with 1/6, 1/3, and 1/2 duty cycle.

The operating principle of the proposed FD unit is described as follows. Assume the input V_{in} is a periodic pulse, the turnstile acts as a digital counter since the potential of SN (and hence V_{drain}) changes discretely with the number of input periods. If N=6 for instance, we set the V_{drain} that corresponds to $q_{SN} = 6e$ beyond the reset threshold. This can be done by adjusting V_{SS} and V_{PC} as the NMOS operates at the sub-threshold region. Once V_{drain} is larger than this threshold, the multiplexor switches from V_B to V_{reset}. This makes the potential of the SN as well as V_{drain} and V_{out} return back to their initial values, and the turnstile starts over to count the input periods from $q_{SN} = 0$. Therefore, for 6 periods of V_{in}, the FD unit generates one pulse for the output V_{out} with the frequency $f_{out} = f_{in}/6$, where f_{in} is the frequency of V_{in}. In general, the value of N can be adjusted by C_{G2}, V_{PC} and V_{SS}. Also, the duty cycle of the output can be anywhere from 1/N, 2/N, …, up to (N–1)/N by adjusting the threshold voltage of the comparator. Fig. 2.9 shows the simulation results of a 6-division unit with different duty cycles (the parameters are: I_{dc1} = 40nA, $C_S = C_D = C_{G1} = C_{G2}$ = 0.1aF, I_{dc2} = 1uA, V_{ss} = −140mV, V_{pc} = 578mV, V_B = 100mV, and V_{reset} = −158mV).

Compared with current frequency dividers, the proposed circuit is very compact. A single FD unit only consists of one SET turnstile, one SET transistor and nine MOS transistors (one NMOS in SET/MOS circuit, two PMOS as constant current sources, and six transistors in the extra logic). For instance, to realize 8-division, one needs only one FD unit whose N is adjusted to 8. That means the whole circuit consists of two SEDs and nine MOSFETs. In contrast, the traditional architecture requires four D flip-flops which contain about 100 transistors.

2.3.3 Frequency-voltage level converter

In some circuit applications, the digital information is represented in frequency domain. For example, in the frequency modulation scheme proposed in [84], the result of a multiplier is represented as the ratio of the output frequency and a reference frequency (refer to Fig. 2.10).

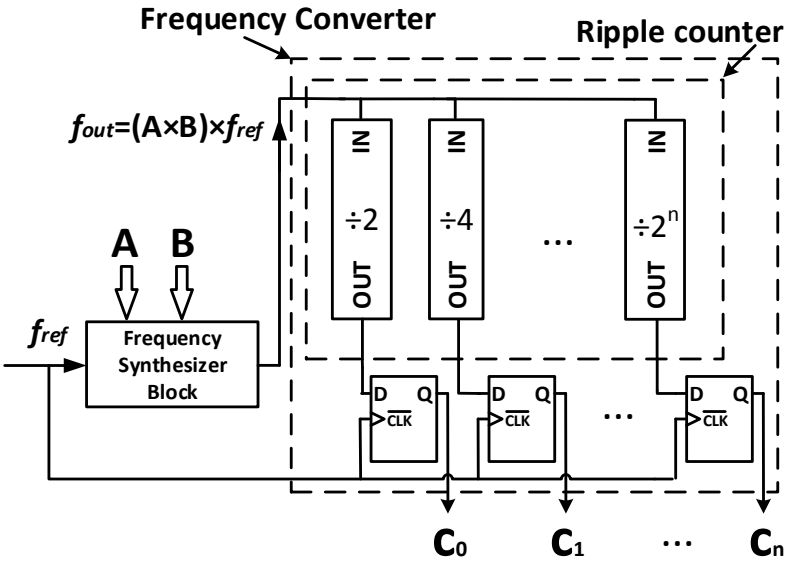


Figure 2.10. SET turnstile FD units used as a frequency-voltage level converter, where ($C_{n,\dots}C_1C_0$)=(A×B).

The output frequency of the frequency synthesizer block (FSB) is (A×B) times the reference frequency. To convert this ratio (i.e. A×B) to a digital output, a number of SET turnstile digital FD units are used to implement the frequency-voltage level converter. The proposed converter consists of a ripple counter and an output module. By connecting these FD units in parallel (set N = 2, 4, … 2^n for FD units where the parameters can be adjusted accordingly as discussed above), the ripple counter counts the number of pulses of the input signal which is connected to the FSB's output. The output module consists of several D flip-flops that latch the results from the ripple counter. The reference frequency is used as a clock signal. Therefore, the binary results reflect the ratio of f_{out} and f_{ref} (or, more specifically, ($C_{n,\dots}C_1C_0$) = (A×B)). In real applications, the FD units need to be reset following the output generation. Fig. 2.11 shows the simulation results for A×B = 110.

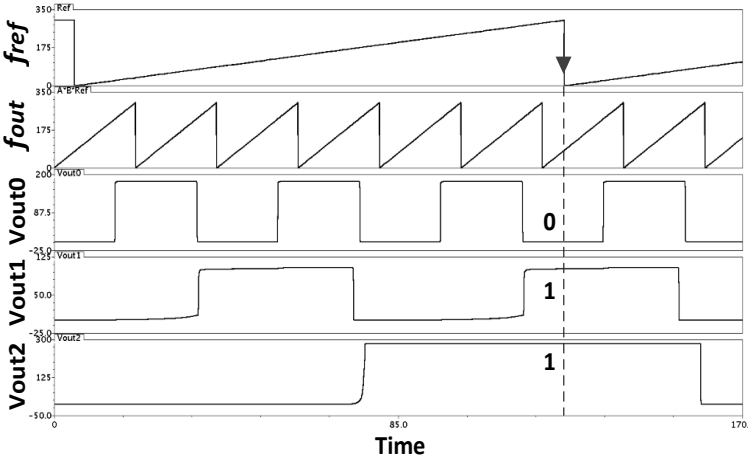


Figure 2.11. Simulation results of Fig. 11 for A×B = (110).

2.3.4 Phase-frequency detector

SET turnstiles can also be used for implementation of phase-frequency detector (PFD). Fig. 2.12 shows the schematic, which consists of two SET turnstiles and a series SET/MOS circuit as an electrometer. The bias voltages of turnstiles 1 and 2 are connected to V_B and $-V_B$, respectively. Two pulse signals V_{in1} and V_{in2} are applied to their gate terminals. One turnstile transfers electrons at frequency f_1, while the other does so at frequency f_2. These two processes will conversely influence the island voltage of the SET transistor that is couple-connected with the two turnstiles.

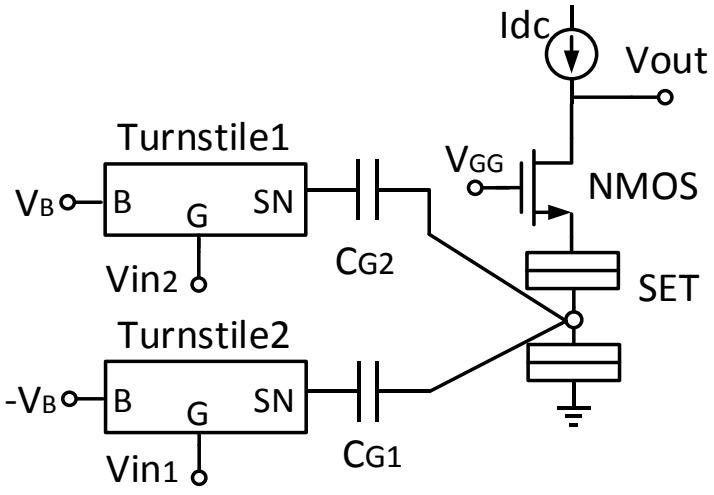


Figure 2.12. Schematic of a SET turnstile PFD.

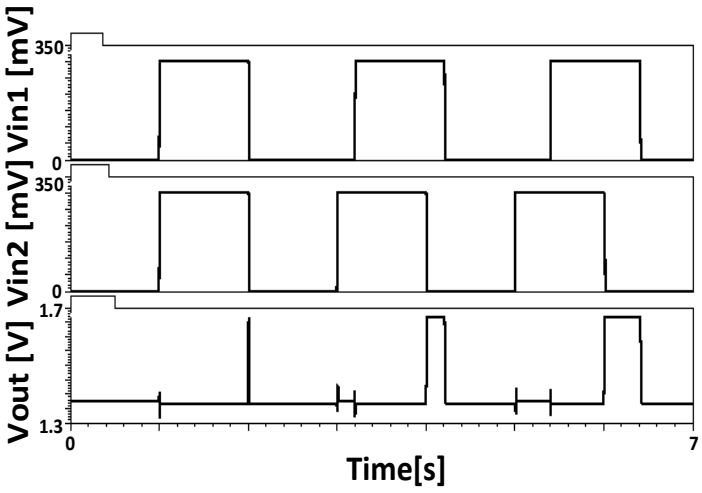


Figure 2.13. Simulation result of Fig. 2.12 in Cadence.

Assume initially there is no extra charge on SN_1 and SN_2, and that the phase of second input signal V_{in2} is ahead. When the falling edge of V_{in2} arrives, one electron is transferred to SN_2, reducing the potential of SN_2. This reduction is detected and amplified by the SET/MOS circuit. Similarly, at the falling edge of V_{in1}, an electron is pulled from SN_1, which increases the potential of SN_1. This will make the output voltage go back to its initial value. Therefore, the pulse width of the output voltage of SET/MOS circuit reflects the phase error of the two input signals. The parameters of turnstiles in Fig. 2.12 are the same as before, and other components are: $C_S = C_D = C_{G1} = C_{G2} = 0.1$aF, $V_{gg} = 655$mV, and $I_{dc} = 50$nA which is to bias the NMOS to work at the sub-threshold region. Fig. 2.13 shows the simulation results.

For comparison, the performance of PFD using different technologies was estimated. The result is summarized in Table 2.1. In [68], a flip-flop is built using SET transistors and threshold logic gates, and a PFD is implemented as two flip-flops. In the proposed PFD, it contains two turnstiles, one SET transistor and two MOSFETs (one NMOS in SET/MOS circuit and one PMOS as a constant-current source). For the typical current MOS approach [106], a PFD consists of about 30 transistors. In the proposed circuit, I_{dc} is generically set to about 50nA, and V_{DD} is set to 1.8V in 0.18μm technology. Thus, the static power is estimated as ~90nW. For a loading capacitance of 1fF at frequency of 500MHz, the dynamic power is nearly 1μW. It can be seen from Table I that the pure SET circuit consumes extremely low power, but requires more circuit components. However, the proposed turnstile PFD is compact with reasonably-low power.

TABLE 2.1. Estimated Performance for a PFD using different Designs

Design	**Area**		**Power (500MHz)**
	No. of MOSFETs	*No. of SE Components*	
Pure SET [68]	0	~ 160	~12pw
Proposed turnstile based (0.18µm)	2	16	~1uW
Pure MOS(0.25µm) [106]	~30	0	1.4mW

2.3.5 Implementation of the conscience mechanism in competitive learning

Artificial neural networks, which consist of many nonlinear parallel processing elements, have been investigated for decades in the field of speech and image recognition [9]. Among them is the self-organizing feature maps proposed by Kohonen in 1984 [16]. The idea of this self-learning neural network is to reinforce the processing elements with maximum response, which is accomplished by modifying the weight vectors of the winning neurons more responsive to the current input. One of major problems with Kohonen learning is the large number of iterations (training data) required to reach a proper solution. This is mainly due to so-called "dead neurons" whose weights remain almost unchanged during the whole learning process. These dead neurons may come from the improper initial values of weights [102] or input data with a special probability density function (e.g., input vector space with one or several gaps). One of the conventional solutions to activate these dead neurons is so-called conscience mechanism first proposed by De Sieno [103]. Its main idea is to bias the competition process so that all neurons have chances to win and take part in the learning (weight adaption) process.

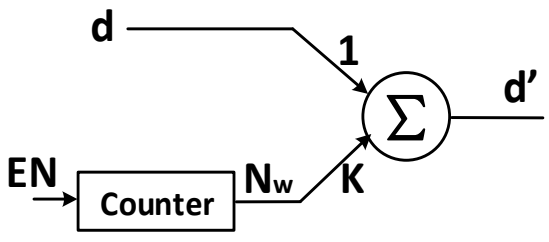


Figure 2.14. Block diagram of proposed conscience mechanism architecture.

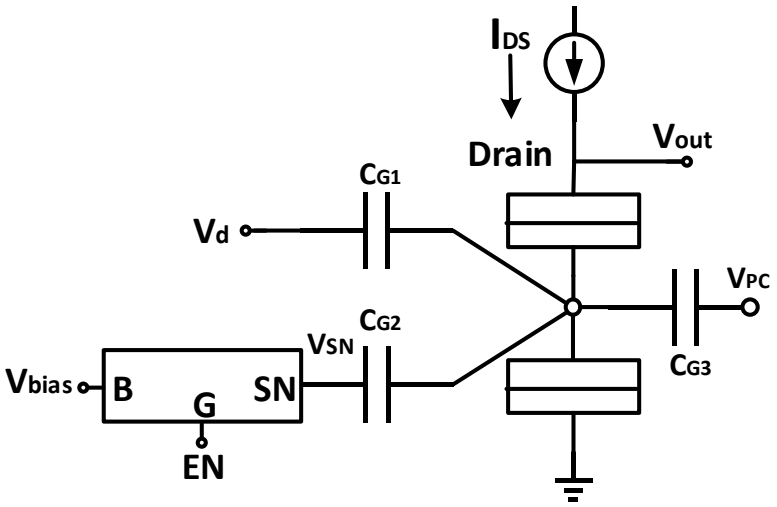


Figure 2.15. Proposed circuit structure to implement conscience mechanism.

Here I present a new circuit architecture that realizes the conscience mechanism by the combination of conventional CMOS characteristics and the electron transfer behavior within SEDs. I implement the conscience mechanism using an analog ripple counter, which is based on electron transfer characteristics within SET turnstiles. The proposed circuit architecture has significant benefits in terms of power dissipation and area cost, as compared to its CMOS counterpart.

There are several methods to realize the above-mentioned conscience mechanism. One of them is the so-called frequency sensitive competitive learning (FSCL) presented by Ahalt *et al.* [104], where the counting elements affiliated to every neuron are used. These elements are able to evaluate the winning times for each neuron, then bias the learning competition. For a specific neuron, this is done by adding an excess value to its original value of Euclidean distance (*d*), which is treated as the measure of dissimilarity between the training input data and its weight. Since this extra value is proportional to the number of wins in current neuron during the past competition (denoted by N_w), the winning likelihood of this neuron decreases in the next iterations (input training data), thus allowing other neurons to compete to win [105]. The updated Euclidean distance *d'* can be expressed as:

$$d' = d + K \cdot N_w \tag{2.8}$$

where K is the gain factor which can be adjusted to control the strength of the conscience mechanism. Thus, the basic building block of the conscience mechanism operation is an analog ripple counter, whose output represents the number of ripples (wins) at the input (EN signal), as shown in Fig. 2.14. The counter can be easily implemented with hybrid SET/MOS architectures with significant reduction in circuit complexity, as shown in Fig. 2.15, which consists of a SET turnstile (as the ripple counter), a SET transistor (as the weighted adder), and a current source to bias the SET transistor. The gate terminal of the SET turnstile is connected to EN signal which has two logic values (high or low) indicating whether the neuron is the winner at current competition. In other words, for the winning neuron, its EN voltage shows a logic high value while EN signals for other neurons remain logic low. This intermittent pulse signal is then used to control the electron transfer within the turnstiles. Since the voltage on the storage node changes linearly with the electron charges accumulated (denoted by Δq), the potential change at SN can be expressed as:

$$\Delta V_{SN} = \frac{4(C_T + C_G) \cdot \Delta q}{C_T^2 + 2C_T C_G + 4C_{SN}(C_T + C_G)} \tag{2.9}$$

where C_T, C_G and C_{SN} are the capacitances of tunnel junctions, gate terminal and storage node, respectively (see Fig. 2.3(a)), and $\Delta q = N_w e$. The drain voltage of SET transistor is a weighted sum of input voltages with weights determined by the ratios of each gate capacitance (C_{Gi}) and total capacitance on the island ($C_\Sigma = C_{TD} + C_{TS} + \sum C_{Gi}$).

It is understood that for current-biased SET transistors, the amplitude of the drain voltage oscillation is inversely proportional to C_Σ with positive slopes of $C_{Gi} / (C_\Sigma - C_{TD})$, with respect to V_{Gi} (i = 1, 2, 3). In our design, the input voltage range is limited within half a period so that the drain voltage increases monotonically with gate voltages; and the third gate terminal connected to V_{PC} is used to adjust the phase of Coulomb oscillation so that the output drain voltage equals to V_d (which corresponds to the original distance d) when there are no extra charges at the SN. Therefore, the output voltage can be expressed in terms of V_d and V_{SN} as follows:

$$\Delta V_{out} = \frac{C_{G1}}{C_\Sigma - C_{TD}} \cdot \Delta V_d + \frac{C_{G2}}{C_\Sigma - C_{TD}} \cdot \Delta V_{SN} \tag{2.10}$$

By substituting ΔV_{SN} of (2.9), the gain factor K of conscience mechanism can be estimated by:

$$K \cong \frac{C_{G2}}{C_{G1}} \cdot \frac{4(C_T + C_G) \cdot e}{C_T^2 + 2C_T C_G + 4C_{SN}(C_T + C_G)} \tag{2.11}$$

which depends on capacitances values in SET transistors and turnstiles. This indicates that the proposed voltage-mode circuit design takes advantage of SET characteristics for a compact structure.

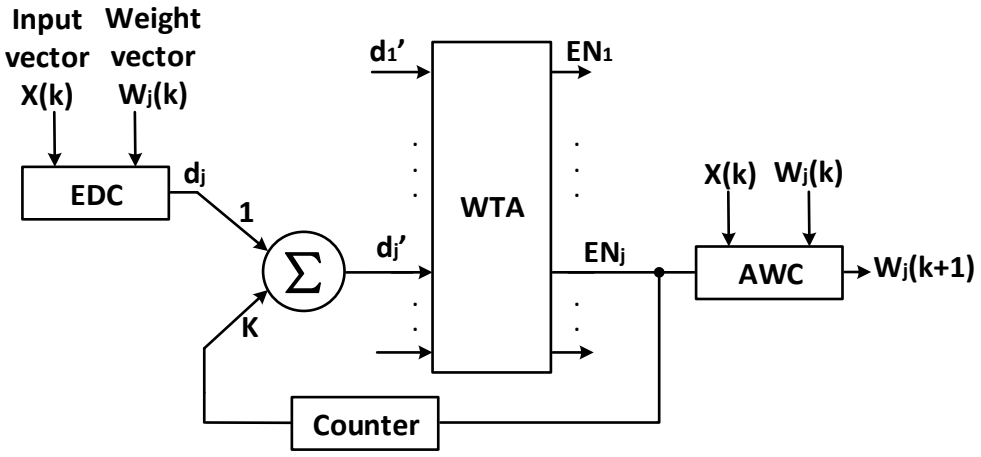


Figure 2.16. Block diagram of the whole self-organizing neural network.

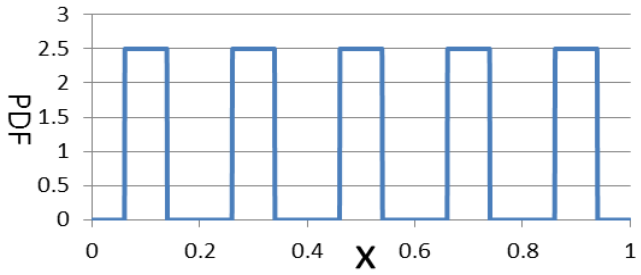


Figure 2.17. Probability density function of the input data used in exemplar test.

At first, I perform system level simulation to verify the effectiveness of conscience mechanism architecture for different choices of parameters value. The block diagram of the entire neural network system is shown in Fig. 2.16, where it includes four main modules which are Euclidean distance calculation (EDC), winner-take-all block (WTA), adaptive weight change (AWC) and proposed conscience mechanism elements. In this study, the quantization error was used as the measure of the quality of self-learning process. An example test was conducted with 1-D input training data which is uniformly distributed within several small intervals of {0.1, 0.3, 0.5, 0.7, 0.9}. The probability density function of input data is illustrated in Fig. 2.17. The

proposed neural network has five adaptive neurons with the initial weight value around 0.5. The conscience bias factor K is set to {0, 0.01, 0.1} in order to compare the performances of different conscience strengths. The results are illustrated in Fig. 2.18, where the quantization error decreases as more training cycles are accomplished.

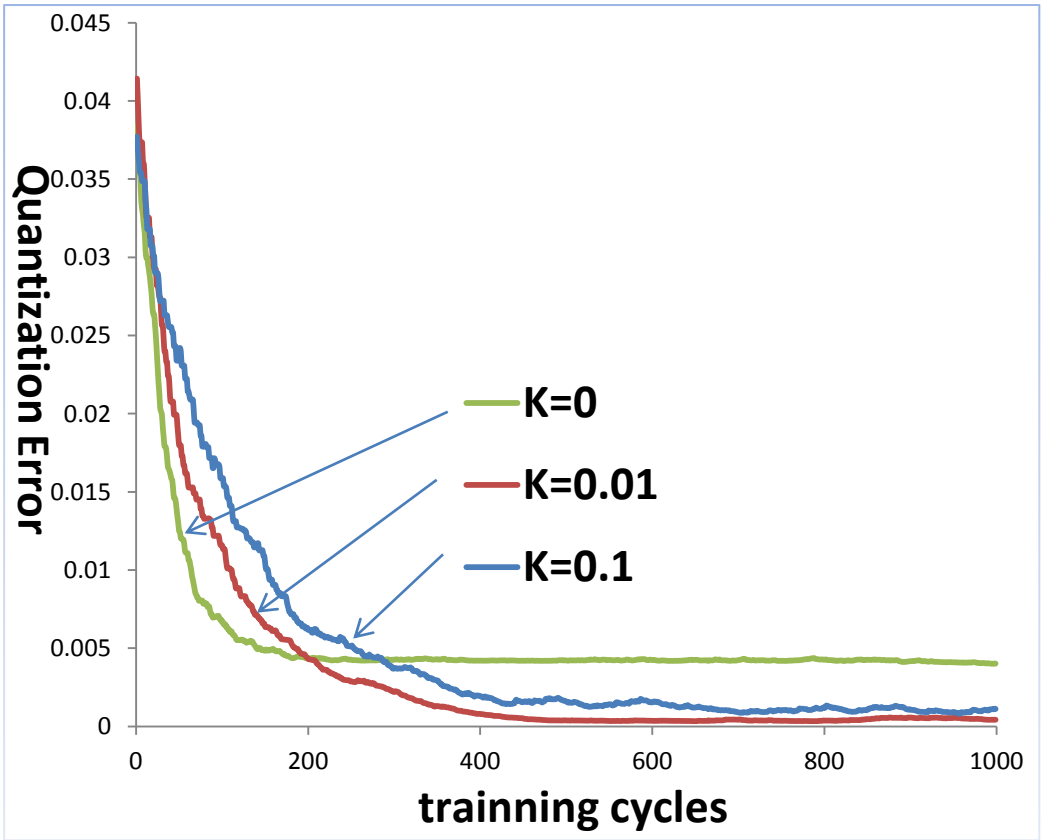
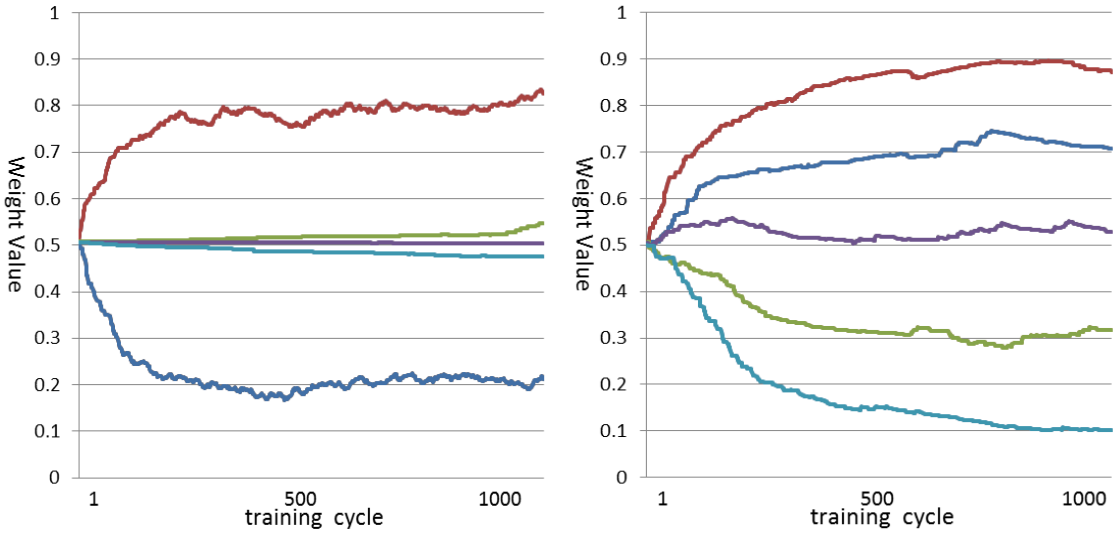


Figure 2.18. Performances of learning process with different conscience mechanism strengths.

When the conscience mechanism is turned off (K = 0), the network has the highest learning speed at the beginning, but the final quantization error is also high. With the conscience mechanism, however, the weights converge slowly with lower error. It can be seen from this specific example that the network with smaller conscience bias factor will have better performance in terms of convergence speed and quantization error, as compared to larger K value. This inspires us to use small capacitance value for C_{G2}, which is proportional to the conscience gain factor as shown in (2.11). In addition, the weight adaptation process for the cases of K = 0 and K = 0.01 is shown in Fig. 2.19(a) and (b), respectively. It can be seen that when the conscience mechanism is turned off, the proper distribution of neuron weights is not shown even

at the end of run. This leads to its high quantization error as in Fig. 2.18. However, the obtained results in Fig. 2.19(b) show that all five neurons have been properly placed to represent each input data spaces.



(a) (b)

Figure 2.19. (a) Weights adaptation process when the conscience mechanism is turned off (K = 0), (b) Weights adaptation process with conscience mechanism (K = 0.01).

The proposed circuit architecture was simulated under Cadence environment. The parameters for simulation are summarized in Table 2.2. The EN was set to be a voltage pulse signal with magnitude of 300mV. The V_d is set as a ramp signal ranging from 100mV to 150mV over a period of 5μs. Fig. 2.20 shows the obtained timing diagrams of analog ripple counter. It can be seen clearly that as a ripple of V_{EN} comes in, an electron is transferred out of the storage node. The potential V_{SN} then increases accordingly. The output voltage is proportional the V_d; and it also reflects the potential change at SN due to the electron transfer by voltage shifting. The DC response (which is not shown for simplicity) indicates that the variation for V_{out} is 17.3mV with $V_d \in$ [0.1V, 0.2V], and that the voltage shift due to single electron transfer is 0.9mV. Thus, the gain factor of conscience mechanism can be calculated as $K = 0.9\times10^{-3}\times100/17.3 \approx 0.0052$, which is very close to an estimated value of 0.0053 calculated by (2.11).

One of the unique and important features of SET turnstile is the stochastic nature of electron tunneling involved. Here when V_{bias} or V_G changing from its preferred value of electron transfer,

there is still a non-zero possibility of electron tunneling through the junctions due to thermal energy (if the temperature is not ideal). This thermal tunneling effect can be suppressed by setting bias voltage farther from its original preferred value, which also means a lower conscience gain factor in stochastic sense under same temperature. When the extreme low (but non-zero) conscience strength is needed (as I have shown previously), a corresponding setting or adjustment of bias voltage is more realistic for application circuits especially at non-ideal temperature environment, and also due to the difficulty in realization and tuning of extreme small capacitances C_{Gi}.

TABLE 2.2. Simulation Parameters

Temperature	300K
C_T	0.1aF
C_G	0.5aF
C_{SN}	30aF
C_{TS}, C_{TD}, C_{G1}, C_{G2}, C_{G3}	0.1aF
R_T, R_{TS}, R_{TD}	1MΩ
I_{DC}	20nA
V_{PC}	–570mV
V_{bias}	100mV

At a relatively high temperature, the thermal energy will introduce the undesired error tunneling, which may cause the outputs of conscience mechanism blocks deviated from their original value. However, since the deviations happen at all neurons, their overall effect would be eliminated statistically by the following WTA block. It is widely known that one of the essential features with neural networks is their high fault tolerance, i.e., the overall performance would not be impaired significantly with a few damaged neurons [9]. This characteristic allows us to employ SET devices, which are less reliable for room-temperature operation, but have advantages of low power and area cost.

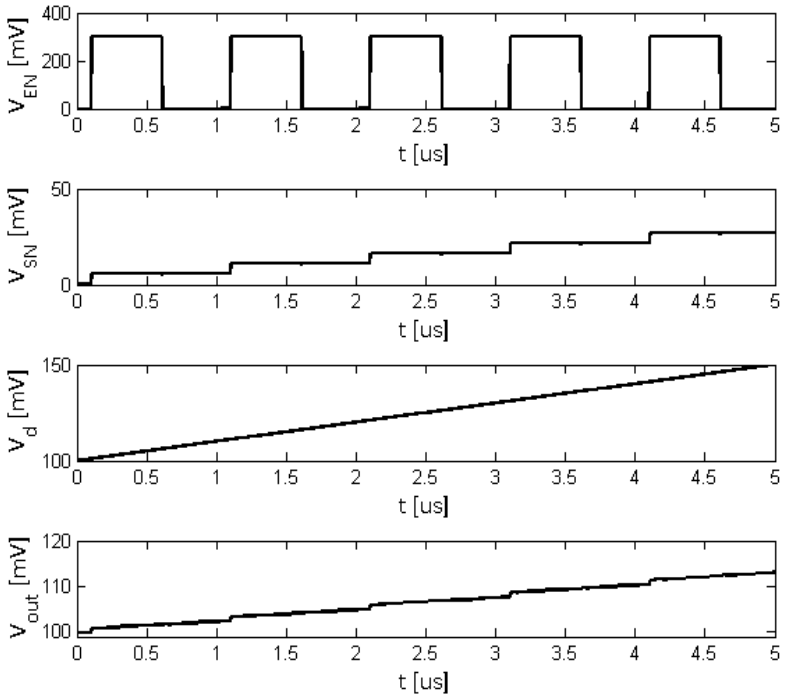


Figure 2.20. Transient response of the proposed SET based conscience mechanism circuit.

It should be pointed out that the implementation of conscience mechanism with CMOS technology would require a high area cost with more transistors (~40 transistors as mentioned in [105]). With even smaller feature size of SET devices, the proposed circuit architecture becomes more area-effective. In terms of power consumption, the CMOS implementation (where the current mode is usually employed) for an analog counter would require a DC current of around ~10μA. While this is acceptable for small-scale designs, it would demand extremely high power dissipation for a large number of neurons involved. In comparison, the static current in our design is ~10nA, which represents a 100x improvement in static power. Therefore, it can be concluded that the proposed SET/MOS architecture outperforms its CMOS counterpart in terms of both area cost and power consumption.

A potential problem with electron transfer is that the voltage swing due to charge accumulation is very limited (only few tens of mV). Also, the current used to bias SET transistors has to be small enough in order to enable the Coulomb oscillation, leading to a limited voltage gain with this architecture. This may cause some problem in logic circuits such as low noise margins and drivability. Fortunately, the inherent small output swing is actually preferable in our design that requires extremely low conscience mechanism strength. However, the following WTA

block must have enough high resolution to detect this potential changes due to electron transfers. In addition, for other parts in the Kohonen learning networks, SET-based implementation is also a feasible option that permits simple circuit structure and efficient operations. Further research are needed for novel circuit architecture design and advanced processing technologies.

In summary, in this section I have analyzed the impact of using SEDs in neural network circuits design. As an example, the hybrid SET/MOS implementation of conscience mechanism operation were proposed and studied. The novel characteristics of SEDs circuits were found to be compatible with the operation principle of conscience mechanism. The proposed SET-based architecture has unique features that provide significant improvements in circuit area and power consumption. Both system-level and transistor-level simulation results showed the effectiveness of our design in competitive learning process. In real applications, the unreliable feature of SEDs can be suppressed by the inherent fault tolerance of neural networks, which makes the SET-based architectures promising in the future for large-scale high density circuit design.

2.4 Summary

Since the charges on single-electron islands in turnstiles are quantized, their behaviors could be difficult to model using conventional components with continuous conduction current. On the other hand, some SE devices exhibit oscillation characteristics with both positive and negative transconductances. It would be a non-trivial task to accomplish the proper switching between these two transconductance regions using equivalent circuits that consist of resistors and voltage sources. While widely used, the SPICE is empirical in nature, and fails to utilize the physical parameters of SE devices.

This motivates us to develop a new Verilog-A model for SET-based turnstiles towards hybrid circuit design. Our simulation results have shown that the proposed work gains an insight to the process of single-electron transfers involved. Also, this compact model allows us to simulate large hybrid circuits within fraction of a second. Different application circuits have also been discussed to demonstrate the advantages of utilizing SET turnstiles for implementation. Especially, I emphasized the application in neural networks since the characteristics of SEDs can be fully exploited; at the same time the fault-tolerant feature of neural networks is able to compensate the some drawbacks of SEDs.

CHAPTER 3

DYNAMIC CHARACTERISTICS AND THE STATISTICAL MODELING OF SINGLE-ELECTRON BASED CIRCUITS

3.1 Background

The past decade has seen growing studies on single electronics, thanks to the availability of production processes for small structures. However, due to the fact that nanoscale devices utilize a few electrons for signal processing, the transient response and switching delay with SET circuits become an increasingly big concern, and demands accurate models in order to gain insight into their dynamic characteristics [66]. Unlike MOSFET devices where the current can flow through the conductor as electrons are free to move through the lattice of atomic nuclei, the tunnel junctions in SET circuits only allow electrons to penetrate the energy barrier one by one. Particularly, with the assumption that only one electron can tunnel at a time (i.e., high-order tunnels are ignored), the tunnel events within the circuit can be approximately characterized by the so-called birth-death process, which is a special case of continuous-time Markov process. Given that the dynamic behavior of SET circuits is stochastic in nature, the switching delay cannot be measured and analyzed in the traditional sense.

We study the stochastic nature of SET circuits and propose a statistical method for quick delay estimation. The effectiveness of proposed method will be analyzed theoretically; and our numerical evaluations show that despite some approximations used, the proposed method provides a simple and efficient way to model the dynamic behavior of SET circuits, even at relatively-high temperatures. I also look at the delay of typical SET logic gates, and show the advantages of the proposed method over other approaches.

The rest of this Chapter is organized as follows. Section 3.2 begins with analysis of a statistical delay model for SET circuits under low temperature and its numerical evaluation, followed by examples of a SET inverter and NAND gate in Section 3.3. In Section 3.4, I compare the performances of some existing simulation methods for SEDs in terms of delay estimation. In Section 3.5, I discuss the impact of non-ideal inputs on the delay, and extend the proposed model to consider the temperature effects. The Chapter is concluded in Section 3.6.

3.2 Method

3.2.1 Poisson process

Single-electron tunneling is a stochastic phenomenon and its theory can only predict probability rate Γ of the possible tunneling events [54]. Based on the orthodox theory, one can describe the SET system with a Master Equation [10]:

$$\frac{\partial P_i(t)}{\partial t} = \sum_{j \neq i} [\Gamma_{j \to i} P_j(t) - \Gamma_{i \to j} P_i(t)] \tag{3.1}$$

where $P_i(t)$, $i = 0, 1, \ldots, N$ denotes the time-dependent occupation probability of each state (here, a state represents a specific voltage and/or charge distribution in the circuit). $\Gamma_{i \to j}$ is the tunnel rate or transition rate (i.e., probability per unit time) from state i to state j, which depends on circuit parameters and temperature [3]. Solving the Master Equation (ME) requires a set of states together with the transition probabilities of every state to any other state in the set. Generally speaking, the number of electrons on a Coulomb island could be any integer values, leading to an infinite dimension of state space. However, in actual circuits, this number is usually very limited. In terms of delay estimation for functional SET logic circuits where the digital information is represented by several stable states, we can simplify the ME with some reasonable approximation.

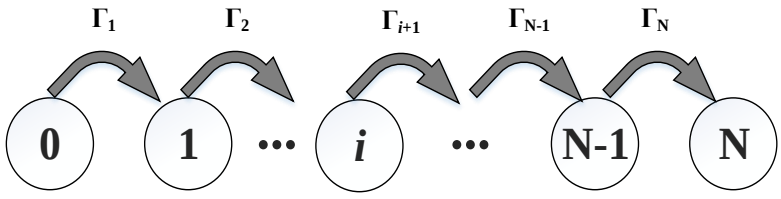


Figure 3.1. The Poisson process in a system with N+1 states $\{0, 1, \ldots, N\}$ and N tunnel events whose tunnel rates are $\{\Gamma_1, \Gamma_2, ..., \Gamma_N\}$. The system starts at state 0, and evolves monotonously to higher state numbers, until it arrives at the final state N. Γ_i is the abbreviation of $\Gamma_{i-1 \to i}$, which is the tunnel rate from state $i-1$ to state i.

The logic transition within a SET circuit involves the transitions between two states (or charge distributions). If we assume that the temperature is extremely low (ideally, T~0K and the general case under high temperatures will be discussed in Section 3.5), and that the tunnel rate for co-tunneling is zero (i.e., a state can only change to its neighboring states), then the state transitions can be viewed as a Poisson process, as illustrated in Fig. 3.1. The Poisson process starts from an

initial probability vector $\{P_0(t_0) = 1, P_i(t_0) = 0 \text{ for } i \neq 0\}$, and evolves monotonously to higher state numbers. Assume that a logic transition involves N tunnel events in total (i.e., N excess electrons produce a logic level change at the output), which make state transfer from state 0 to 1, 2, …, and eventually to the final state N. When evaluating the circuit delay, we shall consider the state distribution $P_N(t)$, which gives the time-dependent probability of the system reaching the state N. Therefore, for a specific moment T_N, one can define an error probability $P_{error,\,N}(T_N) = 1 - P_N(T_N)$, which determines the statistical accuracy of estimating the delay as T_N. In other words, the circuit delay is associated with a certain value of state probability or error probability.

More specifically, if all N tunnel events have same tunnel rate (i.e., $\Gamma_1 = \Gamma_2 = ... = \Gamma_N = \Gamma$, which corresponds to a homogeneous Poisson process), the state probability of states 0 to $N-1$ follow the Poisson distribution. The time-dependent probability of the system reaching state N is (refer to [66]):

$$P_N(t) = 1 - \sum_{i=0}^{N-1} P_i(t) = 1 - \sum_{i=0}^{N-1} \frac{(\Gamma t)^i}{i!} \cdot \exp(-\Gamma t) \tag{3.2}$$

For a more general case when N tunnel event rates are all different (i.e., $\Gamma_1 \neq \Gamma_2 \neq ... \neq \Gamma_N$, which corresponds to a non-homogeneous Poisson process), the $P_N(t)$ is governed by the so-called *Bateman* equation [56] and can be written as:

$$P_N(t) = \lim_{\Gamma_{N+1} \to 0} \left[\frac{1}{\Gamma_{N+1}} \cdot \sum_{i=1}^{N+1} \Gamma_i \cdot \left(\prod_{j=1,\, j \neq i}^{N+1} \frac{\Gamma_j}{\Gamma_j - \Gamma_i} \right) \cdot \exp(-\Gamma_i t) \right] = 1 - \sum_{i=1}^{N} \left(\prod_{j=1,\, j \neq i}^{N} \frac{\Gamma_j}{\Gamma_j - \Gamma_i} \right) \cdot \exp(-\Gamma_i t) \tag{3.3}$$

By solving (3.3), one can comprehensively describe the dynamic behavior of the system. To this end, we define N random variables $t_1, t_2, \ldots, t_N$ to denote the arrival time of each state. The state probability distribution $P_N(t)$ also represents a cumulative distribution function (CDF) of t_N, i.e.,

$$\begin{cases} P_N(t) = P(t_N \leq t), t \in [0, +\infty) \\ P_N(0) = 0, P_N(\infty) = 1 \end{cases} \tag{3.4}$$

Unfortunately, calculating $P_N(t)$ using (3.3) could be either numerically unstable (if N is a large number and some tunnel rates are very close, but not equal, to one another) or computationally tricky (when more than one groups of tunnel rates share the same value) [56, 66]. Also, the Master Equation method can hardly solve a large-scale circuit as the computational cost of solving the ME will grow exponentially with circuit size. This motivates us to develop some alternative methods, one of which will be proposed below.

3.2.2 Proposed statistical method

To introduce the proposed method, we first define $\Delta t_i = t_i - t_{i-1}$ to be the time interval between state i and state $i-1$ (i.e., the i-th inter-arrival time), or the "life time" of state $i-1$. Based on the orthodox theory, electrons tunnel independently from island to island through junctions, changing the charge distribution (or state) in the circuit. The time spent at a state only depends on the transition rate from this state to the next state. Thus, Δt_i's (i = 1, 2,…, N) are N independent random variables, and each of them follows an exponential distribution with the probability density function (PDF) of:

$$f_{\Delta t_i}(t) = \Gamma_i \exp(-\Gamma_i \cdot t) \tag{3.5}$$

Every Δt_i has its expected value $E(\Delta t_i) = 1/\Gamma_i$ and variance $D(\Delta t_i) = 1/\Gamma_i^2$, where the tunnel rate Γ_i can be interpreted as the time-dependent Poisson intensity for a non-homogeneous Poisson process. Since the arrival time t_N can be expressed as the sum of the N stochastic variables Δt_i (i = 1, 2,…, N), the *Lyapunov* central limit theorem applies if N is extremely large [13]:

$$\lim_{N\to\infty} P\left\{ \frac{t_N - E(\sum_{i=1}^{N} \Delta t_i)}{\sqrt{\sum_{i=1}^{N} D(\Delta t_i)}} \le t' \right\} = \lim_{N\to\infty} P\left\{ \frac{t_N - \mu}{\sigma} \le t' \right\} = \int_{-\infty}^{t'} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

$$\textit{where} \quad \begin{cases} \mu = E(\sum_{i=1}^{N} \Delta t_i) = \sum_{i=1}^{N} E(\Delta t_i) = \sum_{i=1}^{N} \frac{1}{\Gamma_i} \\ \sigma^2 = \sum_{i=1}^{N} D(\Delta t_i) = \sum_{i=1}^{N} \frac{1}{\Gamma_i^2} \end{cases} \tag{3.6}$$

This indicates that the random variable $(t_N - \mu)/\sigma$ converges in a standard normal distribution, as N becomes large. Therefore, the probability distribution of arrival time t_N can be approximated by a normal distribution with an expected value of μ and standard deviation of σ, when N is relative large.

The procedure of the proposed delay estimation method is summarized as follows:

(i) For a certain logic transition, calculate tunnel rates for all N tunnel events (i.e., Γ_1, Γ_2, ... Γ_N) according to the orthodox theory.

(ii) Find the expected value μ and variance σ^2 using (3.6).

(iii) The CDF of the switching delay t_N is approximated by a normal distribution ~*Gaussian* (μ, σ^2).

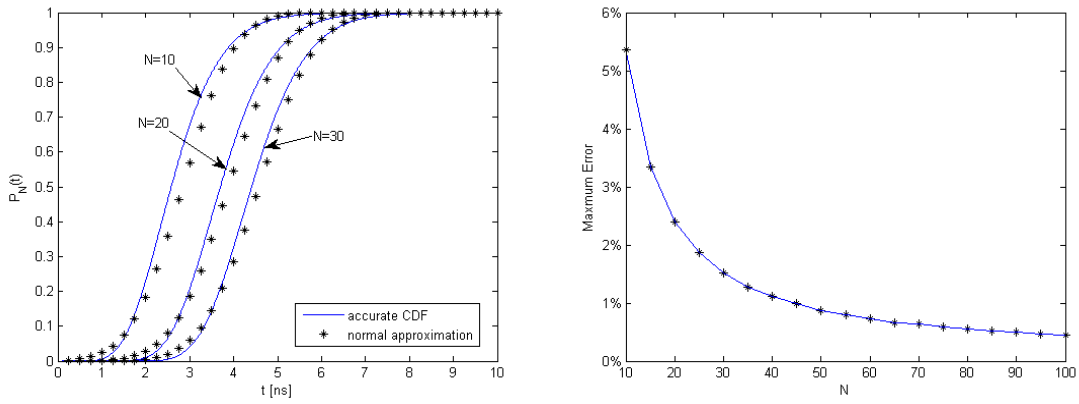
(iv) For a specific value of $P_{error,\,N}$, calculate the delay T_N using the following function:

$$T_N = \Phi^{-1}(1 - P_{error,N}) \tag{3.7}$$

where $\Phi^{-1}(\cdot)$ is the inverse function of the CDF for the *Gaussian* random variable t_N ~*Gaussian* (μ, σ^2).

3.2.3 Numerical evaluation

Consider a system with N+1 states, as shown in Fig.3.1 where all tunnel rates are different, i.e., $\Gamma_1 \neq \Gamma_2 \neq ... \neq \Gamma_N$. I begin with a study on approximation using a normal distribution for different values of N. For instance, assume N = 10, 20 and 30, and the value for every Γ_i is randomly chosen in the range (0, N] for the sake of convenience.



(a) (b)

Figure 3.2. (a) The accurate cumulative distribution (solid lines) and its statistical approximation (symbols) of the arrival time t_N for the final state with N = 10, 20, and 30; (b) The maximum delay error by the proposed approximation for different values of N.

The accurate cumulative distribution of t_N, which is calculated from (3.3), and its approximation using the normal distribution are plotted in Fig. 3.2(a). It can be seen from the figure that the normal distribution result is in good agreement with the accurate distribution of t_N, especially as N increases. This implies that one can estimate the CDF of the switching delay t_N in real applications using a normal distribution ~*Gaussian* (μ, σ^2), without having to solve equations

(3.3) and (3.4) which might be numerically unstable. To use (3.6) and (3.7) for delay estimation, N has to be large enough to make a good approximation. Fig. 3.2(b) shows the maximum error of the proposed method against N, which is defined as:

$$Error_{\max} = \max_{P_N \in (0,1)} \{ \frac{|T_{accurate}(P_N) - T_{approximate}(P_N)|}{T_{accurate}(P_N)} \} \tag{3.8}$$

where $T_{accurate}$ and $T_{approximate}$ are the delay results from the exact solution of ME and the proposed method, respectively. As expected, the maximum error decreases as N increases (in particular, the error is less than 2% for $N > 30$).

It can be seen from Fig. 3.2(b) that this relative error is still acceptable (less than 6%) even for a relatively-small value of N (N~10), which is a common feature of SEDs. For multi-stage logic circuits consisting of a large number of SET gates, the number of tunnel events involved is relatively large, which means that a better accuracy can be expected for the proposed delay model.

3.3 Examples

3.3.1 SET-based inverter

Here I show an example of how to use the proposed method to estimate the delay of the well-known *Tucker's* inverter [23], which consists of two single-electron transistors, as shown in Fig. 3.3. Consider the output transition from logic"0" to logic "1" ($V_{OUT}(0) \rightarrow V_{OUT}(1)$) when the input V_G transits from 6.67mV to 0V.

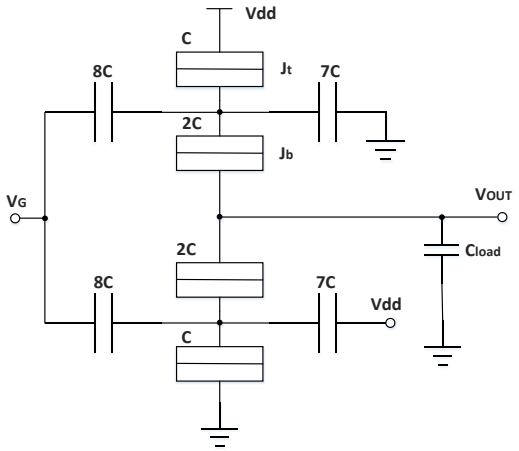


Figure 3.3. The *Tucker's* inverter circuit schematic. The parameters are the same as in [66], i.e., V_{dd} = 6.67mV, C = 1aF, R_T = 1MΩ (resistance for all tunnel junctions), C_{load} = 360aF and T = 0K.

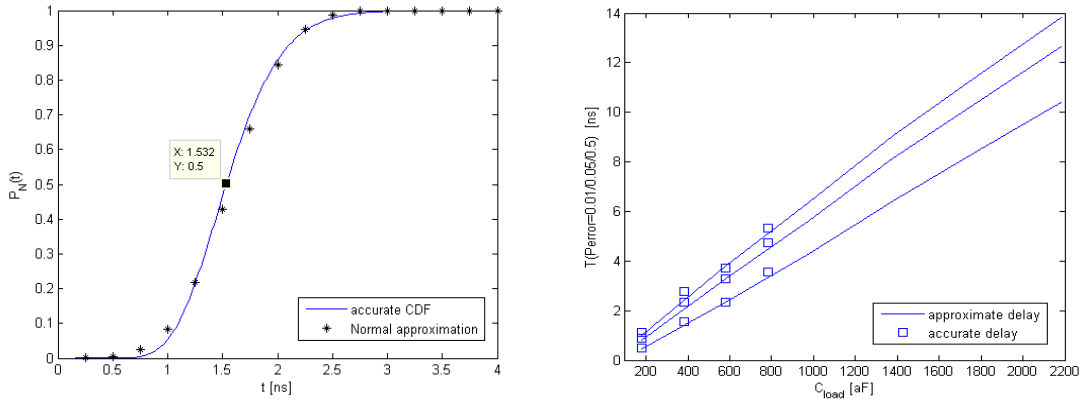
Here, I look at the transient step response at the output node, and the switching delay is defined as the epoch when the output voltage hits 90% of its highest value. For a 0-to-1 transition, only the top transistor is involved in electron tunneling, and all tunnel events will take place alternately in junctions J_t and J_b. The total N tunnel rates can be calculated below (see [66] for details):

$$\begin{cases} N_{tot} = \left\lceil \left(\dfrac{V_{sw} \cdot C_{load}}{e} \right) \right\rceil \\ N = 2 \cdot N_e = 2 \cdot \left\lceil (0.9 \times N_{tot}) \right\rceil \\ \Gamma_{b,i} = (N_{tot} - i + 1) \cdot \Delta\Gamma_b, \Delta\Gamma_b = \dfrac{8}{9 R_T C_{load}} \\ \Gamma_{t,i} = \dfrac{V_{dd} - V_{OUT}(0)}{e R_T} + (N_{tot} - i) \cdot \Delta\Gamma_t, \Delta\Gamma_t = \dfrac{1}{9 R_T C_{load}} \\ where \quad i = 1, 2, ..., N_e \end{cases} \tag{3.9}$$

where N_e is the number of tunnel events that need to be considered within each tunnel junction. Based on the definition of switching delay, 90% of total electron tunnels (denoted by N_{tot}) are computed. $\Gamma_{b,i}$ and $\Gamma_{t,i}$ are the tunnel rates of the i-th tunnel event within J_b and J_t, and $\Delta\Gamma_b$ and $\Delta\Gamma_t$ are the difference of tunnel rates between two consecutive tunnel events within J_t and J_b, respectively. It can be seen that the tunnel rates have the highest value at the beginning and start to drop gradually as i increases. V_{sw} is the output voltage swing, and e is the electron charge. Based on the proposed normal distribution approximation, the expected value and variance of t_N are estimated as follows:

$$\begin{cases} \mu = \sum_{i=1}^{N_e} \dfrac{1}{\Gamma_{t,i}} + \sum_{i=1}^{N_e} \dfrac{1}{\Gamma_{b,i}} \\ \sigma^2 = \sum_{i=1}^{N_e} \dfrac{1}{\Gamma_{t,i}^2} + \sum_{i=1}^{N_e} \dfrac{1}{\Gamma_{b,i}^2} \end{cases} \tag{3.10}$$

With the above equations (3.9) and (3.10), the circuit delay can be estimated by the normal distribution ~*Gaussian* (μ, σ^2). In other words, the proposed method can be used in general to approximately describe the dynamic behavior of a circuit by using a quick estimation of μ and σ^2 from (3.10). For the *Tucker's* inverter in particular, we can plot the CDF of t_N from the normal distribution approximation using different load C_{load} (i.e., different value of N). Again, the solution of above-mentioned Master Equation is used as a reference. The comparison results for C_{load} = 360aF are shown in Fig. 3.4(a), which validate the proposed normal approximation. The estimated delay for $P_{error,\,N}$ = 0.5, 0.05 and 0.01 is shown in Fig. 3.4(b).



(a) (b)

Figure 3.4. (a) Comparison of the accurate method (solid line) and normal approximation (symbols) in estimating $P_N(t)$ for Fig. 3.3 with C_{load} = 360aF. Particularly, the data cursor shows that the error probability is 0.5 for the delay of 1.532ns; (b) The inverter delay versus load capacitance, with the $P_{error,\,N}$ being chosen as 0.5, 0.05, and 0.01.

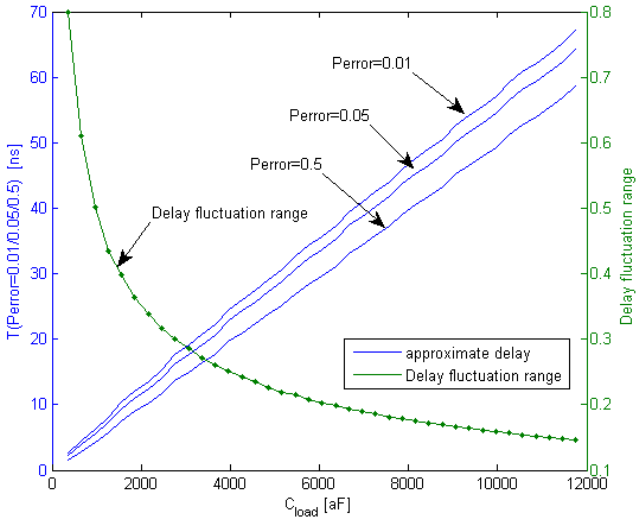


Figure 3.5. The inverter delay and delay fluctuation range ($3\sigma/\mu$) versus load capacitance.

It can be seen that the results from the statistical method match those from numerical solutions of ME for a variety of error probabilities and load capacitances; and it is also worth mentioning that the accurate delay from the ME for C_{load} > 800aF (i.e., $N \geq 50$) is not available in Fig. 3.4(b), due to the fact that computation of (3.3) becomes numerically unstable for large N.

As the total number of tunnels N increases, so do the expected value and standard deviation of switching delay, as shown in (3.6) and (3.10). Fig. 3.5 shows the switching delay of the inverter (under different error probabilities) as a function of load capacitance (or *N*). As the switching delay obeys the Gaussian distribution, there is 99.7% possibility that it falls within the range of $[\mu-3\sigma, \mu+3\sigma]$. Thus, the delay fluctuation can be measured by the ratio of the mean value and standard deviation (i.e., $3\sigma/\mu$), which is also plotted in Fig. 3.5 (dotted curve). This delay fluctuation range is significant for a small value of *N*, but becomes relatively small when more tunnel events are involved in a logic transition (around ±15%, as can be seen in the figure, where three delay curves are almost parallel as C_{load} keeps increasing). This indicates that as *N* increases, the relative delay variation becomes smaller while the absolute delay gets larger. This result can also be expected from (3.6) or (3.10), where the expected value increases linearly with *N* while the standard deviation is proportional to the square root of *N*.

3.3.2 SET-based NAND gate

In addition, as another example, a two-input NAND gate using complementary SET transistors is shown in Fig. 3.6, where its structure is very similar to its CMOS counterpart as two SET transistors are connected either in series or parallel. It should be mentioned that for the cases of output transition due to single input change, only one SET transistor will be involved in electron tunneling and then its switching delay can be analyzed using the same manner as we discussed before for inverter gate. Therefore, in the following we only consider the case of output transition due to both inputs changing together.

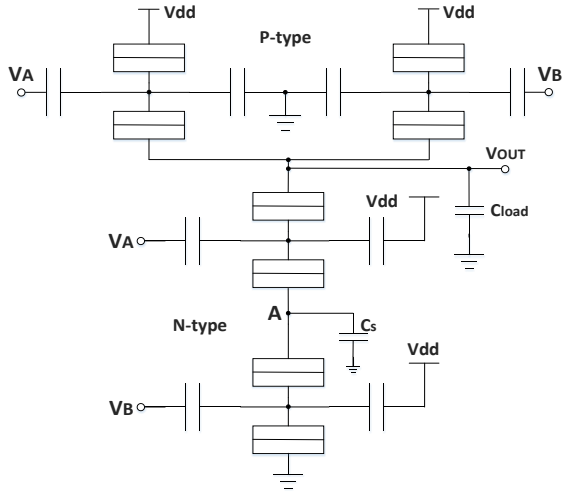


Figure 3.6. Circuit schematic of two-input NAND gate using complementary SET transistors.

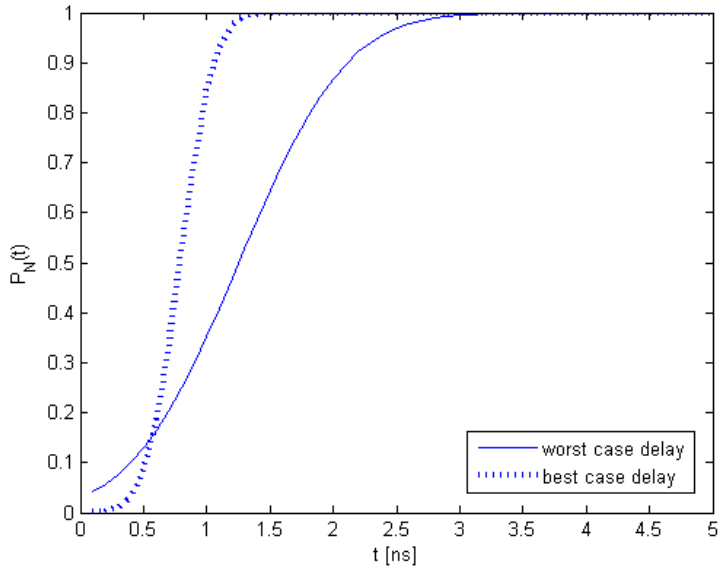


Figure 3.7. The switching delay of 2-input NAND gate under the worst case and best case.

For instance, for the NAND gate in Fig. 3.6, assume both inputs make a transition from logic "0" to "1" at the same time. We consider it as the worst case as electrons have to tunnel through two n-type SET transistors in series then discharge the load capacitance. The tunnel process is described briefly as follow: at first the electrons begin to tunnel through the top n-SET transistor. As the results, the potential of node A is raised; then activates the electron tunnels at bottom n-SET (which will compensate this potential change on node A). It should be mentioned that a large capacitance C_s is added at node A in order to suppress the voltage fluctuations, as we did in [66]. Then the sequential electron tunnels happened in top n-SET can be considered independently. Therefore, the tunnel rates in this case can be estimated using similar method as in (3.9); the time-dependent probability that output hits its 90% point (i.e., the CDF of switching delay) is plotted in Fig. 3.7 (solid line labeled as the worst case delay).

To estimate the switching delay when both inputs make a transition from logic "1" to "0" simultaneously (known as the best case), it is reasonable to assume that two p-type transistors are equally responsible for electron tunneling throughout the output transition [66]. Therefore, the transition rates between two states in the system are doubled from original ones, while the number of tunneling keeps the same. The CDF of its switching delay is also plotted in Fig. 3.7 (shown as the best case delay), where the delay (as well as its fluctuation range) in this case is much lower than it is under the worst case scenario, as expected. Also, by comparing Fig. 3.4(a) and 3.7, we

notice that the delay of NAND gate is slightly shorter than the inverter case, even under its worst case. The reason behind this is the output swing reduction, which is mainly due to the two series-connected transistors.

For other type of logic gates, the NOR gate has the similar switching delay as NAND gate due to the structural symmetry; and in terms of XOR gate, since the output would remain the same value when two inputs are changing together, so there is only one SET transistor being involved during output transition for all input pattern. As we mentioned before, its delay analysis is similar with the inverter case. Here I skip the delay analysis of NOR and XOR gate for simplicity.

3.4 Comparison

As mentioned above, the delay for SET circuits has to be considered in a probabilistic sense, as opposed to the deterministic delay for CMOS counterparts [66]. In this section, several existing simulation methods will be compared and evaluated in terms of delay estimation of single-electron circuits, including the Master Equation method, Monte Carlo method and macro-modeling method. Also discussed is the step estimation method that was recently proposed in [66]. More general explanations on some of these methods can be found in [7].

3.4.1 Master equation method

The operation of SET circuits is characterized by the stochastic tunneling of individual electrons across tunnel junctions. Therefore, the delay estimation is to calculate the probabilities of the electron population in all the Coulomb islands, and this can be done by solving the Master Equation. Once the time-dependent probabilities that the circuit takes on each certain charge distribution are found, the accurate transient characteristics of the circuit will become available from a statistical point of view.

The major drawback of ME method is that in order to obtain all the equations, one has to know all relevant states that the circuit may occupy [7], which may not always available. However, this would not be a problem for delay estimation if we are dealing with a properly-designed and functional circuit. The real challenge is how to find the solution $P_i(t)$ or, particularly when $i = N$, $P_N(t)$ in (3.3) which could be numerically unstable (some approximation methods were used in some previous research works [3, 54]). For the example circuit presented in the previous section, the ME equations will become numerically unstable if N goes beyond 50.

The ME method is quite suitable for small circuits in terms of delay estimation, as it can lead to accurate results by considering secondary effects (such as thermal tunneling, background

charges and high-order tunneling). However, the high computational cost for realistic applications places a great demand for more efficient approaches.

3.4.2 Monte Carlo simulation

The basic idea of MC simulation is to use pseudo-random numbers to determine the time interval Δt_i between two consecutive tunnel events:

$$\begin{cases} \Delta t_i = -\ln(r)/\Gamma_i, r \sim U(0,1) \\ t_N = \sum_{i=1}^{N} \Delta t_i \end{cases} \tag{3.11}$$

where r is a random number uniformly distributed over the interval (0, 1). The delay measured by this method is also a random variable, leading to the fluctuation among different MC trials.

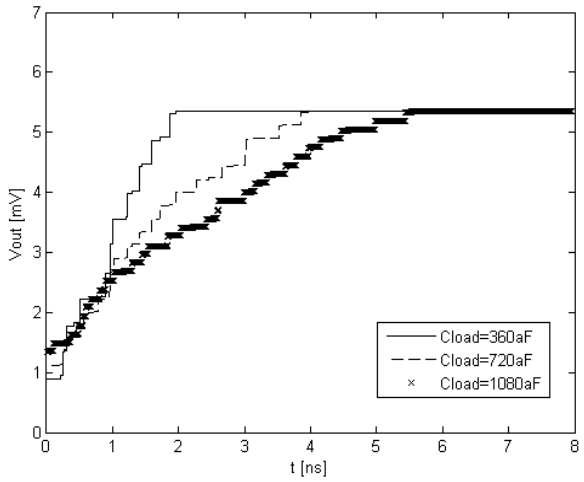


Figure 3.8. The transient step response of *Tucker's* inverter (Fig. 3.3) to an ideal input signal using Monte Carlo simulation.

Therefore, one usually needs to run MC simulation many times and take the average of all the results [57]. With an increasing number (k) of MC trials, the law of large numbers [13] shows that the sample average T_{avg} will converge to the expected value of t_N, i.e.,

$$\lim_{k\to\infty}\{T_{avg} = \frac{\sum_{j=1}^{k} t_{N,j}}{k}\} = \mu = \sum_{i=1}^{N} \frac{1}{\Gamma_i} \tag{3.12}$$

This can be interpreted as follows: after running a large number of MC trials, there will be a very high probability that the average delay is getting close to its expected value μ, which corresponds to 50% error probability. However, this provides less useful information about the dynamic behavior of SET circuits.

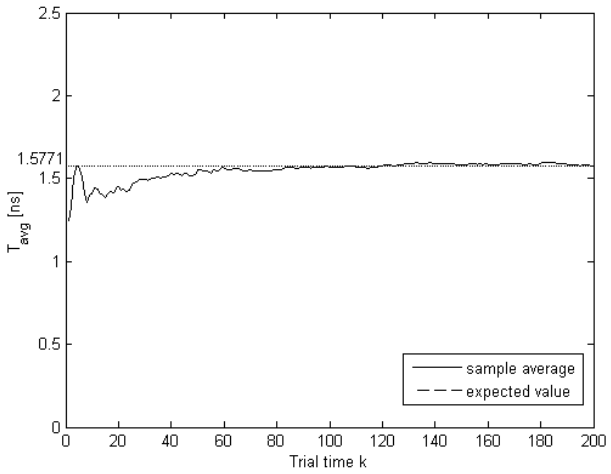


Figure 3.9. The sample average delay T_{avg} versus MC trial time k for the *Tucker's* inverter of Fig. 3.3.

The transient results of the *Tucker's* inverter given by the MC are shown in Fig. 3.8, where the load capacitance is chosen to be 360aF, 720aF, and 1080aF. Figure 3.9 shows the average delay from multiple MC simulations for the inverter with C_{load} = 360aF. As the number of trials k increases, the average delay T_{avg} converges to μ which, in this particular case, is around 1.57ns (also refer to the data cursor in Fig. 3.4(a)). Therefore, the delay results from MC simulation should be understood as follows: the desired electron transport occurs before T_{avg} with 50% error probability.

The accurate PDF of t_N is calculated as the derivative of $P_N(t)$ in (3.3), as shown in Fig. 3.10. For comparison, I also plot in the figure a histogram out of the data obtained by multiple runs of the MC simulation, which agrees well with the accurate PDF. It can be seen from the figure that the error probability for different delay can be estimated by using MC results. For instance, given a specific epoch T_N, the $P_{error}(T_N)$ can be calculated as the percentage of MC trials whose delay is larger than T_N. More MC trials will promise a better estimation.

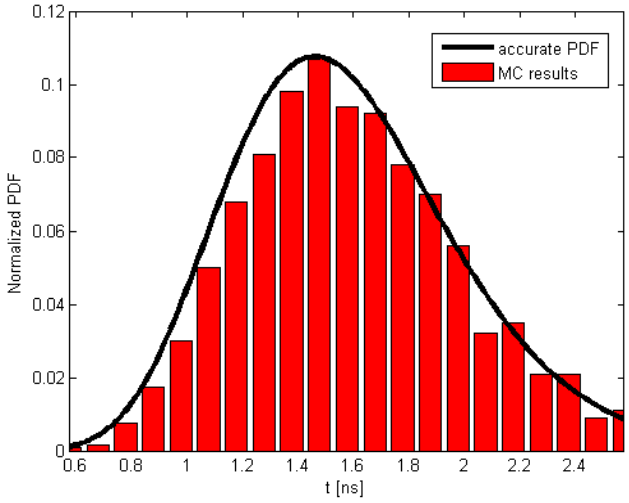


Figure 3.10. The accurate PDF from the ME (solid line), and the statistical histogram obtained by multiple runs of MC simulations on the inverter of Fig. 3.3.

A major disadvantage of MC method is that it would be very time-consuming when simulating some rare processes such as co-tunneling or thermal tunneling [52]. Nevertheless, for some cases where only stationary characteristics of a circuit are of interest, the MC simulation can give good results within a reasonable time. More results on circuit delay by the MC method can be found in [57] and [59].

3.4.3 Macro-modeling method

The main requirement of the macro-modeling method is that the interconnection capacitances have to be large enough to guarantee that SEDs are independent circuit elements. While dealing with a large circuit, one can simulate each isolated SED based on the Master Equation, and then take the interconnections into account using *Kirchhoff's* law. This would significantly reduce the computation complexity during simulation.

For example, the MIB model [47] is one of those macro-models that are widely used for single-electron transistors. In the following, I briefly describe how this model is used for delay analysis with the aforementioned *Tucker's* inverter. The basic idea is to model the SET transistor as a voltage-controlled current source: $I_{DS} = f(V_D, V_S, V_G)$, which means the drain-source current I_{DS} is determined by the voltage at three terminals. Assume the input voltage V_G is ideal (i.e., $V_G(t=0^-) = 6.67mV$, $V_G(t=0^+) = 0V$). During the output switching from logic "0" to logic "1", $V_D = V_{dd}$ is a constant, and $V_S = V_O$. Thus, I_{DS} only depends on the output voltage, i.e., $I_{DS} = f(V_D, V_S,$

V_G) = $I_{DS}(V_O)$. The switching delay is due to the load capacitance charged by a time-dependent current I_{DS}. In other words, we have:

$$\frac{dV_O}{dt} = \frac{I_{DS}(V_O)}{C_{load}} \Rightarrow T_{MIB} = C_{load} \cdot \int_{V_O(0)}^{V_O(0)+0.9V_{Swing}} \frac{dV_O}{I_{DS}(V_O)} \tag{3.13}$$

Since the above integration over V_O turns out to be a constant, the estimated delay from the macro-modeling is simply proportional to the load capacitance. This may provide us with some qualitative results for comparative study, but is not an accurate quantitative description. The reason behind this is that the MIB macro-model is derived from the steady state ME ($\partial p_i/\partial t=0$), and hence cannot be used generally for simulating dynamic behaviors (unless in some special cases – see [63, 98] for further details). Again, I use the *Tucker's* inverter of Fig.3.3 as an example, and plot in Fig. 3.11 the output transient response using the MIB model with different load capacitances. Fig. 3.12 shows the comparison of delay estimated by the macro-model simulation and our proposed method. It is worth noting that the macro-model treats the delay as a deterministic value (rather than a stochastic variable) in the sense that the circuit is always in a quasi-steady state, and the delay can be simply calculated from (3.13). With our statistical method, however, different error probabilities lead to different delays, as shown in Fig. 3.12.

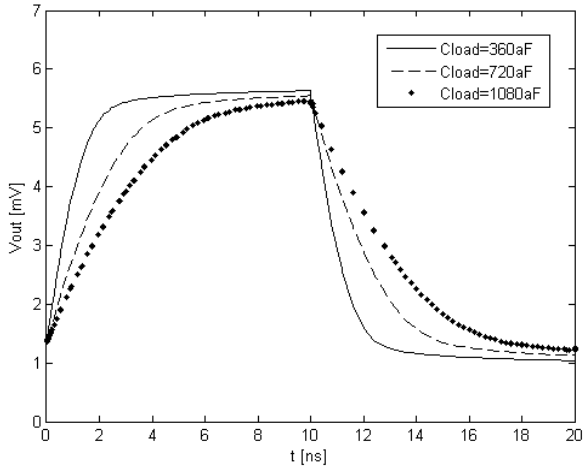


Figure 3.11. The transient response of *Tucker's* inverter to an ideal input signal using the MIB macro-model.

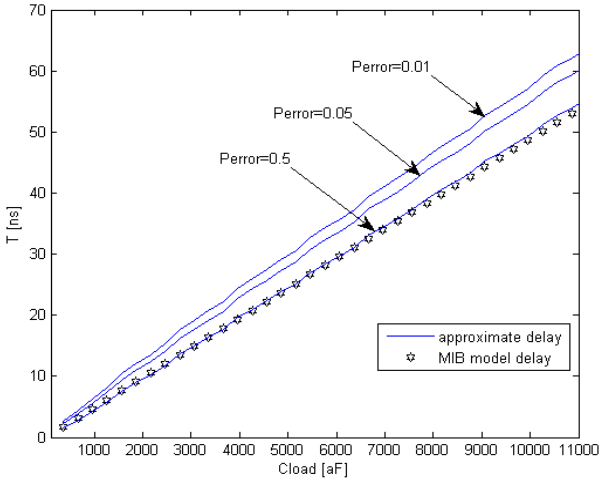


Figure 3.12. The simulated delay versus load capacitance for *Tucker*'s inverter (Fig. 3.3) using the MIB macro-model (hexagons) and the proposed method (solid lines) for $P_{error,\,N}$ = 0.5, 0.05, and 0.01.

Some recent models [64, 65] for SET transistors claim that they can also conduct the transient process simulation by treating the SET transistor as a current source and solving the time-dependent Master Equation. However, as electron tunnels are assumed to occur instantaneously, the current cannot be captured properly during state transitions. Thus, using a deterministic steady-current will not fully characterize the transient transition which is a stochastic process. Another difficulty with the macro-modeling is that the derivative of tunnel rate Γ is discontinuous near the threshold of Coulomb blockade (especially at a low temperature). This may create some convergence issues with traditional circuit simulators (e.g., SPICE). Nevertheless, the macro-modeling seems to be the only efficient simulation method for large circuits (especially for SET/MOS hybrid circuits) [47, 98].

3.4.4 The step estimation method

This section is a review and extension of the work reported in [66], where the so-called step estimation method was introduced to solve (3.3) in an approximate way using the concept of *equivalent* tunnel rate Γ_{eq}. Consider a circuit with N tunnel events as depicted in Fig. 3.1. For a specific value of error probability $P_{error,N}$, the corresponding delay T_{eq} is approximated as the delay of another circuit with only a single event whose tunnel rate is Γ_{eq}, i.e.,

$$P_{error,N}(T_{eq}) = \sum_{i=1}^{N}\left(\prod_{j=1,\,j\neq i}^{N}\frac{\Gamma_j}{\Gamma_j-\Gamma_i}\right)\cdot\exp(-\Gamma_i T_{eq}) = \exp(-\Gamma_{eq}T_{eq}) \tag{3.14}$$

To find Γ_{eq}, two adjacent tunnel events in the original circuit are considered at a time, and their tunnel rates are combined to obtain an equivalent one which is then combined with the next tunnel rate in the next step of calculation. The process continues with Γ_{eq} being updated until all tunnel rates are taken [66]. Once the equivalent tunnel rate Γ_{eq} is known, one can easily obtain the delay T_{eq} from (3.14). However, in order to comprehensively describe the circuit's dynamic characteristics, the delay under different values of $P_{error,\,N}$ has to be estimated, and hence Γ_{eq} needs to be calculated repeatedly.

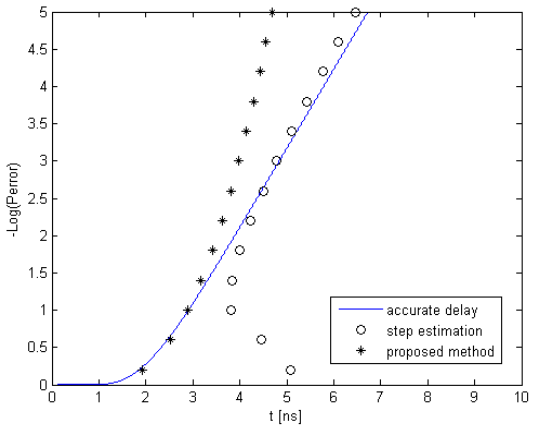


Figure 3.13. The comparison of delay estimation results for different error probabilities using the accurate ME method (solid line) from (3.3), step estimation method (circles), and the proposed statistical method (stars).

Again, using the *Tucker's* inverter as an example, we plot in Fig. 3.13 the step estimation results for different error probabilities in comparison with the accurate solution of (3.3) and the proposed statistical method. It can be seen from the figure that the step estimation is quite accurate for small error probabilities ($P_{error,\,N} < 10^{-2}$), but it fails to give reasonable delay estimation as $P_{error,\,N}$ becomes larger (say, $P_{error,\,N} \sim 0.1$). For the proposed method, the opposite is true: it has good accuracy for relatively large $P_{error,\,N}$ ($P_{error,\,N} > 0.01$), but poor performance with a small value of $P_{error,\,N}$. This suggests that the combination of step estimation and the proposed method offers a fast and accurate dynamic behavior analysis, depending on the value of $P_{error,\,N}$.

3.5 Discussion

In this section, I first discuss the temperature effect with our delay model. It is found from simulations that under certain assumptions, the proposed model only needs to be modified slightly

in order to consider the operation at relatively high temperatures. Then, we show the temperature effects with some existing methods. The delay for non-ideal inputs as well as multi-stage circuits is also briefly studied.

3.5.1 The temperature effect

In the previous sections, we assume that the temperature is extremely low (T~0K), which means that electrons can tunnel through junctions only when the tunneling reduces the system's free energy. In this case, electrons can only tunnel in one direction, and the system evolves monotonously. However, as the temperature increases, electron tunnel events can take place even if they lead to an increase in free energy. This implies that bidirectional electron tunnelings have to be considered. The thermal energy may introduce the so-called inverse tunneling that takes the system back to previous states and lengthens the logic transition process. On the other hand, the tunnel rates will increase under a high temperature (according to the orthodox theory), which tends to reduce the delay time. Therefore, the overall impacts of the temperature on the switching delay make the circuit's dynamic behavior noisier and more difficult to analyze. In what follows, we show that the delay analysis procedure can be simplified significantly under some reasonable assumptions.

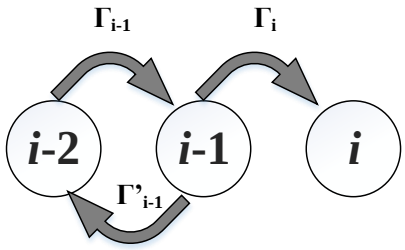


Figure 3.14. The simplified state transition diagram.

Considering the fact that SEDs traditionally work under a relatively-low temperature to enable the Coulomb blockade effect, we assume that the inverse tunneling due to the thermal energy cannot take place consecutively. For instance, if the system is on state $i-1$ at present, then it can only transfer back to state $i-2$ before moving forward to state i (this is called the 1st-order inverse tunneling assumption in this work). The Δt_i shall be defined as the time interval from the time the system arrives at state $i-1$ and the time it reaches state i (both for the first time). In order to obtain the time-dependent distribution given an initial condition that the system is on state $i-1$ at $t = 0$,

we look at the distribution of the delay for the system to arrive at state *i* for the first time (Δt_i). The state transition diagram is shown in Fig. 3.14, and its Fokker-Plank description [126] is written as:

$$Q = \begin{bmatrix} -\Gamma_{i-1} & \Gamma_{i-1} & 0 \\ \Gamma'_{i-1} & -(\Gamma'_{i-1}+\Gamma_i) & \Gamma_i \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{cases} \dfrac{d}{dt}[P_{i-2}(t), P_{i-1}(t), P_i(t)] = [P_{i-2}(t), P_{i-1}(t), P_i(t)] \cdot Q \\ [P_{i-2}(0), P_{i-1}(0), P_i(0)] = [0,1,0] \end{cases} \tag{3.15}$$

The PDF of Δt_i can be expressed as $d[P_i(t)]/dt$. While this PDF does not follow the exponential distribution as in the case of zero temperature (as shown in (3.5)), only its mean value and variance are required for our statistical analysis. In other words, (3.6) still applies regardless of the distributions of each variable Δt_i. Solving (3.15) leads to the expected value and variance as follows:

$$E(\Delta t_i) = \int_0^{+\infty} [\frac{dP_i(t)}{dt} \cdot t] dt = \frac{1}{\Gamma_i} \cdot (1 + \frac{\Gamma'_{i-1}}{\Gamma_{i-1}})$$

$$D(\Delta t_i) = \int_0^{+\infty} [\frac{dP_i(t)}{dt} \cdot t^2] dt - [E(\Delta t_i)]^2 = \frac{1}{\Gamma_i^2} \cdot [(1 + \frac{\Gamma'_{i-1}}{\Gamma_{i-1}})^2 + \frac{2\Gamma'_{i-1}\Gamma_i}{\Gamma_{i-1}^2}] \tag{3.16}$$

It can be seen from (3.16) that due to a high temperature raises both expected value and variance of Δt_i by a factor that is related to the ratio of backward and forward tunnel rates (i.e., Γ'_{i-1} / Γ_i). This ratio depends on the temperature as well as the voltage across tunnel junctions, and becomes greater when either the temperature goes higher or a lower voltage is applied across the junction. Under low temperature, this ratio has extremely low value since in general its value has an exponential relation with temperature. In this case, the inverse tunnel rates are usually several orders of magnitude smaller than the desired tunnel ones, making their effects can be ignored. However, as temperature becomes relative high, those inverse tunnels could take place more often especially when the voltage across is low. For instance, for the inverter example at T = 40K, at the beginning of the logic transition this ratio has the order of magnitude around 10^{-3} since the inverse tunnels are suppressed by the relative high voltage across junctions; and the ratio value increases to ~0.20 at the end where the voltage is close to threshold of electron tunnel.

If the obtained new expected value and variance are used to substitute original ones in (3.6), the proposed delay estimation method can be applied without any further modification. Fig. 3.15

plots the estimated delay of aforementioned inverter under different temperatures using (3.15) and (3.16) against the accurate result.

It can be seen from the figures that with our 1st-order inverse tunneling approximation, the proposed delay model can capture the circuit dynamic characteristics very accurately at relatively low temperatures ($T \leq 50K$), as shown in Fig. 3.15(a) and (b). At $T = 70K$ (see Fig. 3.15(c)), the result from the proposed model become less accurate. This suggests that if the temperature further increases, the 2nd-order inverse tunneling need to be taken into account. When considering one more consecutive inverse tunneling (which is called the 2nd-order assumption), the mean and variance of Δt_i can be computed similarly as follows:

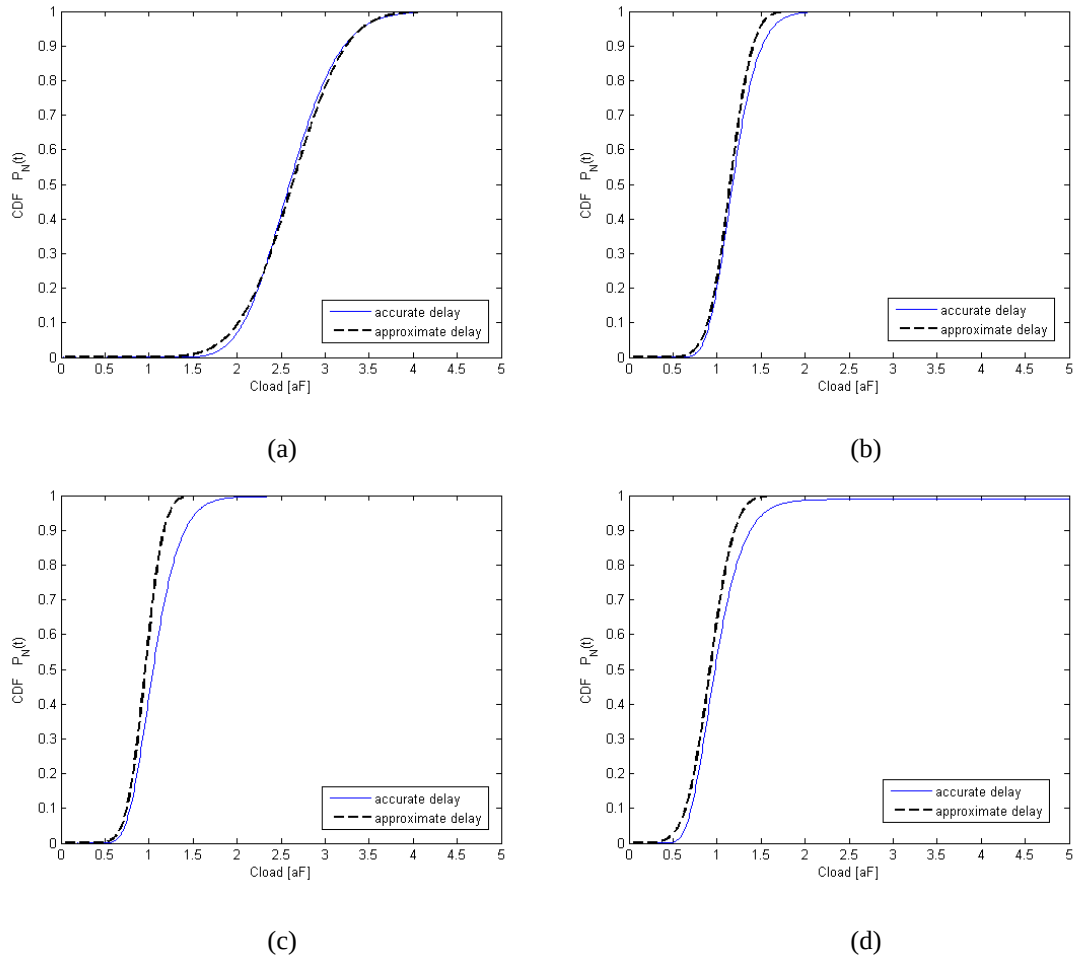
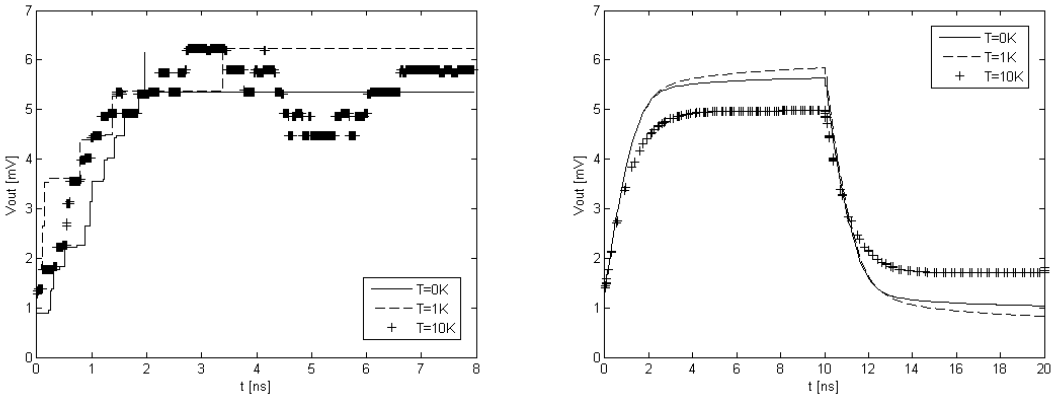


Figure 3.15. The performance of proposed delay estimation method under different temperatures: (a) 10K, (b) 50K, (c) 70K (with 1st-order inverse tunneling assumption), and (d) 90K (with 2nd-order inverse tunneling assumption). The other circuit parameters are the same as in the previous inverter example.

$$E(\Delta t_i) = \frac{1}{\Gamma_i} \cdot [\frac{\Gamma'_{i-1}}{\Gamma_{i-1}} \cdot (1 + \frac{\Gamma'_{i-2}}{\Gamma_{i-2}}) + 1]$$

$$D(\Delta t_i) = \frac{1}{\Gamma_i^2} \cdot [(1 + \frac{\Gamma'_{i-1}}{\Gamma_{i-1}} \cdot (1 + \frac{\Gamma'_{i-2}}{\Gamma_{i-2}}))^2 + \frac{2\Gamma'_{i-1}\Gamma_i}{\Gamma_{i-1}^2}((1 + \frac{\Gamma'_{i-2}}{\Gamma_{i-2}})^2 + \frac{\Gamma'_{i-2}\Gamma_{i-1}}{\Gamma_{i-2}^2})] \quad (3.17)$$

The performance of the delay model using 2nd-order assumption is shown in Fig. 3.15(d), which gives a good accuracy even under T = 90K. Generally speaking, in order to achieve good accuracy at high temperatures, a higher order inverse tunneling has to be included. However, a relatively low temperature condition is commonly applied for SEDs to avoid the thermal fluctuation. Thus, our 1st- and 2nd-order assumptions are usually sufficient in most cases.



(a) (b)

Figure 3.16. Temperature effects on Fig. 3.3 with C_{load} = 360aF using the MC simulation (a) and MIB model (b).

We also took a look at the transient response using two existing simulation methods (i.e., MC simulation and macro-modeling method). Fig. 3.16 shows the simulation results of Fig. 3.3 with T = 0K, 1K, 10K using the MC simulation and MIB model method. It can be seen that the two methods respond to the temperature effect differently. The MC method is able to depict the transient process of the circuit in a very straightforward way, while the macro-modeling method is more efficient. The results of the latter are only of relative accuracy, which is useful when simulating large-scale circuits for delay comparison. What can also be seen from the figure is the fact that as the temperature increases, so does the number of states that the circuit may occupy. When T = 0K, it takes the total *N* tunnel events for the circuit to finish the logic transition. However, with increasing temperature, the circuit will no longer stabilize at state *N*. For instance,

the MC results of Fig. 3.16 (a) show that the final voltage with T > 0K is higher than that for T = 0K. Similar results can also be seen in Fig. 3.16 (b) with the macro-modeling method. Moreover, as the temperature climbs to 10K, the circuit will not stabilize at a single state, but at several states instead (refer to Fig. 3.16 (a)). In Fig. 3.16(b), a lower voltage swing is obtained for T = 10K, which represents an ensemble average of multiple states. However, the transient response plotted by the macro model may not make much sense under a high temperature, for which the steady-state ME ($\partial p_i/\partial t=0$) cannot be satisfied.

It should be noticed that the higher tunnel rates with the thermal energy tend to make the logic transition faster. As a result, the switching delay decreases monotonously from T = 10K to 90K, as can be seen from Fig. 3.15. This has inspired the designers to utilize the thermal energy for improved circuit performance (i.e., with larger output swing and less delay). The idea of exploiting the thermal energy in SET circuits was presented for the first time by *Oya* [100, 101], in which a circuit system (so-called stochastic resonance system) was designed using a model of noise-exploiting neural networks. Since the temperature effect is usually a big concern in SET circuits, taking advantage of thermal energy could be an interesting topic for future research.

3.5.2 Non-ideal inputs and multi-stage circuit

In previous discussions, we assume an ideal input voltage with zero rise- or fall-time. To investigate how the transition time of input signals may affect the switching delay at the output, I used three methods (i.e., MC, ME and macro-modeling methods) for the *Tucker's* inverter with the input voltage's fall time being 0ns, 1ns and 2ns. The simulation results with MC, ME and macro-modeling methods are shown in Figs. 3.17, 3.18 and 3.19, respectively. It can be seen from these figures that an increased fall-time will generate a longer switching delay, as expected. Again, while the ME results are accurate, the MC simulation is statistically reliable, and the macro-modeling makes sense only from a perspective of relative accuracy.

In a multi-stage logic circuit, the delay of its longest path (or the path consisting of gates with a smaller value of tunnel rate) generally gives the overall delay. However, the delay of individual gates creates a non-ideal input signal for their fanout gates. Therefore, an alternative method to analyze the circuit delay is to characterize this non-ideal signal as a tunnel event whose tunnel rate corresponds to its rise- for fall-time, and then use some kind of combination of the step estimation method and proposed statistical method. Further research work is needed to deal with this issue.

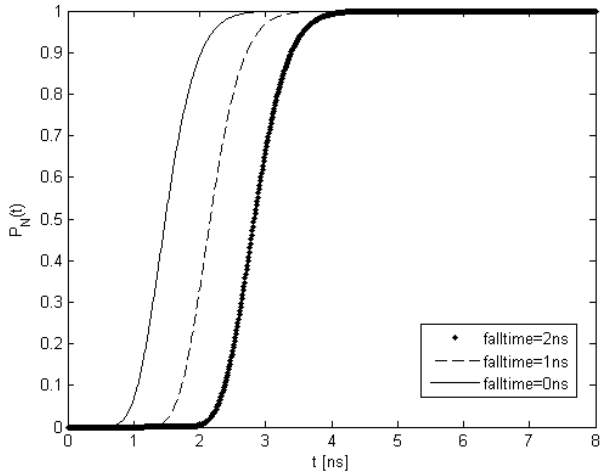


Figure 3.17. The CDF of $P_N(t)$ for Fig. 3.3 using the solution of Master Equation assuming the input signal's fall-time of 0ns, 1ns, and 2ns.

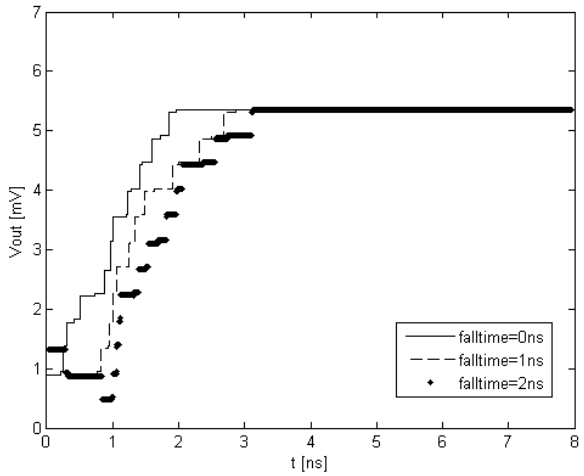


Figure 3.18. The transient response of Fig. 3.3 with ideal or non-ideal input signals using the MC simulation.

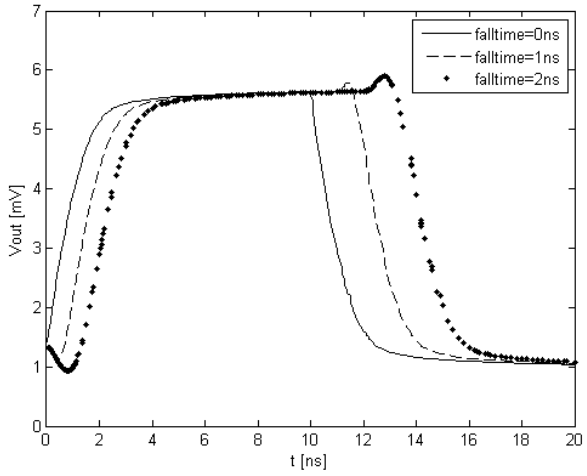


Figure 3.19. The transient response of Fig. 3.3 with ideal or non-ideal input signals using the MIB macro-model.

3.6 Summary

In this Chapter the stochastic nature of SET circuits has been studied with the proposed statistical delay modeling. It has been demonstrated that the switching delay can be approximated by the normal distribution for quick estimation, even at a relatively-high temperature. The advantages of the proposed method have been shown using typical logic gates as an example. A comparative study on existing methods for delay estimation has been conducted, which helps designers better understand the obtained simulation results. Effects of temperature and non-ideal input signals on the delay of SET circuits have also been briefly discussed.

CHAPTER 4

RELIABILITY ANALYSIS OF COMBINATION LOGIC CIRCUITS

4.1 Background

As CMOS technology further scales down, circuit designers are facing new challenges that were not much critical in the past, such as quantum effects, large power dissipation, and low reliability [145]. In particular, reliability has become one of the increasingly serious issues, partly due to low voltage/current threshold, electromigration, and process variations (such as power supply instability and device mismatch). As CMOS devices reach their fundamental physical limits, on the other hand, non-conventional nanometer-scale electronic components and sophisticated architectures/technologies have been studied and fabricated, as we have mentioned in previous chapters. These devices typically require low-temperature operation, are more sensitive to a variety of random noises, and thus are statistically less reliable than their CMOS counterparts [155]. This has led to significant interests in reliability analysis, and motivated the investigation of reliability-oriented architectures using unreliable components for circuit design.

Reliability analysis for combinational circuits is computationally expensive in general, due to an exponentially growing number of input patterns as well as possible signal correlations involved. For instance, some analytical methods have been developed, but they seem to be unrealistic even for medium-size circuits. It is believed that the task of determining exactly the output reliabilities for arbitrary logic circuits can't be solved within polynomial time. When analytical methods become less practical, an alternative solution is to either use statistical approaches (such as Monte-Carlo simulation), or resort to heuristic algorithms in order to make a reasonable tradeoff between the accuracy and efficiency. A brief review of previous works on reliability analysis will be given in next section.

In this Chapter I presents a fast and recursive model for reliability analysis using the concept of equivalent reliability, which is based on the observation that circuit output reliability is a result of the cumulative effects of all unreliable devices within the circuit. When it comes to any specific gate, an equivalent reliability can be used for its output to account for the effects of all errors caused by the gate itself as well as its transitive fan-in cone. By calculating equivalent reliabilities recursively gate by gate throughout the circuit, the whole procedure of reliability analysis can be done in a more efficient way. This equivalent model is exact for circuits without signal correlations. For circuits containing reconvergent fan-outs, the proposed model captures the

signal and reliability correlations in the original circuit by using correlation coefficients from its error-free version. Our simulation results show that the proposed approach provides a significant speedup over Monte-Carlo simulation, and more accurate results than other existing methods.

The remainder of this Chapter is organized as follows. Section 4.2 presents a background introduction of reliability analysis and some related works. Section 4.3 describes the proposed model in details. Section 4.4 shows simulation results with a comparative study. I give discussions with possible future work in Section 4.5, and conclude the paper in Section 4.6.

4.2 Prior work

Reliability of a logic signal is defined as the probability that its value is correct (or equals to its error-free value). The signal may become unreliable due to the errors of its driving gate and/or input signals. If we use the classical *von Neumann* model [157] for gate errors, each gate can be associated independently with an error probability ε (throughout the paper, the terms "error" and "failure" will be used interchangeably). In other words, a gate is modeled as a binary symmetric channel, which can generate a bit flip (from 0→1 or 1→0) at its output (known as *von Neumann* error) symmetrically with same error probability [154]. The resulting errors are mainly due to random noises and temporary environmental influences, rather than permanent physical defects. In physic level, there are many different sources of noise, such as crosstalk, thermal noises, and cosmic radiations. In order to represent these effects in electric level, a nominal voltage can be used, whose value is traditionally modeled by a *Gaussian* distribution. Thus, a gate failure probability can be defined as the probability that its nominal voltage exceeds a certain threshold value (i.e., noise margin). In the real world, every gate i in the circuit has an independent error probability ε_i (or gate reliability $r_i = 1 - \varepsilon_i$), which is assumed to be localized and statistically stable. Also, it is assumed that any gate failure probability (or failure rate) is a constant within [0, 0.5] (or, $r_i \in [0.5, 1]$) regardless of its input signal values, while this may not be totally true in some nonconventional circuits [155]. This assumption is just for convenience of discussion, and our approach also applies to the cases where gate failure probabilities do depend on its input logic values, as will become clear later in the paper.

For any signal inside a circuit, we can associate it with a reliability pair $\{r^0, r^1\}$, depending on the error-free value ("0" or "1") of the signal, where r^0 (or r^1) represents a conditional probability of the signal being "0" (or "1") given its error-free value is "0" (or "1"). In other words, for a specific signal s, $r_s^0 = P\{s = s^* \mid s^* = 0\}$ and $r_s^1 = P\{s = s^* \mid s^* = \text{"1"}\}$, where s^* is the error-free version of s (for the remainder of the paper, the symbol "*" is used to indicate "error-free" when

referring to signals, and the terms "error-free", "reliable" and "correct" are used interchangeably). However, primary input signals are assumed to be reliable (i.e., their reliability is 1), and their probabilities are independent of each other. In this work, the probability of signal *s* is by default defined as the probability of the signal being logic "1", and is expressed as P_s or $P\{s = \text{"1"}\}$. The probability of *s* being "0" is denoted by $P\{s = \text{"0"}\}$, which is equal to $1-P_s$. Also, the "0" and "1" represent logic values, and hence the quotation marks are omitted for brevity (unless otherwise specified).

The problem of reliability analysis for combinational circuits is stated briefly as follows: For given probabilities of primary inputs and individual gate reliabilities in the circuit, find the reliabilities of individual primary outputs, or more specifically, the reliability pair $\{r_j^0, r_j^1\}$ for each primary output F_j (j = 1, 2, …, m, where m is the number of primary outputs), where r_j^0 (or r_j^1) denotes a conditional probability that the j-th output is logic 0 (or 1) when its error-free value is meant to be 0 (or 1). Once r_j^0 and r_j^1 are found, the average reliability for this output is given by

$$r_j = P_j^* r_j^1 + (1-P_j^*) r_j^0 \tag{4.1}$$

where P_j^* is the probability of primary output F_j (or, more exactly, F_j^*) when the circuit is error-free. The actual probability of the output (when the circuit is not error-free) can be calculated as

$$P_j = P_j^* r_j^1 + (1-P_j^*)(1-r_j^0) \tag{4.2}$$

As a typical statistical method, Monte-Carlo simulation [161] has been widely used for reliability evaluation. However, a large number of simulation runs are statistically required to reach a stable result. It typically takes up to hours for the MC method to obtain good results, depending on circuit size. While improvements in efficiency can be made using some speedup procedure (such as non-Bernoulli sequences introduced in [162]), the computation time still grows exponentially with circuit size. Another drawback with simulation-based methods is lack of flexibility, since the simulation process needs to be repeated for any changes in gate error probability.

Recently, some analytical methods for reliability calculation have been proposed, including techniques using probabilistic gate models (PGMs) [145], probability transfer matrices (PTMs) [147], Bayesian network [131], and Boolean difference-based error calculator (BDEC) [150]. The PGM method works perfectly for small circuits or correlation-free circuits. The PTM and Bayesian network techniques can provide accurate results, but remain computation-intensive for large circuits with signal correlation, due to an exponential runtime with the number of

reconvergent fanouts or the demand for prohibitively-huge data storage in either probability transfer matrices or conditional probability tables. Since these methods try to exhaustively calculate joint probability distributions in the circuit, they are considered as brute-force ways to solve an NP-hard problem. The BDEC is a fast gate-level probability error propagation model, where only local reconvergent fanouts are considered by the so-called level collapsing.

Many other heuristic methods for reliability analysis have also been investigated with various assumptions and/or approximations. The observability-based and single-pass reliability analysis presented in [154] is such an example. The observability-based approach tries to provide a closed-form expression for circuit reliability by using observability metrics to quantify the impact of a gate failure on the circuit output, but is only suitable for small circuits and for high gate reliabilities in large circuits. The single-pass reliability analysis algorithm, on the other hand, is provably exact for circuits without reconvergent fan-out, and uses high-order correlation coefficients to handle correlation effects. Unfortunately, its computational complexity increases with the number of correlation coefficients. In [152, 153], some hybrid methods were also investigated by considering the combination of exact analysis with probabilistic measures. However, these approaches suffer from unacceptably high errors in estimating the circuit reliability, as the correlation of signal probabilities and/or reliabilities within the circuit is not well captured.

4.3 Method

The proposed reliability model is based on the concept of *equivalent reliability* (ER), and hence named the ER model. The main idea is to calculate the equivalent reliability at the output of a specific gate using the probability and reliability information (including the correlations involved) from its fanin (input) signals as well as the reliability of the gate itself. The obtained equivalent reliabilities are then propagated to next level of gates in a recursive fashion. This propagation process starts from primary inputs, takes one gate at a time in a topological order, and stops at primary outputs. Thus, the reliabilities at primary outputs reflect the cumulative effects of all unreliable gates on the propagation path (i.e., in their transitive fan-in cones). In this Section, I first introduce the concept of equivalent reliability, and then discuss details of the proposed reliability analysis model along with some examples.

4.3.1 Equivalent reliability (ER)

Definition 1 (*dummy buffer*): *A dummy buffer* is a single-input single-output gate which is associated with a reliability pair $\{r_{eq}^{0}, r_{eq}^{1}\}$, where r_{eq}^{0} (or r_{eq}^{1}) denotes a conditional probability

that the buffer's output is logic 0 (or 1) given its input being logic 0 (or 1). A reliable dummy buffer is a dummy buffer whose output is always equal to the input with $r_{eq}^{0} = r_{eq}^{1} = 1$.

Consider an *n*-input single-output combinational circuit with *N* logic gates, whose reliabilities are represented by $r_i \leq 1$ (i = 1, 2, …, N), as shown in Fig. 4.1(a). We construct its equivalent structure by using a companion error-free circuit followed by a dummy buffer, as shown in Fig. 1(b). This error-free circuit has a same topological structure as the original circuit, but all gate reliabilities are set to 1. The reliability pair $\{r_{eq}^{0}, r_{eq}^{1}\}$ of the dummy buffer represents the equivalent reliability pair for the output *F*, and hence reflects the cumulative effects of all errors within the original circuit on the output reliability. The signal F^{*} in Fig. 4.1(b) corresponds to error-free output signal. Generally speaking, the output *F* can be just an internal signal within the circuit. In this case, the circuit of Fig. 4.1 (a) simply corresponds to the transitive-fanin cone of *F*.

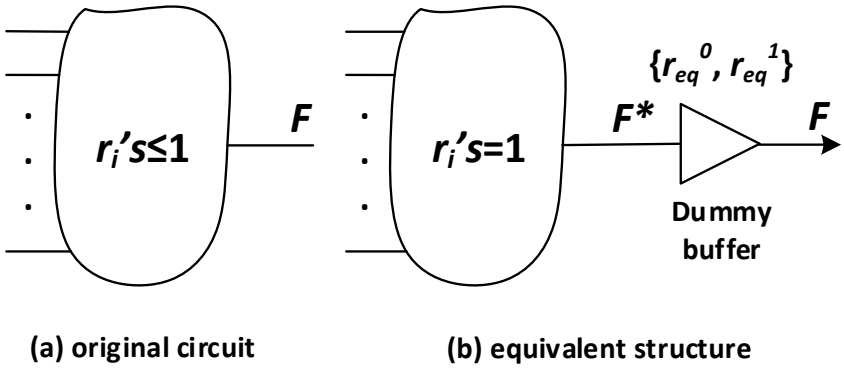
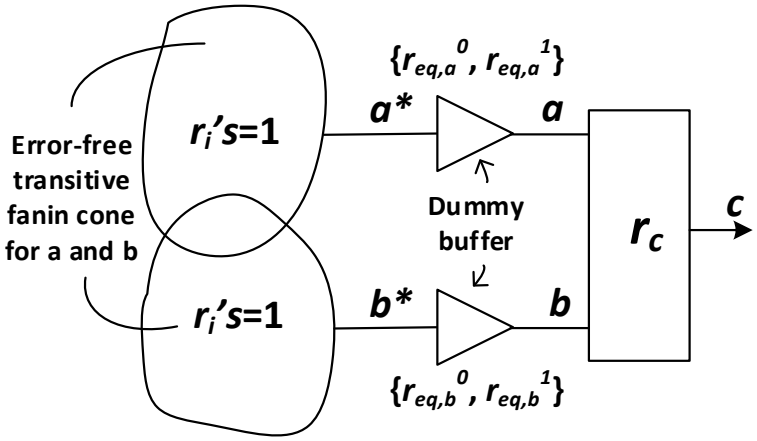


Figure 4.1. Combinational circuit and its equivalent structure.

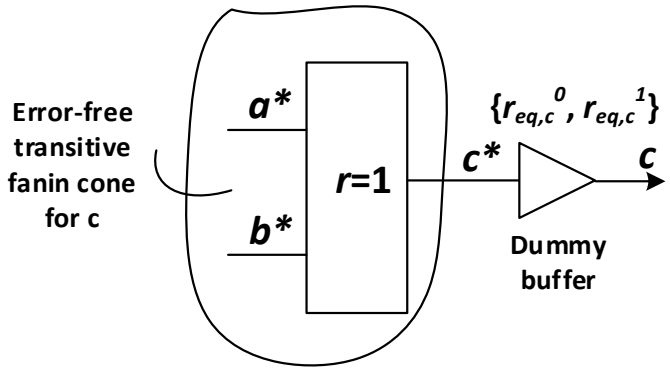
Finding exact values for r_{eq}^{0} and r_{eq}^{1} in Fig. 4.1 can be computationally expensive and difficult because: (a) they depend on primary input probabilities, and generally require exponentially-growing computation time which would be prohibitive as the number of inputs (*n*) becomes large, (b) the signals within the circuit may be correlated, and/or (c) the reliabilities at some internal nodes may also be correlated with each other. However, if we can develop a core computational step to enable the propagation of equivalent reliabilities for internal signals by applying it recursively to all gates throughout the circuit, then it will significantly speed up the process of computation. Once $\{r_{eq}^{0}, r_{eq}^{1}\}$ is available, the output reliability and probability can be found according to (4.1) and (4.2) where r_j^{0} and r_j^{1} shall be replaced by r_{eq}^{0} and r_{eq}^{1}, respectively. Details on the reliability propagation are described below.

4.3.2 Propagation of Probability and Reliability

Consider a 2-input gate with reliability of r_c, as shown in Fig. 4.2(a), where $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ denote the equivalent reliability pairs for both inputs a and b, respectively. The equivalent structure of Fig. 4.2(a) is shown in Fig. 4.2(b), where $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ is the equivalent reliability pair for the output c. Note that a^*, b^* and c^* in the figure denote the error-free signals, while a, b and c represent the actual signals with errors. I show below how to find $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ for given probabilities of a^*, b^* and c^* (i.e., P_{a*}, P_{b*} and P_{c*}) by considering the following two cases: i) when $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ are independent, and ii) when $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ are correlated.



(a) circuit before propagation



(b) equivalent circuit after propagation

Figure 4.2 Probability and reliability propagation for 2-input gate.

A. Independent Case

When the inputs *a* and *b* are independent, their equivalent reliabilities would be independent as well. However, the independence of equivalent reliabilities does not necessarily mean that their corresponding signals are uncorrelated. For instance, if all gates/signals shared by the transitive-fanin cones of both inputs are error-free, the equivalent reliability pair for one input will not depend on that for the other, while both inputs are still correlated. Without loss of generality, we assume, in the following, the independence of equivalent reliabilities with signal correlation.

Definition 2 (*input probability vectors*): For two input signals *a* and *b* of a logic gate, as shown in Fig. 4.2(a), *the error-free input probability vector* $\mathbf{P}^*$ is defined as a 1×4 matrix where each element represents a joint probability of error-free signals a^* and b^*, i.e.,

$$\mathbf{P}^* = [P_{00}^* \quad P_{01}^* \quad P_{10}^* \quad P_{11}^*] = [P\{a^*b^* = 00\} \quad P\{a^*b^* = 01\} \quad P\{a^*b^* = 10\} \quad P\{a^*b^* = 11\}] \tag{4.3}$$

and *the actual input probability vector* **P** is defined similarly for actual signals *a* and *b*, i.e.,

$$\mathbf{P} = [P_{00} \quad P_{01} \quad P_{10} \quad P_{11}] = [P\{ab = 00\} \quad P\{ab = 01\} \quad P\{ab = 10\} \quad P\{ab = 11\}] \tag{4.4}$$

where $P\{a^*b^* = ij\}$ (or $P\{ab = ij\}$) is a joint probability for a^* (or *a*) =*i* and b^* (or *b*) =*j* with *i*, *j* = 0 or 1. In other words, $P\{a^*b^* = ij\} = P\{a^* = i \mid b^* = j\} \cdot P(b^* = j)$, and $P\{ab = ij\} = P\{a = i \mid b = j\} \cdot P(b = j)$.

The relationship between **P*** and **P** can be described as:

$$\mathbf{P} = \mathbf{P}^* \cdot \mathbf{M} \tag{4.5}$$

where **M** is a 4×4 probability transfer matrix given by

$$\mathbf{M} = \begin{bmatrix} p_{00}^{00} & p_{00}^{01} & p_{00}^{10} & p_{00}^{11} \\ p_{01}^{00} & p_{01}^{01} & p_{01}^{10} & p_{01}^{11} \\ p_{10}^{00} & p_{10}^{01} & p_{10}^{10} & p_{10}^{11} \\ p_{11}^{00} & p_{11}^{01} & p_{11}^{10} & p_{11}^{11} \end{bmatrix} \tag{4.6}$$

where each element p_{ij}^{kl} represents a conditional probability for $ab = kl$ given $a^*b^* = ij$, i.e., $p_{ij}^{kl} = P\{ab = kl \mid a^*b^* = ij\}$ with *i*, *j*, *k*, *l* = 0 or 1. Due to the independence of equivalent reliabilities for the two inputs (refer to Fig. 4.2(a)), (4.6) can be expressed as

$$\mathbf{M}_{ind} = \begin{bmatrix} r^0_{eq,a} r^0_{eq,b} & r^0_{eq,a}(1-r^0_{eq,b}) & (1-r^0_{eq,a}) r^0_{eq,b} & (1-r^0_{eq,a})(1-r^0_{eq,b}) \\ r^0_{eq,a}(1-r^1_{eq,b}) & r^0_{eq,a} r^1_{eq,b} & (1-r^0_{eq,a})(1-r^1_{eq,b}) & (1-r^0_{eq,a}) r^1_{eq,b} \\ (1-r^1_{eq,a}) r^0_{eq,b} & (1-r^1_{eq,a})(1-r^0_{eq,b}) & r^1_{eq,a} r^0_{eq,b} & r^1_{eq,a}(1-r^0_{eq,b}) \\ (1-r^1_{eq,a})(1-r^1_{eq,b}) & (1-r^1_{eq,a}) r^1_{eq,b} & r^1_{eq,a}(1-r^1_{eq,b}) & r^1_{eq,a} r^1_{eq,b} \end{bmatrix} \tag{4.7}$$

Under the special case when both inputs are fully reliable (i.e., $r_{eq,a}{}^0 = r_{eq,a}{}^1 = r_{eq,b}{}^0 = r_{eq,b}{}^1 = 1$), the matrix $\mathbf{M}_{ind}$ in (7) will become a unit matrix **I**, leading to $\mathbf{P} = \mathbf{P}^*$.

Definition 3 (*output reliability vector*): For a logic gate with two input signals *a* and *b* (refer to Fig. 4.2(a)), *the output reliability vector* **R** is a 4×1 matrix where each element is a conditional probability for its output *c* being a specific value k_0 given *ab* = 00, 01, 10 or 11, i.e.,

$$\boldsymbol{R} = [\boldsymbol{P}\{\boldsymbol{c} = k_0 \mid \boldsymbol{ab} = \boldsymbol{ij}\}] \quad \boldsymbol{i}, \boldsymbol{j} = \mathbf{0}, \mathbf{1} \tag{4.8}$$

where $k_0 = 0$ for AND, NOR and XNOR gates, and $k_0 = 1$ for NAND, OR and XOR gates.

Table 4.1 Output Reliability Vector **R** for Different Gates

Gate Type	**R**
AND	$[r_c \;\; r_c \;\; r_c \;\; 1-r_c]^T$
NAND	
OR	$[1-r_c \;\; r_c \;\; r_c \;\; r_c]^T$
NOR	
XOR	$[1-r_c \;\; r_c \;\; r_c \;\; 1-r_c]^T$
XNOR	

Table 4.1 shows the output reliability vector **R** for a variety of 2-input logic gates. In order to calculate the equivalent reliability pair $\{r_{eq,c}{}^0, r_{eq,c}{}^1\}$ for the output *c* of Fig. 4.2(b), we can split the vector $\mathbf{P}^*$ of (4.3) into two sub-vectors $\mathbf{P}_0{}^*$ and $\mathbf{P}_1{}^*$, which correspond to combination of a^* and b^* values leading to an error-free output of 0 and 1, respectively, and split the matrix **M** of (4.6) into two sub-matrices $\mathbf{M}_0$ and $\mathbf{M}_1$ accordingly. For instance, if the gate of Fig. 4.2 is an AND gate, then let

$$\left.\begin{aligned} \mathbf{P}_0^* &= [P^*_{00} \quad P^*_{01} \quad P^*_{10}] \\ \mathbf{P}_1^* &= [P^*_{11}] \end{aligned}\right\} \tag{4.9}$$

and **M** is split as follows:

$$\mathbf{M}=\begin{bmatrix} p_{00}^{00} & p_{00}^{01} & p_{00}^{10} & p_{00}^{11} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ p_{01}^{00} & p_{01}^{01} & p_{01}^{10} & p_{01}^{11} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ p_{10}^{00} & p_{10}^{01} & p_{10}^{10} & p_{10}^{11} \\ ---------- \\ p_{11}^{00} & p_{11}^{01} & p_{11}^{10} & p_{11}^{11} \end{bmatrix}=\begin{bmatrix} \mathbf{M}_{00} \\ \cdots\cdots \\ \mathbf{M}_{01} \\ \cdots\cdots \\ \mathbf{M}_{10} \\ -- \\ \mathbf{M}_{11} \end{bmatrix}=\begin{bmatrix} \mathbf{M}_0 \\ -- \\ \mathbf{M}_1 \end{bmatrix} \tag{4.10}$$

From Table 4.1, the output reliability vector for AND gate is

$$\mathbf{R}_{AND}=[r_c \quad r_c \quad r_c \quad 1-r_c]^T \tag{4.11}$$

The equivalent reliability pair for the output c is derived as

$$\left.\begin{aligned} r_{eq,c}^{0}(AND)&=\mathbf{P}_0^{*}\cdot\mathbf{M}_0\cdot\mathbf{R}_{AND}/(1-P_{c^*}) \\ r_{eq,c}^{1}(AND)&=\mathbf{M}_1\cdot(\mathbf{1}-\mathbf{R}_{AND}) \end{aligned}\right\} \tag{4.12}$$

where P_{c^*} is the probability of error-free signal c^*, and $\mathbf{1}=[1\ 1\ 1\ 1]^{\mathrm{T}}$.

Let $K_{ab}^{*}=P\{a^*b^*=11\}$. The error-free input probability vector $\mathbf{P}^*$ of (4.3) can be rewritten as:

$$\mathbf{P}^{*}=[1-P_{a^*}-P_{b^*}+K_{ab}^{*} \quad P_{b^*}-K_{ab}^{*} \quad P_{a^*}-K_{ab}^{*} \quad K_{ab}^{*}] \tag{4.13}$$

where P_{a^*} and P_{b^*} are the probabilities of error-free signals a^* and b^*, respectively, and K_{ab}^{*} can be easily obtained from P_{a^*}, P_{b^*} and P_{c^*}, depending on the gate type. For a 2-input AND gate in particular, $K_{ab}^{*}=P_{c^*}$. Table 4.2 shows the value of K_{ab}^{*} for different 2-input gates. Therefore, (4.12) can be calculated using $\{r_{eq,a}^{0}, r_{eq,a}^{1}\}$, $\{r_{eq,b}^{0}, r_{eq,b}^{1}\}$, P_{a^*}, P_{b^*} and P_{c^*}.

Table 4.2 Calculation of K_{ab}^{*}

Gate Type	K_{ab}^{*}
AND	P_{c^*}
NAND	$1-P_{c^*}$
OR	$P_{a^*}+P_{b^*}-P_{c^*}$
NOR	$P_{a^*}+P_{b^*}+P_{c^*}-1$
XOR	$(P_{a^*}+P_{b^*}-P_{c^*})/2$
XNOR	$(P_{a^*}+P_{b^*}+P_{c^*}-1)/2$

Table 4.3 Organizing $\mathbf{P}_0^*$, $\mathbf{P}_1^*$, $\mathbf{M}_0$ and $\mathbf{M}_1$ for 2-Input Gates

Gate type	**Input probability vector** $\mathbf{P^*}=[P_{00}^* \; P_{01}^* \; P_{10}^* \; P_{11}^*]$		**Prob. transfer matrix** $\mathbf{M}=[\mathbf{M}_{00} \; \mathbf{M}_{01} \; \mathbf{M}_{10} \; \mathbf{M}_{11}]^T$	
	$\mathbf{P}_0^*$	$\mathbf{P}_1^*$	$\mathbf{M}_0$	$\mathbf{M}_1$
AND	$[P_{00}^* \; P_{01}^* \; P_{10}^*]$	$[P_{11}^*]$	$\begin{bmatrix}\mathbf{M}_{00}\\ \mathbf{M}_{01}\\ \mathbf{M}_{10}\end{bmatrix}$	$\mathbf{M}_{11}$
NAND	$[P_{11}^*]$	$[P_{00}^* \; P_{01}^* \; P_{10}^*]$	$\mathbf{M}_{11}$	$\begin{bmatrix}\mathbf{M}_{00}\\ \mathbf{M}_{01}\\ \mathbf{M}_{10}\end{bmatrix}$
OR	$[P_{00}^*]$	$[P_{01}^* P_{10}^* P_{11}^*]$	$\mathbf{M}_{00}$	$\begin{bmatrix}\mathbf{M}_{01}\\ \mathbf{M}_{10}\\ \mathbf{M}_{11}\end{bmatrix}$
NOR	$[P_{01}^* P_{10}^* P_{11}^*]$	$[P_{00}^*]$	$\begin{bmatrix}\mathbf{M}_{01}\\ \mathbf{M}_{10}\\ \mathbf{M}_{11}\end{bmatrix}$	$\mathbf{M}_{00}$
XOR	$[P_{00}^* \; P_{11}^*]$	$[P_{01}^* \; P_{10}^*]$	$\begin{bmatrix}\mathbf{M}_{00}\\ \mathbf{M}_{11}\end{bmatrix}$	$\begin{bmatrix}\mathbf{M}_{01}\\ \mathbf{M}_{10}\end{bmatrix}$
XNOR	$[P_{01}^* \; P_{10}^*]$	$[P_{00}^* \; P_{11}^*]$	$\begin{bmatrix}\mathbf{M}_{01}\\ \mathbf{M}_{10}\end{bmatrix}$	$\begin{bmatrix}\mathbf{M}_{00}\\ \mathbf{M}_{11}\end{bmatrix}$

Table 4.4 Calculation of Equivalent Reliability Pair at Output for 2-Input Gates

Gate type	$r_{eq,c}^{\;0}$	$r_{eq,c}^{\;1}$
AND	$\mathbf{P}_0^* \cdot \mathbf{M}_0 \cdot \mathbf{R}_{AND} / (1 - P_{c*})$	$\mathbf{M}_1 \cdot (\mathbf{1} - \mathbf{R}_{AND})$
NAND	$\mathbf{M}_0 \cdot (\mathbf{1} - \mathbf{R}_{NAND})$	$\mathbf{P}_1^* \cdot \mathbf{M}_1 \cdot \mathbf{R}_{NAND} / P_{c*}$
OR	$\mathbf{M}_0 \cdot (\mathbf{1} - \mathbf{R}_{OR})$	$\mathbf{P}_1^* \cdot \mathbf{M}_1 \cdot \mathbf{R}_{OR} / P_{c*}$
NOR	$\mathbf{P}_0^* \cdot \mathbf{M}_0 \cdot \mathbf{R}_{NOR} / (1 - P_{c*})$	$\mathbf{M}_1 \cdot (\mathbf{1} - \mathbf{R}_{NOR})$
XOR	$\mathbf{P}_0^* \cdot \mathbf{M}_0 \cdot (\mathbf{1} - \mathbf{R}_{XOR}) / (1 - P_{c*})$	$\mathbf{P}_1^* \cdot \mathbf{M}_1 \cdot \mathbf{R}_{XOR} / P_{c*}$
XNOR	$\mathbf{P}_0^* \cdot \mathbf{M}_0 \cdot \mathbf{R}_{XNOR} / (1 - P_{c*})$	$\mathbf{P}_1^* \cdot \mathbf{M}_1 \cdot (\mathbf{1} - \mathbf{R}_{XNOR}) / P_{c*}$

The sub-vectors/matrices $\mathbf{P}_0^*$, $\mathbf{P}_1^*$, $\mathbf{M}_0$ and $\mathbf{M}_1$ given in (4.9) and (4.10) are just for AND gate. For other types of gate, both $\mathbf{P}^*$ and $\mathbf{M}$ can be split in a similar way. Table 4.3 shows these sub-vectors/matrices for 2-input AND/NAND, OR/NOR, and XOR/XNOR gates. The equivalent reliability pair $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ at the output c for different gates is calculated using Table 4.4, where $\mathbf{1} = [1\ 1\ 1\ 1]^{\mathrm{T}}$.

B. Correlated Case

When the equivalent reliabilities of inputs a and b are correlated (and so are the inputs by themselves), the computation of equivalent reliability pair for the output c becomes a bit more complicated. A typical example of such correlation is shown in Fig. 4.3(a), where $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ are the equivalent reliability pairs for inputs a and b, respectively. Clearly, these two pairs are correlated due to the signal correlation between a and b. However, the reliabilities of two fan-in gates (i.e., r_a and r_b) are still independent of each other. In other words, $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ are only partially correlated. This motivates us to split each of these reliability pairs equivalently into two dummy buffers, as shown in Fig. 4.3(b), where $r_a^{\ 0} = r_a^{\ 1} = r_a$ *and* $r_b^{\ 0} = r_b^{\ 1} = r_b$ are original reliabilities of two fan-in gates, while $\{r_{a'}^{\ 0}, r_{a'}^{\ 1}\}$ and $\{r_{b'}^{\ 0}, r_{b'}^{\ 1}\}$ are derived as follows to ensure the equivalent :

$$\left.\begin{aligned} r_{eq,a}^{i} &= r_{a'}^{i} r_a + (1 - r_{a'}^{i})(1 - r_a) \\ r_{eq,b}^{i} &= r_{b'}^{i} r_b + (1 - r_{b'}^{i})(1 - r_b) \end{aligned}\right\} \tag{4.14}$$

or

$$\left.\begin{aligned} r_{a'}^{i} &= (r_{eq,a}^{i} - 1 + r_a)/(2r_a - 1) \\ r_{b'}^{i} &= (r_{eq,b}^{i} - 1 + r_b)/(2r_b - 1) \end{aligned}\right\} \tag{4.15}$$

where i = 0, 1. Thus, the correlation between $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ can be dealt with by only considering the correlation between $\{r_{a'}^{\ 0}, r_{a'}^{\ 1}\}$ and $\{r_{b'}^{\ 0}, r_{b'}^{\ 1}\}$. Since r_a and r_b are independent, the probability transfer matrix (denoted by $\mathbf{M}$') from *a'*and *b'* to a and b is given by (4.7) where we let $r_{eq,a}^{\ 0} = r_{eq,a}^{\ 1} = r_a$, and $r_{eq,b}^{\ 0} = r_{eq,b}^{\ 1} = r_b$, i.e.,

$$\mathbf{M}' = \begin{bmatrix} r_a r_b & r_a(1-r_b) & (1-r_a)r_b & (1-r_a)(1-r_b) \\ r_a(1-r_b) & r_a r_b & (1-r_a)(1-r_b) & (1-r_a)r_b \\ (1-r_a)r_b & (1-r_a)(1-r_b) & r_a r_b & r_a(1-r_b) \\ (1-r_a)(1-r_b) & (1-r_a)r_b & r_a(1-r_b) & r_a r_b \end{bmatrix} \tag{4.16}$$

which is a symmetrical 4×4 matrix.

Therefore, the question becomes: How to find the matrix of probability transfer (denoted by $\mathbf{M}_{\text{cor}}$) from a^* and b^* *to* a' and b'? To answer this question, I first introduce a correlation coefficient between a^* and b^* as follows:

$$\theta^{(i,j)} = \frac{P\{a^* b^* = ij\} - P(a^* = i) \cdot P(b^* = j)}{\sqrt{P(a^* = i)[1 - P(a^* = i)]P(b^* = j)[1 - P(b^* = j)]}}, \quad i, j = 0 \text{ or } 1 \tag{4.17}$$

where $-1 \le \theta^{(i,j)} \le 1$. From (4.3) and (4.13) with $P_{a^*} = 1 - P\{a^* = 0\}$ and $P_{b^*} = 1 - P\{b^* = 0\}$, (4.17) can be rewritten as

$$\theta^{(i,i)} = -\theta^{(i,j)} = \frac{K_{ab}^* - P_{a^*} \cdot P_{b^*}}{\sqrt{P_{a^*}(1 - P_{a^*}) P_{b^*}(1 - P_{b^*})}}, \quad i, j = 0 \text{ or } 1, i \ne j \tag{4.18}$$

where $K_{ab}^* = P\{a^* b^* = 11\}$ which can be found from Table 4.2, depending on the gate type. The error-free signals a^* and b^* are independent if $\theta^{(i,j)} = 0$, and are positive-correlated (or negative-correlated) if $\theta^{(i,j)} > 0$ (or $\theta^{(i,j)} < 0$). The strongest correlation occurs when $\theta^{(i,j)} = \pm 1$. Assuming that the reliability correlation between a' and b' is mainly caused by unreliable gates/signals shared by the transitive fan-in cones of a' and b' (this assumption is approximate, but is reasonable especially when all gates in the combinational circuit have same reliability value), then the joint reliability $r_{a'b'}^{(i,j)}$ (i.e., the probability that both a' and b' are reliable) can be estimated as:

$$r_{a'b'}^{(i,j)} = r_{a'}^i r_{b'}^j + \theta^{(i,j)} \cdot \sqrt{r_{a'}^i (1 - r_{a'}^i) r_{b'}^j (1 - r_{b'}^j)} \tag{4.19}$$

where $i, j = 0$ or 1. Thus, the matrix of probability transfer from a^* and b^* *to* a' and b' is obtained by modifying (4.7) as:

$$\mathbf{M}_{cor} = \begin{bmatrix} r_{a'b'}^{(0,0)} & r_{a'}^0 - r_{a'b'}^{(0,0)} & r_{b'}^0 - r_{a'b'}^{(0,0)} & (1 - r_{a'}^0 - r_{b'}^0 + r_{a'b'}^{(0,0)}) \\ r_{a'}^0 - r_{a'b'}^{(0,1)} & r_{a'b'}^{(0,1)} & (1 - r_{a'}^0 - r_{b'}^1 + r_{a'b'}^{(0,1)}) & r_{b'}^1 - r_{a'b'}^{(0,1)} \\ r_{b'}^0 - r_{a'b'}^{(1,0)} & (1 - r_{a'}^1 - r_{b'}^0 + r_{a'b'}^{(1,0)}) & r_{a'b'}^{(1,0)} & r_{a'}^1 - r_{a'b'}^{(1,0)} \\ (1 - r_{a'}^1 - r_{b'}^1 + r_{a'b'}^{(1,1)}) & r_{b'}^1 - r_{a'b'}^{(1,1)} & r_{a'}^1 - r_{a'b'}^{(1,1)} & r_{a'b'}^{(1,1)} \end{bmatrix} \tag{4.20}$$

where $r_{a'b'}^{(i,j)}$ is given by (4.19).

Consider two extreme cases in which $\theta^{(i,j)} = 0$ and 1. When $\theta^{(i,j)} = 0$, the $r_{a'}^i$ and $r_{b'}^j$ are independent and, from (4.19), the joint reliability $r_{a'b'}^{(i,j)} = r_{a'}^i r_{b'}^j$. Therefore, the estimation of (4.20) is accurate. If $\theta^{(i,i)} = 1$, then $r_{a'}^0 = r_{b'}^0 = r^0$, $r_{a'}^1 = r_{b'}^1 = r^1$, and $r_{a'b'}^{(i,i)} = r^i$ (i=0,1), $r_{a'b'}^{(i,j)} = r^i r^j - \sqrt{r^i (1 - r^i) r^j (1 - r^j)}$ (i, j=0,1, $i \ne j$).

The overall matrix **M** of probability transfer (from a^* and b^* to a and b) for Fig. 4.3 is expressed as

$$\mathbf{M} = \mathbf{M}_{cor} \cdot \mathbf{M}' \tag{4.21}$$

where $\mathbf{M}_{\text{cor}}$ and **M**' are given by (4.20) and (4.16), respectively. The equivalent reliability pair $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ at the output c of Fig. 4.3 is calculated by using Tables III and IV, depending on the gate type. In case both $\{r_{a'}^{\ 0}, r_{a'}^{\ 1}\}$ and $\{r_{b'}^{\ 0}, r_{b'}^{\ 1}\}$ are also independent, the matrix **M** of (4.21) shall be equal to $\mathbf{M}_{\text{ind}}$ of (4.7) which assumes the independence of both signal and reliability (this can be easily proved by using (4.15)).

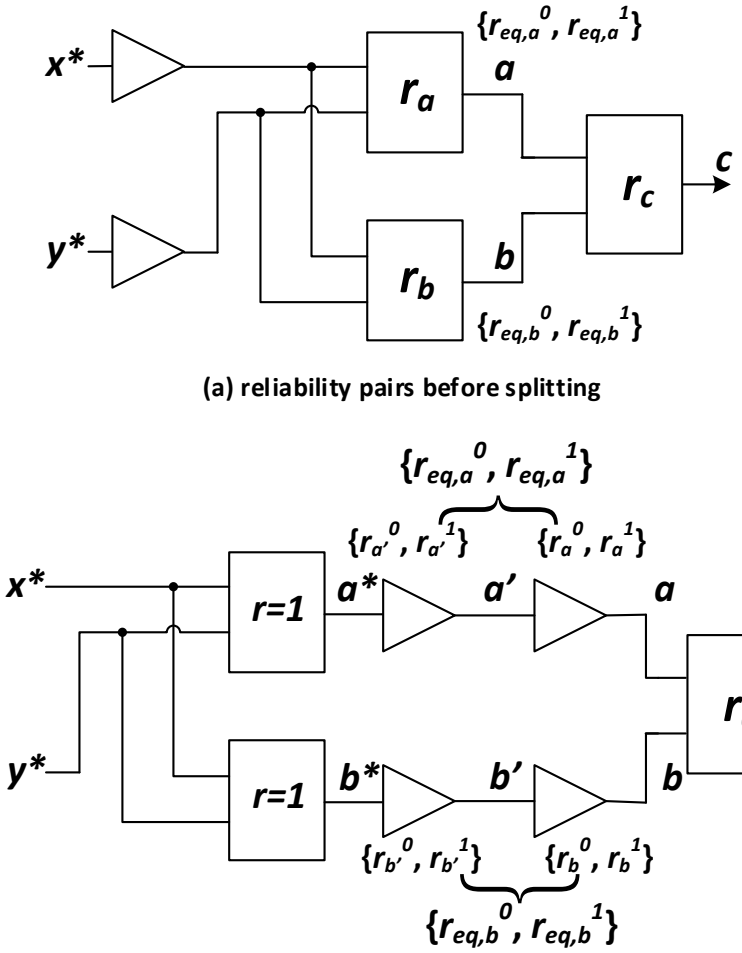
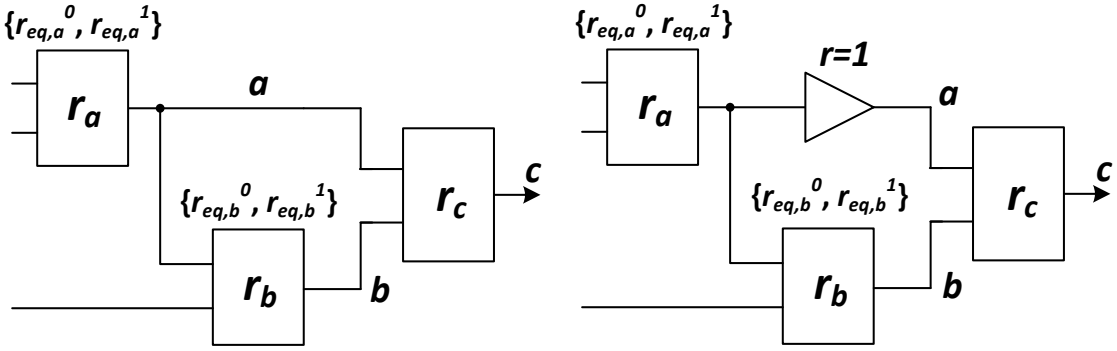


Figure 4.3. Dealing with signal and reliability correlation

Another case in Fig. 4.3, which deserves special attention, is when the signal *a* is a transitive fanin of signal *b*, as illustrated in Fig. 4.4(a), where both equivalent reliability pairs for *a* and *b* will be affected by r_a. In this case, one can simply insert a reliable dummy buffer (i.e., with reliability of 1) into the signal *a*, as shown in Fig. 4.4(b). Mathematically, this is equivalent to setting the value of r_a to 1 before using (4.15), (4.16) and (4.20).



(a) Input a is a transitive fan-in of input b (b) Adding a reliable dummy buffer into input a

Figure 4.4 A special case of Fig. 4.3

4.3.3 Algorithm

In the above discussions, we deal with 2-input gates. For an inverter with only one input, it is straightforward to calculate the equivalent reliability pair $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ at its output *c* as follows:

$$r_{eq,c}^{i} = r_c r_{eq,in}^{1-i} + (1-r_c)(1-r_{eq,in}^{1-i}), \quad i = 0,1 \tag{4.22}$$

where r_c is the reliability of the inverter, and $\{r_{eq,in}^{\ 0}, r_{eq,in}^{\ 1}\}$ denotes the equivalent reliability pair at the input. Also, if any input signal in Fig. 4.3 is a primary input which is assumed to be reliable, then its equivalent reliability pair shall be set to {1, 1} before performing the reliability propagation discussed in the previous section.

If the gate under consideration has more than two inputs (such as 3-input AND gate), the correlation between these inputs would become more complicated. A quick solution is to decompose the gate into a few 2-input gates (e.g., decomposition of 3-input AND gate to two 2-input AND gates), and set the reliability of last gate (i.e., the one that drives the output) to the value of multiple-input gate's reliability, assuming other gates are reliable. In the real world, majority of gates in combinational circuits have only two inputs, and most logic synthesis tools

(such as SIS [158]) also provide an option of doing technology decomposition using two-input gates only.

Based on the proposed ER model, I present an algorithm below to summarize the whole procedure of reliability analysis for combinational circuits. Without loss of generality, we assume the circuit contains 2-input gates only in the algorithm description.

Algorithm: *Reliability Analysis for Combinational Circuits*
Input: Gate reliabilities and primary input probabilities
Output: Reliability of primary outputs
begin
step1: calculate all signal probabilities for error-free circuit;
step2: sort the gates in a topological order;
step3: ***for*** each gate c in the circuit
obtain $\{r_{eq,a}^{\ 0}, r_{eq,a}^{\ 1}\}$ and $\{r_{eq,b}^{\ 0}, r_{eq,b}^{\ 1}\}$ for two inputs a and b;
compute $\{r_{a'}^{\ 0}, r_{a'}^{\ 1}\}and\{r_{b'}^{\ 0}, r_{b'}^{\ 1}\}$ using (4.15);
find $K_{ab}^{\ *}$ from Table 4.2;
find $\mathbf{P_0}^*$ and $\mathbf{P_1}^*$according to (4.13) and Table 4.3;
find $\mathbf{M_0}$ and $\mathbf{M_1}$ according to (4.16)~(4.21) and Table 4.3;
obtain $\mathbf{R}$ from Table I;
compute $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ from Table 4.4;
return $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$;
step4: ***for*** each primary output F_j
let $\{r_j^0, r_j^1\} = \{r_{eq,Fj}^{\ 0}, r_{eq,Fj}^{\ 1}\}$
compute the overall reliability r_j for F_j using (4.1);
return r_j for all outputs;
end

It should be mentioned that the above algorithm requires the computation (in step 1) of error-free signal probabilities for all nodes in the circuit. While this is a most time-consuming step, many efficient approaches have been available (see the following section 4.3.4 for more details). The rest of the algorithm can be implemented with a linear time complexity $O(N)$, where N is the number of gates. This is because the algorithm recursively takes one gate at a time, and it only needs a constant computation time (for just a few 4×4 matrix multiplications) to process each gate for the reliability propagation from its inputs to its output. Any change in reliability at any gate(s)

just requires linear-time computations in the above steps 3 and 4, since steps 1 and 2 can be done only once. This provides good opportunities to explore, in a more efficient way, the effect of individual gate reliabilities on the output reliability. Also, the proposed method can be easily extended for gates that are associated with a reliability pair (instead of a single reliability value being used above), due to the nature of the model which takes both r^0 and r^1 into consideration. More discussions on this extension will be given in Section 4.5.

4.3.4 Time Complexity

When it comes to computation of error-free signal probabilities at all nodes in the circuit, many methods and CAD tools are available. Binary decision diagrams (BDDs) are one of them which have been well documented in literature [159, 160]. It is well known that the efficiency of using BDDs for this computation is determined by their size which in turn depends on the order of input variables. Generally speaking, finding an optimal order of input variables is an NP-complete problem. Finding a good order, however, just needs a polynomial-time in most cases. Another technique that can be used for the above probability analysis is the correlation coefficient method (CCM) [135]. This method deals with signal correlations by considering the propagation of correlation coefficients from fan-out nodes, and accounts for first-order conditional probabilities by using correlation coefficients between a pair of signals. It has been shown that the CCM can produce good results in terms of average error and root mean square (RMS) deviation with a polynomial-time complexity (approximately $O(N^{1.5})$, where N is the number of gates). By contrast, the reliability analysis approaches using PTM or Bayesian Network require an exponential-time complexity in the worst case, due to the fact that they either need a traversal of all input combinations or require an exponentially-growing number of conditional probabilities to support the operation. For small circuits, Monte-Carlo simulation is also a good option for both probability and reliability calculation.

4.3.5 Examples

In this section, I first take two simple examples (one for independent case, and the other for correlation case) to show how to calculate the overall reliability. Then, I use a benchmark circuit C17 to further illustrate the proposed model. As an example for independent case, let us consider a 3-gate circuit, as shown in Fig. 4.5, which has four independent primary inputs x_1 through x_4, implementing the logic function $F = x_1x_2x_3x_4$. Assuming the three gate reliabilities of r_a, r_b and r_c in Fig. 4.5 with all input probabilities of $\sqrt{0.5}$ (i.e., $P_{xi} = \sqrt{0.5} \approx 0.707$ for i = 1~4), we have $\mathbf{P}^*$ = [0.25 0.25 0.25 0.25] for the output gate c, and the probability transfer matrix $\mathbf{M}$ is given by

(4.7) where both $r_{eq,a}^{\ 0}$ and $r_{eq,a}^{\ 1}$ are set to r_a, and both $r_{eq,b}^{\ 0}$ and $r_{eq,b}^{\ 1}$ are set to r_b. Using Tables 4.1~4.4 for AND gate, we derive the equivalent reliability pair at the output F as follows: $r_{eq,F}^{\ 0} = (1+r_c-r_ar_b+2r_ar_br_c)/3$, and $r_{eq,F}^{\ 1} = 1- r_c-r_ar_b+2r_ar_br_c$. The (average) reliability of the circuit is calculated as (refer to (1)): $r_F = P_{F*}\, r_{eq,F}^{\ 1} + (1- P_{F*})\, r_{eq,F}^{\ 0} = 0.25r_{eq,F}^{\ 1} + 0.75r_{eq,F}^{\ 0} = 0.5(1- r_ar_b) + r_ar_br_c$, where $0.5 \le r_F \le r_c$ for $0.5 \le r_c \le 1$. For $r_c = 1$ in particular, $r_F = 0.5(1+r_ar_b)$. This result shows that r_F is always no greater than r_c, but could be higher than r_a or r_b. In this sense, r_c plays a more important role than both r_a and r_b in determining the overall reliability.

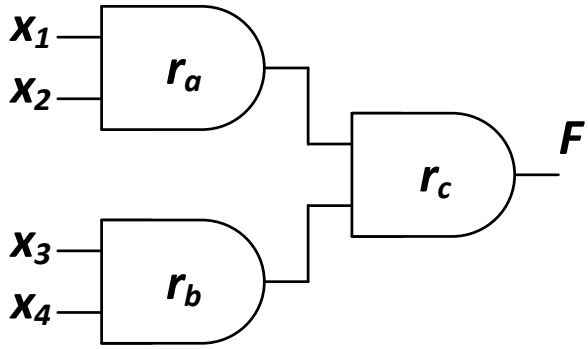


Figure 4.5. An example circuit for independent case.

As one of the simplest examples for correlation case, we look at an inverter with reliability of r_c, and treat it as a NAND gate with two identical inputs (a and b) whose correlation is the strongest (i.e., $\theta^{(i,i)} =1$). Let the equivalent reliability pair of both inputs be $\{r_{eq,in}^{\ 0}, r_{eq,in}^{\ 1}\}$. In other words, let $r_{a'}^{\ 0} = r_{b'}^{\ 0} = r_{eq,in}^{\ 0}$, and $r_{a'}^{\ 1} = r_{b'}^{\ 1} = r_{eq,in}^{\ 1}$ in Fig. 4.3(b), where r_a and r_b shall be set to 1 in this case, assuming a reliable dummy buffer is inserted to each input (refer to Fig. 4.4). By following the procedure of section III-B for the correlation case, one can obtain the equivalent reliability pair $\{r_{eq,c}^{\ 0}, r_{eq,c}^{\ 1}\}$ for its output c, which would be the same as (4.22). This indicates that the proposed model is accurate under this extreme case when handling the signal and reliability correlation through (4.18)~(4.21).

The last example circuit for illustration is shown in Fig. 4.6 which contains six 2-input NAND gates with signal correlations. First, we assume that all primary input probabilities are $P_{in} = 0.5$, and the error probability for each gate is equally 5% (i.e., gate reliability $r_{gate} = 0.95$). I implemented the proposed ER model (i.e., the algorithm presented in Section III-C) on the circuit, and calculated the signal probability and reliability for all nodes which are listed in Table 4.5

where the accurate results from Monte-Carlo simulation are also shown for comparison. It can be clearly seen from the table that our ER model provides highly accurate evaluation results, even for reconvergent gates (such as *G*5 and *G*6 in the circuit). The results also reveal that signal reliabilities keep reducing from the primary inputs to primary outputs due to error propagation.

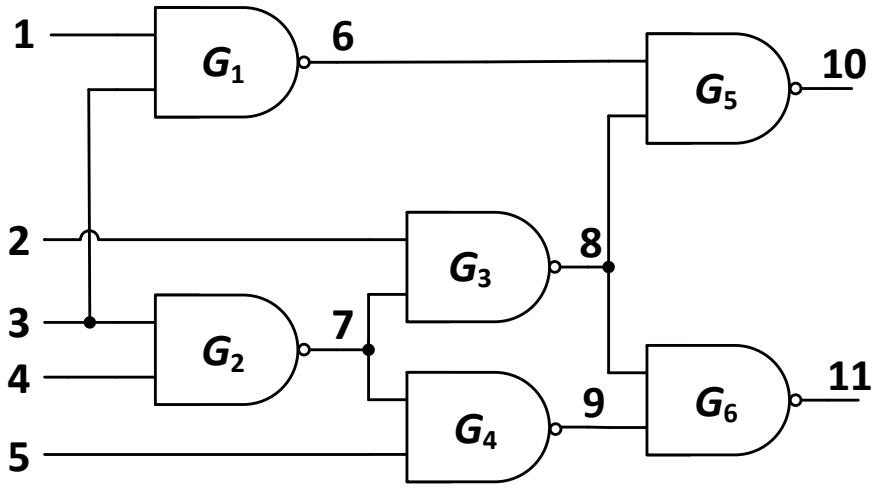


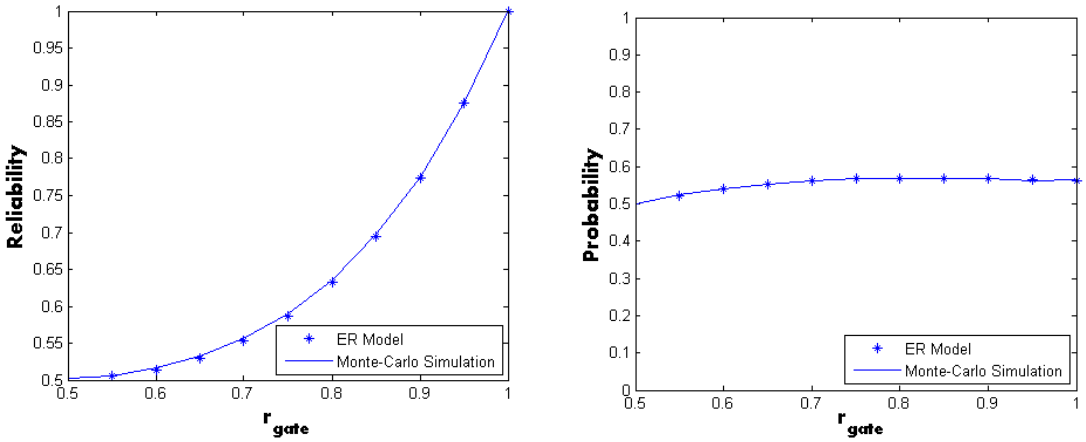
Figure 4.6. Schematic of benchmark circuit C17.

Table 4.5. Signal Probability and Reliability for C17 with $r_{gate} = 0.95$

Node	Monte-Carlo (10^6 runs)		Proposed ER model	
	reliability	*probability*	*equivalent reliability* r_{eq}	*probability*
6, 7	0.9500	0.7250	0.9500	0.7250
8, 9	0.9275	0.6238	0.9275	0.6238
10	0.8757	0.5634	0.8747	0.5644
11	0.8758	0.5635	0.8737	0.5689

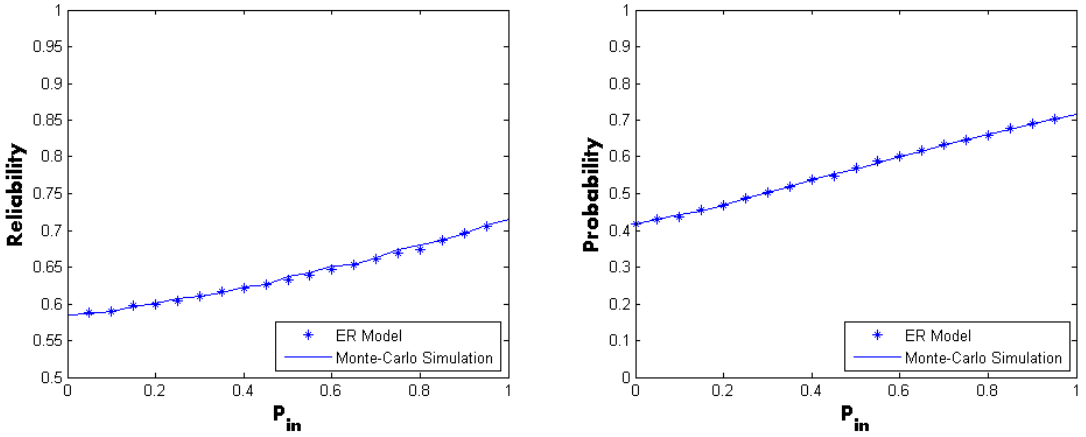
To show the scalability of proposed method, we also calculated the output probability and reliability for C17 with different gate reliabilities and/or input probabilities. Fig. 4.7 shows the

results for the output node 10 (similar results for node 11) under different gate reliabilities with the input probability P_{in} of 0.5. It can be seen from the figure that while the output reliability increases with individual gate reliabilities, the output signal probability is less dependent on gate reliability values under this particular case of P_{in} = 0.5.



(a) Output reliability vs. gate reliability (b) Output probability vs. gate reliability

Figure 4.7. The output reliability and probability for C17 with different gate reliabilities (assuming P_{in} = 0.5)



(a) Output reliability vs. input probability (b) Output probability vs. input probability

Figure 4.8. The output reliability and probability for C17 with different input probabilities (assuming r_{gate} = 0.8)

Fig. 4.8 shows the reliability and probability for the same output node 10 under different input probabilities (assuming the gate reliability r_{gate} of 0.8). As expected, the input probabilities affect

both probability and reliability at the output. The good agreement between the results from MC simulation and our ER model confirms that a very high accuracy can be achieved for this example circuit with signal correlations, regardless of input probability and gate reliability.

4.4 Simulation results

The proposed ER model was implemented in MATLAB. Simulations were performed using a computer with 2.80GHz processor and 4GB RAM. The results from Monte-Carlo based reliability analysis were used as accurate data. Simulation results for reliability evaluation on a number of small circuits (assuming all gate reliabilities of 0.95 and all primary input probabilities of 0.5) are shown in Table 4.6, where both average error and maximum error are calculated over all primary outputs of the circuit (see column 3 of the table). It can be seen from the table that the average error of the ER model in evaluating the reliability is less than 1% (for most cases) with high efficiency of computation.

Table 4.6. Performance of the ER Model on Small Circuits in comparison with MC simulation (r_{gate} = 0.95)

circuit	size	reliability analysis using ER model			Monte-Carlo
		avg. error (%)	max error (%)	runtime (s)	runtime (s)
C17	6	0.53	0.89	0.024	18.60
Full adder (XOR/NAND)	5	0.11	0.21	0.025	7.47
Full adder (NAND)	12	1.45	1.65	0.025	18.04
Comparator	4	0.31	0.88	0.025	7.77
Decoder2	6	0.02	0.03	0.024	5.80

We also took ISCAS'85 benchmarks to further demonstrate the performance of the proposed ER model on large circuits. Simulations were conducted in comparison with PGM method [145] and BDEC algorithm [150]. Again, errors are calculated against Monte-Carlo simulation results.

When running the Monte-Carlo simulation on large circuits, I used a standard deviation computed on the fly as termination criteria. Our simulations show that the standard deviation of 10^{-3} can be obtained for one million runs, which means that the results are accurate for this number of runs with a confidence level of over 99.9% [161]. For the PGM approach, it may become very time consuming and intractable for larger circuits, as its time complexity is exponentially dependent on the number of inputs. Therefore, some approximation techniques are necessary. Assuming the input vectors are uniformly distributed (i.e., all input probabilities of 0.5), I took 10^3 samples of input vectors for simulations with PGM. These samples were randomly chosen from the input vector space for each benchmark circuit.

Table 4.7. Comparison of ER Model, PGM and BDEC for Reliability Analysis on ISCAS'85 Benchmark Circuits with r_{gate} = 0.9

Circuit	size	ER model			PGM approach (10^3 samples) [145]			BDEC algorithm [150]			MC (10^6 runs)
		avg. error (%)	max error (%)	run-time* (s)	avg. error (%)	max error (%)	run-time (s)	avg. error (%)	max error (%)	run-time (s)	run-time (s)
C432	160	0.66	1.67	0.060	0.69	1.8	4.57	13.19	8.60	0.024	185.7
C499	202	0.15	0.34	0.054	0.15	0.54	4.81	8.64	1.63	0.024	211.7
C880	383	0.55	5.00	0.059	1.53	8.72	9.62	4.77	9.73	0.031	391.5
C1355	546	3.42	3.61	0.068	0.95	1.18	14.19	3.14	0.63	0.035	491.8
C1908	880	2.16	5.51	0.080	0.49	1.98	18.10	6.75	21.08	0.042	881.7
C2670	1193	0.96	13.01	0.109	2.61	20.2	27.12	2.83	16.60	0.054	1197.0
C3540	1669	3.26	11.42	0.163	1.89	8.63	38.79	16.46	46.31	0.067	1580.3
C5315	2307	0.85	4.27	0.201	8.91	46.38	61.25	7.34	23.39	0.098	2500.3
C7552	3512	0.9	11.72	0.250	1.30	11.91	75.07	6.72	42.24	0.129	3583.2
average	**-**	**1.43**	**6.28**	**-**	**2.06**	**11.26**	**-**	**7.76**	**18.98**	**-**	**-**

*not including the time spent on calculation of error-free signal probabilities

Table 4.8. Comparison of ER Model, PGM and BDEC for Reliability Analysis on ISCAS'85 Benchmark Circuits with r_{gate} = 0.8

Circuit	size	ER model			PGM approach (10^3 samples) [145]			BDEC algorithm [150]			MC (10^6 runs)
		avg. error (%)	max error (%)	run-time* (s)	avg. error (%)	max error (%)	run-time (s)	avg. error (%)	max error (%)	run-time (s)	runtime (s)
C432	160	0.10	0.32	0.83	0.54	1.59	4.66	13.11	22.94	0.031	182.6
C499	202	0.09	0.26	0.11	0.1	0.31	4.92	7.23	7.42	0.029	202.5
C880	383	0.16	0.82	0.11	0.61	2.83	9.17	2.14	9.56	0.040	372.6
C1355	546	3.13	3.39	0.11	1.26	1.66	12.21	4.75	4.98	0.045	471.6
C1908	880	1.30	1.89	0.16	0.39	0.85	18.69	7.46	11.62	0.048	842.5
C2670	1193	0.47	7.60	0.21	2.43	16.61	25.72	2.08	18.27	0.055	1151.4
C3540	1669	0.7	2.74	0.31	0.077	2.27	39.46	10.43	21.37	0.097	1615.90
C5315	2307	0.24	0.98	0.55	10.88	43.16	61.42	7.33	21.40	0.109	2548.38
C7552	3512	0.18	4.94	0.74	2.68	13.68	75.19	4.87	14.91	0.146	3731.82
average	-	**0.77**	**2.55**	-	**2.18**	**9.22**	-	**6.60**	**14.72**	-	-

*not including the time spent on calculation of error-free signal probabilities

Tables 4.7 and 4.8 show simulation results with r_{gate} = 0.9 and 0.8, respectively, assuming all input probabilities of 0.5. It can be seen from both tables that the average error in reliability analysis using the ER model is around 1%, compared to approximately 2% for the PGM and 7% for the BDEC. Also, for most circuits, the maximum error of output reliabilities produced by the ER model is much less than that with either PGM or BDEC. This is because the ER model takes into account, but none of PGM and BDEC does, (a) the signal and reliability correlations and (b) reliability pairs that are associated with all signals.

In terms of computational efficiency, the BDEC is fastest while it is least accurate, as can be seen from both Tables 4.7 and 4.8 where the MC is most time-consuming with accurate results. The ER model stays somewhere in between. It should be mentioned that the runtime for the ER model shown in these tables does not include the time spent on calculation of error-free signal probabilities. The reason is two-fold: (1) Computation of error-free signal probabilities in a combinational circuit has been well studied, and can be done efficiently in general, depending on the specific method or implementation being used, as discussed in Section 4.3.4, and (2) This

computation needs to be done only once for a given circuit regardless of any changes with gate reliabilities. Thus, it does not necessarily represent an issue of concern or extra runtime, especially when dealing with explorations of gate reliabilities for circuit reliability improvement, which will be further discussed later in Section 4.5. It can be clearly seen from these two tables that the ER model only takes fraction of a second in computation even for large circuits. Therefore, the ER model can still be considered as one that is faster than most existing methods (except BDEC), even with inclusion of the computation time for error-free signal probabilities, which are required in order to handle reliability correlations.

4.5 Discussion and future work

4.5.1 Handling gate reliability pair

So far we assume each gate within the circuit has a single reliability value (i.e., a constant reliability regardless of its input signals). However, this assumption may not be always valid since gate reliability does depend on its input logic values in the real world [155]. For instance, the reliability of a 2-input AND gate for both inputs being "1" may differ from that for both inputs being "0". To handle this situation, we associate each gate g with a reliability pair $\{r_g^0, r_g^1\}$ (as we did for a signal reliability), where r_g^0 (or r_g^1) represents a conditional probability that its output is correct given the error-free output being 0 (or 1). More specifically, let us assume three reliability pairs of $\{r_a^0, r_a^1\}$, $\{r_b^0, r_b^1\}$ and $\{r_c^0, r_c^1\}$ (instead of r_a, r_b and r_c) in Fig. 4.3. Thus, the matrices **M**' of (4.16) and $\mathbf{M}_{\text{cor}}$ of (4.20) as well as **R** in Tables 4.1 and 4.4 need to be modified accordingly for reliability analysis. The details are described below.

First, (4.16) needs to be updated as $\mathbf{M}'_{\text{update}}$ which is obtained from (4.7) where $r_{eq,a}^{\ i}$ and $r_{eq,b}^{\ i}$ are replaced by $r_a^{\ i}$ and $r_b^{\ i}$ (i = 0, 1), respectively, i.e.,

$$\mathbf{M}'_{update} = \begin{bmatrix} r_a^0 r_b^0 & r_a^0(1-r_b^0) & (1-r_a^0)r_b^0 & (1-r_a^0)(1-r_b^0) \\ r_a^0(1-r_b^1) & r_a^0 r_b^1 & (1-r_a^0)(1-r_b^1) & (1-r_a^0)r_b^1 \\ (1-r_a^1)r_b^0 & (1-r_a^1)(1-r_b^0) & r_a^1 r_b^0 & r_a^1(1-r_b^0) \\ (1-r_a^1)(1-r_b^1) & (1-r_a^1)r_b^1 & r_a^1(1-r_b^1) & r_a^1 r_b^1 \end{bmatrix} \tag{4.23}$$

Secondly, equations of (4.14) and (4.15) shall be modified as:

$$\left.\begin{aligned} r_{eq,a}^{i} &= r_{a'}^{i} r_a^{i} + (1-r_{a'}^{i})(1-r_a^{1-i}) \\ r_{eq,b}^{i} &= r_{b'}^{i} r_b^{i} + (1-r_{b'}^{i})(1-r_b^{1-i}) \end{aligned}\right\} \tag{4.24}$$

or

$$\left.\begin{aligned} r_{a'}^{i} &= (r_{eq,a}^{i} - 1 + r_{a}^{1-i})/(r_{a}^{i} - 1 + r_{a}^{1-i}) \\ r_{b'}^{i} &= (r_{eq,b}^{i} - 1 + r_{b}^{1-i})/(r_{b}^{i} - 1 + r_{b}^{1-i}) \end{aligned}\right\} \tag{4.25}$$

where $i = 0, 1$. Therefore, equations (4.18) and (4.19) shall be calculated by using (4.25) (instead of using (4.15) previously). Again, it should be pointed out that if the signal *a* is a transitive fanin of signal *b* (or vice versa), both r_a^0 and r_a^1 (or both r_b^0 and r_b^1) shall be set to 1 before using (4.19), (4.20), (4.23) and (4.25), as discussed in Fig. 4.4.

Finally, the output reliability vector **R** of Table 4.1 needs to be modified as a pair of vectors $\mathbf{R}^0$ and $\mathbf{R}^1$, where $\mathbf{R}^0$ (or $\mathbf{R}^1$) can be obtained from **R** by letting $r_c = r_c^0$ (or r_c^1). This requires the recalculation of $r_{eq,c}^0$ and $r_{eq,c}^1$ in Table 4.4 accordingly as follows: In the $r_{eq,c}^0$ (or $r_{eq,c}^1$) column, all **R** shall be replaced by $\mathbf{R}^0$ (or $\mathbf{R}^1$). For example, $\mathbf{R}_{AND}^0 = [r_c^0 \;\; r_c^0 \;\; r_c^0 \;\; 1 - r_c^0]^{\mathrm{T}}$, and $\mathbf{R}_{AND}^1 = [r_c^1 \;\; r_c^1 \;\; r_c^1 \;\; 1 - r_c^1]^{\mathrm{T}}$.

4.5.2 Multiple outputs

In the above discussions, each individual primary output is processed separately for reliability analysis (refer to (4.1)). For a combinational circuit with multiple outputs, its reliability can also be evaluated as a joint reliability of all outputs. If these primary outputs are independent of each other, the overall (joint) reliability would simply be the product of all individual reliabilities. In most cases, however, these outputs are correlated. This implies that the joint reliability is usually higher than this product, depending on how strong the correlations may be. Since all possible correlations have to be considered, it would be very expensive in computation to find the exact value of joint reliability (with time complexity of $O(2^m)$ in general, where *m* is the number of primary outputs). In the following, I show a more efficient way to estimate the joint reliability. The basic idea is to combine two outputs at a time for treatment of correlations, and use the proposed ER model again to find an equivalent reliability with slight modifications for Table 4.4. Clearly, it will only take $O(m)$ time to complete the computation.

We first take any two primary output signals (denoted again as *a* and *b*) to inputs of a reliable dummy gate (say, AND gate) which is inserted to capture the correlation between *a* and *b*. Assuming the output of the dummy gate is *c*, one can follow the similar procedure given in section 4.3.2 to find the input probability vector $\mathbf{P}^*$ and probability transfer matrix **M** before using Table 4.4. The only difference is that the calculation of $r_{eq,c}^0$ and $r_{eq,c}^1$ in Table 4.4 needs to be modified slightly. Since the signal *c* is considered reliable only when both *a* and *b* are reliable, the

product term $\mathbf{M}_i \cdot \mathbf{R}$ (or $\mathbf{M}_i \cdot (\mathbf{1}-\mathbf{R})$) in both $r_{eq,c}^{\ 0}$ and $r_{eq,c}^{\ 1}$ columns of Table 4.4 (for the case of dummy AND gate, only first row of Table 4.4) shall be replaced by a vector $\mathbf{M}_{c,i}$ (i = 0, 1) which consists of diagonal elements of $\mathbf{M}$. For instance, for AND gate, $\mathbf{M}_{c,0} = [m_{11}\ \ m_{22}\ \ m_{33}]^T$ and $\mathbf{M}_{c,1} = [m_{44}]^T$, where m_{jj} (j = 1~4) is the j-th diagonal element of $\mathbf{M}$, and thus $r_{eq,c}^{\ 0}$ and $r_{eq,c}^{\ 1}$ are calculated as: $r_{eq,c}^{\ 0} = \mathbf{P}^* \cdot \mathbf{M}_{c,0} / (1-P_{c^*})$ and $r_{eq,c}^{\ 1} = \mathbf{M}_{c,1}$, where $\mathbf{P}^*$ is given by (4.13) with $K_{ab}^{\ *} = P_{c^*}$ (for AND gate). Again, a^*, b^* and c^* in (4.13) represent the error-free version of signals a, b and c, respectively. Once these two primary outputs (a and b) is processed, the output c of the dummy gate (with reliability of 1) will be combined with another primary output by using one more dummy AND gate. This process repeats until all primary outputs are taken (with a total of $m-1$ dummy gates required). The joint reliability for all primary outputs is then given by $r_{joint} = P_d^* r_{eq,d}^{\ 1} + (1-P_d^*) r_{eq,d}^{\ 0}$, where $\{r_{eq,d}^{\ 0}, r_{eq,d}^{\ 1}\}$ and $P_d^{\ *}$ are the reliability pair and error-free signal probability, respectively, at the output of the last dummy gate.

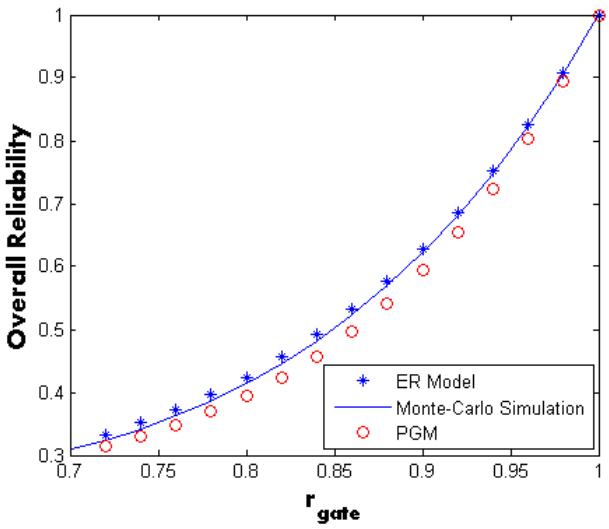


Figure 4.9. Simulation results on the overall output reliability of C17 with different values of gate reliability.

We used C17 below as an example to show the performance of proposed model in handling multiple outputs. Since C17 has two correlated outputs, the overall (joint) output reliability can be found by following the above procedure with only one dummy (AND) gate required. Fig. 4.9 shows the results from the ER model, MC simulation and PGM, which cover different values of gate reliability r_{gate}. It should be noticed that the MC produces accurate results, while the PGM estimates the overall reliability simply as the product of two individual output reliabilities by

ignoring signal correlations. The curves in Fig. 4.9 indicate that the ER model slightly overestimates the overall reliability for C17, but much closer to accurate values than PGM. For instance, when r_{gate} = 0.95 in particular, the exact reliability from MC is 0.7836, compared to 0.7868 generated by our ER model and 0.7619 by PGM.

4.5.3 Reliability improvement

From the above discussions (especially Table 4.4), the overall reliability at the circuit's outputs depend on: i) primary input probabilities which are related to the input probability vector $\mathbf{P}^*$, ii) signal correlations introduced by the circuit topology as a result of logic synthesis, which affect both $\mathbf{P}^*$ and **M** (the probability transfer matrix), and iii) gate types and gate reliabilities, which play a key role in determining **M** and **R** (the output reliability vector). While it would be unlikely to find an exact closed-form expression for the output reliability, the efficiency of the proposed ER model allows us to explore how individual gate reliabilities affect the overall reliability for given primary input probabilities and circuit topology. Intuitively, this can be done gate by gate starting from primary outputs. When processing any gate, one can change its reliability value as well as the equivalent reliability pairs of its fanin gates, and estimate how the equivalent reliability at its output responds to these changes. Different gates and/or input signals may have different impacts on the output reliability. The gates (or inputs) to which the output reliability is most sensitive can be defined as *reliability-critical* gates (or signals). Raising these gates' reliabilities would be critical to the overall reliability improvement. As computation of error-free signal probabilities at all nodes in the circuit (see sections 4.3.3 and 4.3.4) is required only once under our reliability analysis model, the whole process of finding reliability-critical gates can be done in a very efficient manner. This offers a full potential to take advantage of the proposed model for reliability improvement. More research work on this subject is demanded in the future.

Further work is also needed to improve the accuracy of the proposed model. The source of errors in analyzing the reliability from our ER model comes mainly from the approximation that is made in dealing with the reliability correlation using (4.19), which is based on signal correlation. While it makes sense and promises good results with (4.19) especially under some extreme cases where the correlation coefficient θ equals to 0 or ±1, there is still room for improvement in estimating the reliability correlation under general cases where $0< |\theta| <1$. A possible solution is to use Monte-Carlo simulation with a small number of runs for estimation of the reliability correlation, at the cost of increasing computation time.

4.6 Summary

We have presented a reliability analysis model for combinational circuits using the concept of equivalent reliability. Due to its recursive nature, the proposed model only takes a linear time to find the reliability at outputs (excluding computation of error-free signal probabilities within the circuit, which can be done by BDDs efficiently and effectively). Both signal and reliability correlations have been taken into consideration with detailed discussions. Since the signal probability and reliability are dependent of each other, the proposed method uses a reliability pair which represents two separate reliability values for different error-free logic outputs (i.e., "0" or "1"). This helps reduce errors during the propagation of both probability and reliability from primary inputs to outputs. Once the equivalent reliability pair becomes available, signal probabilities within the circuit can also be easily found for other purposes (such as dynamic power estimation). Simulation results with benchmark circuits have shown the advantages of the proposed model in terms of either efficiency or accuracy, or both, when compared with other methods including the PGM, BDEC, as well as MC simulation. Another merit with our reliability model is that it can be further utilized for other potential applications, such as finding the joint reliability for multiple outputs and identifying reliability-critical gates for reliability improvement.

CHAPTER 5

CONCLUSION

In the near future, the eventual end to the roadmap of semiconductors is anticipated which hinders further scaling of CMOS technology. Alternative approaches are desired to satisfy those expectations such as the continued increase in the density of memory and logic blocks beyond the terabit regime. This paper was devoted pursing solutions to modeling and simulation issues of Single electronics that come up as SEDs are increasingly involved in next generation circuit architecture design. It has been proved that the behavior models are more appropriate for SEDs simulation in terms of time complexity and accuracy. In addition, it has been found that the self-learning parallel computing architectures, which are fault-tolerant and usually area consuming, could be implemented by SEDs compatibly. The paper also explored the dynamic characteristics of SE-based circuits and proposed a novel statistical model that could be used to estimate and analyze its transient response both efficiently and accurately. The proposed model was then extended for finite temperature condition in more practical applications. Furthermore, this paper studied the reliability estimation problem for combination logic circuits, as those SEDs are usually unreliable and temperature sensitive. I proposed a new recursive algorithm that could estimate the reliability of any logic circuits within short time compared with existing methods and with reasonable error. It has been shown that the proposed method was 100% accurate for independent case and relatively accurate for correlated case. This fast analysis method showed the possibility of reliability improvement and future fault tolerant circuits design.

In Chapter 1, first I gave some background knowledge of Single electronics and Single electron devices. The brief introduction of SE technology was presented, as well as its motivation (mainly related to the concern that it is able to conquer those drawbacks of conventional devices). The advantages of SE were explained in detail, including the small feature size and low power consumption; and the current state of practical applications using SEDs was described shortly. In addition, the challenges of SE were discussed as well.

In Chapter 2 I started with the working principle of SET-based devices and a short history of their experimental studies, as well as the current fabrication method. Then I introduced the existing simulation and modeling methods of SEDs with comparison and evaluation. Next I focused on the modeling technique of those new devices and presented a novel compact analytical model for single-electron tunneling (SET) based turnstiles by describing its work principles in Verilog modeling language. It has been shown that the new model conquered the drawbacks such

as time consuming in Monte-Carlo framework and discontinuities of traditional SPICE model. Hybrid SET/MOS circuit co-simulations were successfully performed by implementing the proposed model; and extensive simulation results showed the advantages of realizing some application circuits using SED's new functionality, especially for self-learning network works.

In Chapter 3 I discussed the timing analysis of SET-based devices where conventional approaches were usually failed to give accurate results, as the electron tunneling is stochastic in nature and its delay has to be understood in a probabilistic sense. I modeled the delay in circuits as a continuous-time Markov process and presented a statistical method in order to describe their dynamic characteristics, under both ideal and non-ideal temperature environment. I theoretically proved the accuracy of proposed methods; and simulations were shown to verify its effectiveness. For SE logic circuits, I extensively analyzed the switching delay of an inverter as well as the NAND gate as examples, in order to show the detail process of our propose delay estimation method; and some existing simulation methods (including MC simulation, ME, macro-modeling technique, and step estimation method) were also discussed and evaluated for comparison study.

In Chapter 4, I mainly focused on the problem of reliability estimation and analysis of logic circuits, as the reliability of integrated circuits becoming an increasingly big concern with the continuous scaling of CMOs technology and emerging nano-scale SEDs. First I gave a short review regarding to the currents methods that can be used for reliability estimation, followed by our new methods using the concept of equivalent reliability with recursive computation. I showed that the proposed method provided improvements over the state-of-the-art in terms of either efficiency or accuracy, or both. It has been shown that the computation from our model lead to exact results for circuits without reconvergent fanouts, and gave good estimation results for circuits with correlation of both signal and reliability. Simulations on ISCAS'85 benchmark circuits were shown that our approach obtains a significant speedup over MC simulation, and more accurate results than other existing methods. Also I gave some discussion on its potential application of this method such as reliability improvement and fault-tolerant design.

At last, this Chapter 5 concludes this thesis paper.

REFERENCES/BIBLIOGRAPHY

[1] The National Technology Roadmap for Semiconductors (1997). The National Semiconductor Association, San Jose, CA. Available online: www.public.itrs.net

[2] Stephen M. Goodnick and J. Bird, "Quantum-Effect and Single-Electron Devices," *IEEE Trans. Nanotechnology*, vol. 2, no. 4, pp. 368-383, December 2003.

[3] K. Likharev, "Single-electron devices and their applications," *Proceedings of the IEEE*, vol. 87, no. 4, pp. 606-632. April 1999.

[4] D. Goldhaber-Gordon et al., "Overview of Nanoelectronic Devices," *Proceedings of the IEEE*, vol. 85, no. 4, pp.521-540. April 1997.

[5] M. T. Bohr, "Nanotechnology Goals and Challenges for Electronic Applications," *IEEE Trans. Nanotechnology*, vol. 1, no. 1, pp. 56-62, March 2002.

[6] Research Activities in NTT Basic Research Laboratories (volume 15 Fiscal 2004), Nippon Telegraph and Telephone Corporation (NTT) Basic Research Laboratories, Kanagawa, Japan. Available online: www.brl.ntt.co.jp

[7] C. Wassahuber, "Single-electronics—How it works, how it's used, how it's simulated," *Proceedings of the International Symposium on Quality electronic design (ISQED)*, pp. 502-507, 2002.

[8] J. A. Hutchby, G. I. Bourianoff, V.V. Zhirnov, and J. E. Brewer, "Extending the road beyond CMOS," *IEEE Circuits Devices Mag.*, vol. 18, pp. 28–41, Feb. 2002.

[9] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4-22, April 1987.

[10] C. Wasshuber, *Computational Single-electronics*, Springer-Verlag, Wien, 2001.

[11] S. Mahapatra and A. MihaiIonescu, *Hybrid CMOS single-electron- transistor device and circuit design*, Artech House Inc., 2006.

[12] H. Gravert and M. H. Devoret, *Single charge Tunneling—Coulomb Blockade Phenomena in Nanostructures*. New York: Plenum, 1992.

[13] R. Durrett, *Probability: Theory and examples (4th ed.)*, New York: Cambrige Univ. Press, 2004.

[14] J. Havil, *Gamma: Exploring Euler's constant.* Princetion, NJ: Princeton Univ. Press, 2003.

[15] D. M. Harris and S. Harris, *Digital design and computer architecture*, Morgan Kaufmann, 2013.

[16] T. Kohonen, *Self-Organizztion and Associative Memory*, Spring-Verlag, Berlin, 1984.

[17] Yongshun Sun, Rusli and N. Singh, "Room temperature operation of silicon single electron transistor fabricated using optical lithography," *IEEE Trans. Nanotechnology*, vol. 10, no. 1, pp. 96-98, January 2011.

[18] A. Beaumont, C. Dubuc, J. Beauvais, and D. Drouin, "Room temperature single-electron transistor featuring gate-enhanced on-state current," *IEEE Electron Device Lett.*, vol. 30, no. 7, pp. 766–768, July 2009.

[19] K. Nishiguchi, A. Fujiwara, and Y. Ono, "Room-temperature-operating data processing circuit based on single-electron transfer and detection with metal-oxide-semiconductor field-effect transistor technology," *Appl. Phys. Lett.* 88. p. 183101, 2006.

[20] Y. Ono et al., "Si complementary single electron inverter with voltage gain," Appl. Phys. Lett, vol. 76, pp. 3121, 2000.

[21] Y. Ono et al., "Si complementary single electron inverter," International Electron Device Meeting (IEDM'99), pp. 367-370, 1999.

[22] R. H. Chen, A. N. Korotkov, and K. K. Likharev, "Single-electron transistor," *Appl. Phy. Lett.*, vol. 68, pp. 1954-1956, 1996.

[23] J. R. Tucker, "Complementary digital logic based on the Coulomb blockade," *J. Appl. Phys.*, vol. 72, pp. 4399-4413, 1992.

[24] K. Nishiguchi, Y. Ono and A. Fujiwara, "Single electron counting statistics of shot noise in nanowire Si metal-oxide-semiconductor field-effect transistors," *J. Appl. Phys.*, 98, 193502, 2011.

[25] Akira et al, "Current quantization due to single electron transfer in Si-wire charge coupled devices," *J. Appl. Phys.*, 84, 1323, 2004.

[26] J. P. Pekola et al, "Hybrid single electron turnstile: towards a quantum standard of electric current," *2010 Conference on Precision Electromagnetic Measurements*, 2010.

[27] Y. Ono et al, "Silicon single electron turnstile," *Device Research Conference*, pp. 137-138, 2003.

[28] Akira Fujiwara et al, "Single electron turnstile using Si-wire Charge-couple devices," *International Semiconductor Device Research Symposium*, pp. 485-486, 2003.

[29] L. J. Geerligs et al, "Frequency-locked turnstile device for single electrons," *Phys. Rev. Lett.*, vol. 64, pp. 2691-2694, May 1990.

[30] K. Yokoi et al, "Electrical control of capacitance dispersion for single electron turnstile operation in common-gated junction arrays," *J. Appl. Phys.*, 108, 053710, 2010.

[31] A. Kemppinen et al, "Experimental investigation of hybrid single electron turnstiles with high charging energy," *J. Appl. Phys.*, 94, 172108, 2009.

[32] Hiroshi Nakashima and Kiyohiko Uozumi, "Single electron tunneling dynamics in a turnstile designed with asymmetric tunnel barriers," *J. Appl. Phys.*, 89, 5785, 2001.

[33] Chuan-Yu Lin and W. Zhang, "Single electron turnstile pumping with high frequencies," *J. Appl. Phys.*, 99, 072105, 2011.

[34] Yoshinao Mizugaki, "Blocking Charge Oscillation in a series array of two tiny tunnel junctions with Resistive Ground path from its island electrode," *IEEE Trans. Nanotechnology*, vol. 11, no. 1, pp. 194-199, January 2012.

[35] R. J. Schoelkopf et al, "The Radio-Frequency Single-Electron Transistor (RF-SET): A Fast and Ultrasensitive Electrometer," *Science*, 280, 1238, 1998.

[36] J. Bylander, T. Duty and P. Delsing, "Current Measurement by Real-Time counting of single charges," *Nature*, 434, 361, 2005.

[37] Helko E. van den Brom et al, "Counting Electrons One by One: Overview of a joint European research project," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 2, April 2003.

[38] D. V. Averin et al, "Ultimate accuracy of single electron DC current standards," *J. Appl. Phys.*, 73, 1297, 1993.

[39] C. Wasshuber et al., "A comparative study of single electron memories," *IEEE Trans. Electron Devices*, vol. 45, no. 11, pp. 2365-2341, November 1998.

[40] K. Nishiguchi, H. Inokawa, Y. Ono, A. Fujiwara, and Y. Takahashi, "Multilevel memory using single-electron turnstile," *Electronics Letters*, vol. 40, no. 4, pp. 229–230, February 2004.

[41] S. Mahapatra and A. M. Ionescu, "A novel single electron SRAM architecture," *4th IEEE International Conference on Nanotechnology*, pp. 287-290, 2004.

[42] Ioannis Karafyllidis, "Design and simulation of a single electron random access memory array," *IEEE Trans. Circuits and Systems-I*, vol. 49, pp. 1370-1375, 2002.

[43] B. Li and C. Chen, "Design methodology for electron trap memory cells", *8th IEEE International Conference on Nanotechnology*, pp. 22-24, 2008.

[44] Naila Syed and C. Chen, "Low power SET-based SRAM cell using negative differential conductance," *11th IEEE International Conference on Nanotechnology*, pp. 744-747, 2011.

[45] H. Muller and H. Mizuta, "Memory cell simulation on the nanometer scale", *IEEE Trans. Electron Devices*, vol. 47, no. 10, pp. 1826-1830, October 2000.

[46] K. Yano et al, "Room temperature single electron memory", *IEEE Trans. Electron Devices*, vol. 41, no. 9, pp. 1628-1638, September 1994.

[47] S. Mahapatra, A. M. Ionescu, and K. Banerjee, "Analytical modeling of single electron transistor for hybrid CMOS-SET analog IC design," *IEEE Trans. Electron Devices*, vol. 51, no. 11, pp. 1772–1781, November 2004.

[48] Y. Chi, B. Sui et al., "A compact analytical Model for Multi-island single electron transistors," *IEEE 8th International Conference on ASIC (ASICON'09)*, pp. 662-665, 2009.

[49] Sang-Hoon Lee et al., "A practical SPICE model based on the Physics and Characteristics of realistic Single electron Transistors," *IEEE Trans. Nanotechnology*, vol. 1, no. 4, pp. 226-232, December 2002.

[50] C. Hu, S. D. Cotofana, and J. Jiang, “Compact current and current noise models for single electron tunneling transistors,” *4th IEEE Conference on Nanotechnology*, pp. 361-264, 2004.

[51] I. Krout et al., “Drain current model for single electron transistor operating at high temperature”, *8th International Multi-Conference on Systems*, Signals & Devices, 2011.

[52] C. Wasshuber, H. Kosina, and S. Selberherr, "SIMON- A simulator for single-electron tunnel devices and circuits," *IEEE Trans. on Computer-Aided Design*, vol. 16, pp. 937–944, September 1997.

[53] W. Wei, J. Han, and F. Lombardi, "Modeling a single electron turnstile in HSPICE," *GLSVLSI'12*, Salt Lake City, Utah, USA, pp. 16–23, May 2012.

[54] L. R. C. Fonseca, A. N. Korotkov, K. K. Likharev, and A. A. Odintsov, “A numerical study of the dynamics and statistics of single electron systems,” *J. Appl. Phys.*, vol. 78, pp. 3238-3251, 1995.

[55] M. Y. A. Ismail et al., “A new fast and accurate Steady State Master Equation Model for capacitively-coupled Single electron Transistor,” *the 23rd National Radio Science Conference (NRSC 2006)*, pp. D4. 1-10, 2006.

[56] J. Cetnar, “General solution of Bateman equations for nuclear transmutations,” *Annals of Nuclear Energy*, 33, pp. 640-645, 2006.

[57] N. Allec, R. Knobel, and L. Shang, “Adaptive simulation for single-electron devices,” *Design, Automation and Test in Europe (DATE)*, pp. 1021-1026, 2008.

[58] N. Allec, R. Knobel, and L. Shang, “SEMSIM: Adaptive Multiscale simulation for single-electron devices,” *IEEE Trans. Nanotechnology*, vol. 7, no. 3, pp. 351-354, May 2008.

[59] M. Kirihara and K. Taniguchi, “Monte Carlo simulation for single electron circuits,” *Asia and South Pacific Proceedings of the Design Automation Conference (ASP-DAC)*, pp. 333-337, 1997.

[60] G. Zardalidis and I. G. Karafyllidis, “SECS: A new single-electron circuit simulator,” *IEEE Trans. circuit and systems-I*, vol. 55, no. 9, pp. 2774-2784. 2008.

[61] H. Inokawa and Y. Takahashi, “A compact analytical model for asymmetric single-electron tunneling transistors,” *IEEE Trans. Electron Devices*, vol. 50, no. 2, pp. 455-460, 2003.

[62] Y. S. Yu, S. W. Hwang, and D. Ahn *et al*., “Macromodeling for single electron transistor for efficient circuit simulation,” *IEEE Trans. Electron Devices*, vol. 46, no. 8, pp. 1667-1671, 1999.

[63] S. Mahapatra, A. M. Ionescu, and K. Banerjee, “A quasi-analytical SET model for few electron circuit simulation,” *IEEE Electron device letters*, vol. 23, no. 6, pp. 366-368, 2002.

[64] Y. S. Yu, S. W. Hwang and D. Ahn, “Transient modeling of single-electron transistors for efficient circuit simualtion by SPICE,” *IEE Proc.-Circuits Devices Syst.*, vol. 152, no. 6, pp. 691-696, 2005.

[65] S. Amakawa, H. Majima, H. Fukui, M. Fujishima, and K. Hoh, “Single-electron circuit simulation,” *IEICE Trans Electron*, vol. E81-C, no. 1, pp. 21-29, 1997.

[66] C. Chen, “Delay estimation on single-electron-tunneling-based logic gates,” *IEEE Trans. Nanotechnology*, vol. 10, pp. 1254-1263, 2011.

[67] N. M. Zimmermana, E. Hourdakis, Y. Ono, A. Fujiwara, and Y. Takahashi, “Error mechanisms and rates in tunable-barrier single-electron turnstiles and charge-coupled devices,” *J. Appl. Phys.*, 96, 5254, 2004.

[68] C. Lageweg, C. Cotofana, and S. Vassiliadis, “Single electron encoded latches and flip-flops,” *IEEE Trans. Nanotechnology*, vol. 3, no. 2, pp. 237-248, 2004.

[69] Jieun Lee et al., “Comparative Study on Energy-efficiencies of Single electron transistor-based binary Full Adders including nonideal effects,” *IEEE Trans. Nanotechnology*, vol. 10, no. 5, pp. 1180-1190, September 2011.

[70] Ken Uchida et al., “Programmable single electron transistor logic for future low-power intelligent LSI: proposal and room-temperature operation” *IEEE Trans. Electron Devices*, vol. 50, no. 7, pp. 1623-1630, July 2003.

[71] T. Oya, T. Asai, T. Fukui, and Y. Amemiya, “A Majority-logic Device using an irreversible single electron box” *IEEE Trans. Nanotechnology*, vol. 2, no. 1, pp. 15-22, March 2003.

[72] C. Lageweg, S. Cotofana and S. Vassiliadis, "A linear threshold gate Implementation in single electron technology" *IEEE Computer society workshop on VLSI*, pp. 93-98, 2001.

[73] C. Lageweg, S. Cotofana and S. Vassiliadis, "A Full Adder implementation using SET based linear threshold Gates" *9th International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 665-668, 2002.

[74] S. Kasai and H. Hasegawa, "A single electron Binary-Decision-Diagram Quantum logic circuit based on Schottky wrap gate control of a GaAs Nanowire Hexagon," *IEEE Electron Device Letters*, vol. 23, no. 8, pp. 446-447, August 2002.

[75] K. Degawa et al., "A single electron transistor logic gate family and Its Application," *34th International Symposium on Multiple-Valued Logic (ISMVL'04)*, 2004.

[76] G. T. Zardalidis and I. Karafyllidis, "Design and Simulation of a Nanoelectronic single electron universal Fredkin Gate," *IEEE Trans. Circuit and Systems*, vol. 51, no. 12, pp. 2395-2403, December 2004.

[77] L. Gao, F. Alibart and D. B. Strukov, "Programmable CMOS/Memristor threshold Logic," *IEEE Trans. Nanotechnology*, vol. 12, no. 2, pp. 115-119, March 2013.

[78] Ken Uchida et al., "Room Temperature operation of Multifunctional single electron transistor logic," *International Electron Device Meeting (IEDM'00)*, pp. 863-865, 2000.

[79] D. Griveau et al., "Single electron CMOS-like one bit Full Adder," *13th International Conference on Ultimate Integration on silicon (ULIS)*, pp. 77-80, 2012.

[80] N. Asahi, M. Akazawa and Y. Amemiya, "Single electron logic device based on the binary decision diagram," *IEEE Trans. Electron Devices*, vol. 44, no. 7, pp. 1109-1116, July 1997.

[81] G. Deng and C.Chen, "Performance analysis and improvement for hybrid CMOS-SET circuit architectures," *1st Microsystems and Nanoelectronics Research Conference (MNRC'08)*, pp. 109-112, 2008.

[82] G. Deng, G. Ren and C.Chen, "Full Adder design using hybrid CMOS-SET parallel architectures," *9th IEEE International Conference on Nanotechnology*, pp. 206-209, 2009.

[83] G. Deng and C.Chen, "Towards Robust design of hybrid CMOS-SETs using feedback architectures," *10th IEEE International Conference on Nanotechnology*, pp. 1125-1128, 2010.

[84] G. Deng, and C. Chen, "Hybrid CMOS-SET arithmetic circuit design using Coulomb blockade oscillation characteristic," *Journal of Computational and Theoretical Nanoscience*, vol. 8, no. 8, pp. 1–7, 2011.

[85] G. Deng and C.Chen, "A hybrid CMOS-SET multiplier using frequency modulation," *11th IEEE International Conference on Nanotechnology*, pp. 1167-1169, 2011.

[86] G. Deng and C.Chen, "A SET/MOS hybrid multiplier using frequency synthesis", *IEEE Trans. VLSI*, vol. 21, no. 9, pp. 1738-1742, 2013.

[87] G. Deng and C.Chen, "Binary multiplication using hybrid MOS and multi-gate single electron transistors," *IEEE Trans. VLSI*, vol. 21, no. 9, pp. 1573-1582, 2013.

[88] W. Wei, J. Han, and F. Lombardi, "Design and evaluation of a hybrid memory cell by single-electron transfer," *IEEE Trans. Nanotechnology*, vol. 12, no. 1, pp. 57-70, 2013.

[89] X. Ou and N-J. Wu, "Analog-digital and digital-analog converters using single-electron and MOS transistors," *IEEE Trans. Nanotechnology*, vol. 4, no. 6, pp. 722–729, 2005.

[90] W. Zhang, N-J. Wu, T. Hashizume, and S. Kasi, "Novel hybrid voltage controlled ring oscillators using single electron and MOS transistors," *IEEE Trans. Nanotechnology*, vol. 6, no. 2, pp. 146–157, 2007.

[91] W. Zhang and N-J. Wu, "A novel hybrid phase-locked loop frequency synthesizer using single-electron devices and CMOS transistors," *IEEE Trans. Circuits and Systems-I*, vol. 54, pp. 2516-2527, 2007.

[92] W. Zhang and N-J. Wu, "Nanoelectronic circuit architectures based on single-electron turnstiles," *IEEE International Nanoelectronics Conference (INEC)*, pp. 515–519, 2008.

[93] W. Zhang and N-J. Wu, "Performing fast addition and multiplication by transferring single electrons," *7th IEEE International Conference on Nanotechnology*, pp. 690-694, 2007.

[94] W. Zhang and N-J. Wu, "Smart Universal Multiple-Valued logic gates by transferring electrons," *IEEE Trans. Nanotechnology*, vol. 7, no. 4, pp. 440–450, 2008.

[95] Hiroshi Inokawa et al., "A multiple-valued logic and memory with combined single electron and metal-oxide-semiconductor transistors," *IEEE Trans. Electron Devices*, vol. 50, no. 2, pp. 462-470, February 2003.

[96] W. Wei, J. Han and F. Lombardi, "A hybrid memory cell using single electron transfer" *IEEE/ACM International Symposium on Nanoscale architectures*, pp. 16-23, 2011.

[97] P. Junsangsri and F. Lombardi, "Design of a hybrid memory cell using Memristance and Ambipolarity," *IEEE Trans. Nanotechnology*, vol. 12, no. 1, pp. 71-80, January 2013.

[98] R. Parekh, A. Beaunont, J, Beauvais, and D. Drouin, "Simulation and design methodology for hybrid SET-CMOS Integrated logic at 22-nm room-temperature operation," *IEEE Trans. Electron Devices*, vol. 59, no. 4, pp. 918-923, 2012.

[99] Changyun Zhu et al, "Characterization of single electron tunneling transistors for designing low-power embedded systems," *Trans. VLSI*, vol. 17, no. 5, pp. 646-659, May 2009.

[100] T. Oya, "Thermal-noise-exploiting operation of single-electron majority logic circuits with conventional clock signals," *IEEE Trans. Nanotechnology*, vol. 11, no. 1, pp. 134-138, 2012.

[101] T. Oya, T. Asai, R. Kagaya, T. Hirose, and Y. Amemiya, "Neuronal synchrony detection on single-electron neural network," *Chaos, Solitions Fractals*, vol. 27, no. 4, pp. 887-894, 2006.

[102] T. Talaska and R. Dlugosz, "Initialization mechanism in Kohonen neural network inplemented in CMOS technology," *in Proc. Eur. Symp. Artif. Neural Netw.*, pp. 337-342, April 2008.

[103] D. DeSieno, "Adding a conscience to competitive learning," *in Proc. IEEE Conf. Neural Netw.*, pp. 117-124, July 1988.

[104] S. C. Ahalt et al., "Competitive learning algorithms for vector quantization," *Neural Netw.*, vol. 3, pp. 277-290, 1990.

[105] R. Dlugosz, T. Talaska, W. Pedrycz and R. Wojtyna, "Realization of the conscience mechanism in CMOS implementation of winner-takes-all self-organizing neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 961-971, June 2010.

[106] M. Mansuri, D. Liu, and C. K. Yang, "Fast frequency acquisition phase-frequency detectors for GSamples/s phase-locked loops," *IEEE Journal of Solid-State Circuit*, vol. 37, no. 10, pp. 1331–1334, 2002.

[107] S. Levantino, L. Romano, S. Pellerano, C. Samori, and A. L. Lacaita, "Phase noise in digital frequency dividers," *IEEE Journal of Solid-State Circuit*, vol. 39, no. 5, pp. 775–784, May 2004.

[108] Fangzhou Yu and Jinwen Zhang, "Single-walled Carbon nanotube pirani gauges prepared by DEP assembly," *IEEE Trans. Nanotechnology*, vol. 12, no. 3, pp. 323-329, May 2013.

[109] Ime J. Umoh et al, "A dual-gate Graphene FET model for circuit simulation: SPICE implementation," *IEEE Trans. Nanotechnology*, vol. 12, no. 3, pp. 427-435, May 2013.

[110] Damien Querlioz et al, "Immunity to Device variation in a spiking neural network with memristive nanodevices," *IEEE Trans. Nanotechnology*, vol. 12, no. 3, pp. 288-295, May 2013.

[111] K. Sridharan et al, "Efficient Multiternary Digit Adder Design in CNTFET technology," *IEEE Trans. Nanotechnology*, vol. 12, no. 3, pp. 283-287, May 2013.

[112] Y. Xia et al, "An Integrated optimization approach for nanohybrid circuit cell mapping," *IEEE Trans. Nanotechnology*, vol. 10, no. 6, pp. 1275-1284,November 2011.

[113] E. M. Izhikevich et al, "Bursts as a unit of neural information: selective communication via resonance," *Trends in Neuroscience*, March 2003.

[114] K. Mohanram and N. A. Touba, "Cost-effective approach for reducing soft error failure rate in logic circuits," *International Test Conference (ITC)*, pp. 893-901, 2003.

[115] S. Gustavsson et al, "Counting Statistics of single electron transport in a Quantum Dot," *Physical Review Letters*, 96, 076605, 2006.

[116] G. Y. Hu and R. F. O'Connell, "Exact solution for charge solitons in two coupled one-dimensional arrays of small tunnel junctions," *Physical Review*, vol. 54, no. 3, pp. 1522-1525, 1996.

[117] D. M. Kaplan, V. A. Sverdlov and K. K. Likharev, "Shot noise in frustrated single electron arrays," *Appl. Phys. Lett*, 83, 2662, 2003.

[118] Ivana Djuric et al, "Shot noise in resonant tunneling through an interacting Quantum Dot with intradot spin-flip scattering," *IEEE Trans. Nanotechnology*, vol. 4, no. 1, pp. 71-76, January 2005.

[119] G. Kiesslich et al, "Shot noise in tunneling through a quantum dot array," *Phys. Stat. Sol*, no. 4, pp. 1293-1296, 2003.

[120] S. F. Babiker and R. Naeem, "Shot noise suppression in single electron transistors," *IEEE Trans. Nanotechnology*, vol. 11, no. 6, pp. 1267-1272, November 2012.

[121] S. Babiker, "Simulation of single electron transport in Nanostructured Quantum Dots," *IEEE Trans. Nanotechnology*, vol. 52, no. 3, pp. 392-396, March 2005.

[122] S. Babiker, "Shot noise in single electron tunneling systems: A semiclassical model," *IEEE Trans. Nanotechnology*, vol. 10, no. 5, pp. 1191-1195, 2011..

[123] Xuejun Chen et al, "Neuromorphic pattern recognition using SET technology," *6th World Congress on Intelligent Control and Automation*, pp. 9958-9960, 2006.

[124] Brian Munsky and Mustafa Khammash, "The finite state projection algorithm for the solution of the chemical master equation," *J. Chem. Phys.*, 124, 044104, 2006.

[125] Shinichi et al, "Physical random number generator based on MOS structure after soft breakdown," *IEEE Journal of Solid-state Circuits*, vol. 39, no. 8, August 2004

[126] R. D. Yates and D. J. Goodman, *Probability and stochastic processes*. USA: John Wiley & Sons, 1999.

[127] Y. Taur et al, "CMOS scaling into the nanometer regime," *in Proc. IEEE*, vol. 85, pp. 486-504, April 1997.

[128] T. A. Fulton and G. J. Dolan, "Observation of single-electron charging effects in small tunnel junctions," *Phy. Rev. Lett.*, vol. 59, no. 1, pp. 109-112, 1987.

[129] D. V. Arevin and K. K. Likharev, "Single electronics: a correlated transfer of single electrons and Copper pairs in systems of small tunnel junctions," In *Mesoscopic phenomena in solids*. North-Holland, Amsterdam, pp. 173-271, 1991.

[130] G. J. Dolan, "Offset masks for lift-off photoprocessing," Appl. Phys. Lett.., vol. 31, pp. 337-339, 1977.

[131] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *IEEE Trans. VLSI*, vol. 17, no. 1, 2009.

[132] T, Rejimon and S. Bhanja, "Scalable probabilistic computing models using Bayesian network," *in Proc. IEEE Int. Midw. Symp. Circuits Syst. (MWSCAS)*, pp. 712-715, 2005.

[133] J. T. Flaquer et al, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities," *Microelectronics Reliability*, vol. 50, pp. 1215-1218, 2010.

[134] L. Chen and M. B. Tahoori, "An efficient probability framework for error propagation and correlation estimation," *in Proc. IEEE 18th Int. Symp. On-Line Testing (IOLTS)*, pp. 170-175, 2012.

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

[136] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Testability measures in pseudorandom testing," *IEEE Trans. on Computer-Aided Design*, vol. 11, no. 6, pp. 794-800, June 1992.

[137] R. I. Bahar, J. Chen and J. Mundy, "A probabilistic-based design for nanoscale computation," in Nano, *Quantum and Molecular Computing: Implications to High Level Design and Validation*, S. Shukla and R. I. Bahar, Eds. Norwell, MA: Kluwer, Chapter 5, 2004.

[138] R. I. Bahar, J. Mundy and J. Chen, "A probability-based design methodology for nanoscale computation," *in Proc. Int. Conf. Computer-Aided Design*, pp. 480-486, November 2003.

[139] A. R. Kermany et al, "A study of MRF-based circuit implementation," *in Proc. Int. Conf. Electronic Design*, pp. 1-4, December 2008.

[140] D. Bhaduri et al, "Nanolab-A tool for evaluating reliability od defect-tolerant nanoarchitectures," *IEEE Trans. Nanotechnology*, vol. 4, no. 4, pp. 381-394, July 2005.

[141] X. Lu, J. Li and W. Zhang, "On the probabilistic characterization of Nano-based circuits," *IEEE Trans. Nanotechnology*, vol. 8, no. 2, pp. 258-259, March 2009.

[142] J. Han and P. Jonker, “A system architecture solution for unreliable nanoelectronic devices,” *IEEE Trans. Nanotechnology*, vol. 1, no. 4, pp. 201-208, December 2002.

[143] J. B. Gao et al, “Bifurcations and fundamental error bounds for fault-tolerant computations,” *IEEE Trans. Nanotechnology*, vol. 4, no. 4, pp. 395-402, July 2005.

[144] J. Han et al, “Faults, error bounds and reliability of nanoelectronic circuits,” in *Proc. Int. Conf. Application-specific Systems, architecture and processors* (*ASAP*’05), pp. 247-253, 2005.

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

[146] J. Han et al, “On the reliability of computational structures using majority logic,” *IEEE Trans. Nanotechnology*, vol. 10, no. 5, pp. 1009-1022, September 2011.

[147] S. Krisshnaswamy, G. Viamontes, I. Markov, and J. Hayes, “Accurate reliability evaluation and enhancement via probabilistic transfer matrices,” *in Proc. Des. Autom. Test Eur.*, 2005, pp. 282-287.

[148] S. Krisshnaswamy, G. Viamontes, I. Markov, and J. Hayes, “Probabilistic transfer matrices in symbolic reliability analysis of logic circuits,” *ACM Trans. Design Automation of electronic systems*, vol. 13, no. 1, 2007.

[149] W. Ibrahim et al, “On the reliability of majority gates full adders,” *IEEE Trans. Nanotechnology*, vol. 7, no. 1, pp. 56-67, January 2008.

[150] N. Mohyuddin. E. Pakbaznia, and M. Pedram, “Probabilistic error propagation in logic circuits using boolean difference calculus,” *in IEEE Intl. Conf. on Computer Design*, pp. 7-13, 2008.

[151] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang and F. Lombardi, “A stochastic computational approch for accurate and efficient reliability evaluation,” *IEEE Trans. Computers*, to be published.

[152] S. Sivaswamy, K. Bazargan, and M. Riedel, “Estimation and optimization of reliability of noisy digital circuits,” *in Intl. Symp. on Quality of Electronic Design*, pp. 213-219, 2009.

[153] 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 Syst.*, vol. 28, no. 1, pp. 74-86, 2009.

[154] M. R. Choudhury and K. Mohanram, “Reliability analysis of logic circuits,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 3, pp. 392-405, 2009.

[155] C. Chen and Y. Mao, “A statistical reliability model for single-electron threshold logic,” *IEEE Trans. Electron Devices*, vol. 55, no. 6, pp. 1547-1553, 2008.

[156] C. Chen, “Reliability-driven gate replication for nanometer-scale digital logic,” *IEEE Trans. Nanotechnology*, vol. 6, no. 3, pp. 303-308, May 2007.

[157] J. von Neumann, “Probabilistic logics and the synthesis of reliable organisms from unreliable components,” in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds. NJ: Princetion Univ. Press, 1956, pp. 43-98.

[158] E. M. Sentovich et al., “SIS: A system for sequential circuit synthesis,” Univ. California, Berkeley, UCB/ERL M92/41, 1992.

[159] R. E. Bryant, “Graph-based algorithms for Boolean function manipulation,” *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[160] K. S. Brace, R. L. Rudell, and R. E. Bryant, “Efficient implementation of a BDD package,” *in Proc. 27th ACM/IEEE DAC*, 1990, pp. 40–45.

[161] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*. JohnWiley & Sons, 2008.

[162] H. Chen and J. Han, “Stochastic computational models for accurate reliability evaluation of logic circuits,” *in Proc. Great Lakes Symposium on VLSI*, pp. 61-66, 2010.

APPENDICES

Appendix A: The Verilog Code of SET-turnstile

```
`include "discipline.h"
`include "constants.h"

module turnstile(g,b,nn5);

        inout g,b,nn5;
        electrical g,b,nn5;

analog function real SIGN;
        input x;
        real x;
            begin
                if (x<0)
                    SIGN = -1;
                else
                    SIGN = 1;
            end
    endfunction

parameter real CT=1e-18;
parameter real Cg=0.5e-18;
parameter real Cn=10e-18;
parameter real RT=1e6;

real n3,q3,n5,q5;
real Vn4,Vn3,Vn2;
real Vg,Vb,Vn5;
real Ce;
real Vc,V21,V43;

analog
begin
        Vg=V(g);
        Vb=V(b);


        q3=-n3*`P_Q;
        q5=-n5*`P_Q;
        Ce=(CT*Cn/(CT+2*Cn)+Cg)*CT/(CT*Cn/(CT+2*Cn)+Cg+CT);
        Vc=0.5*`P_Q/(CT+Ce);
        Vn5=(V(b)+2*Cg*V(g)/CT+(4*CT+4*Cg)*q5/(CT*CT)+2*q3/CT)*CT*CT/(4*CT*Cn+CT*CT+4*Cn*Cg+2*CT*Cg);
        Vn2=(3*CT*Cn+CT*CT+2*Cn*Cg+CT*Cg)*Vn5/(CT*CT)-(3*CT+2*Cg)*q5/(CT*CT)-q3/CT-Cg*Vg/CT;
        Vn4=(Cn+CT)*Vn5/CT-q5/CT;
        Vn3=(2*Cn+CT)*Vn5/CT-2*q5/CT;
        V43=Vn4-Vn3;
        V21=Vn2-Vb;

        @(cross(V21-Vc,+1))
                begin
                n3=n3-SIGN(Vb);
                end
        @(cross(V43-Vc,+1))
                begin
                n5=n5-SIGN(Vb);
                n3=n3+SIGN(Vb);
                end
V(nn5) <+ Vn5;
 end
endmodule
```

Appendix B: The main part of the ER model (matlab code)

```
function [Req0,Req1,PC]=Req1(KA,PA,KB,PB,KC,ReqA0,ReqA1,ReqB0,ReqB1,rA,rB,rc,x)        % PA and PB are useless in this program

ra0=(ReqA0+rA-1)/(2*rA-1);
rb0=(ReqB0+rB-1)/(2*rB-1);
ra1=(ReqA1+rA-1)/(2*rA-1);
rb1=(ReqB1+rB-1)/(2*rB-1);

KAB=Kx(KA,KB,KC,x);     % Calculate K_AB depend on Gate Type

if(KA==1||KB==1||KA==0||KB==0)
    ORAB=0;

    else
        OAB=Cor(KA,KB,KAB);
        ReqA=[1-KA,KA]*[ReqA0,ReqA1]';
        ReqB=[1-KB,KB]*[ReqB0,ReqB1]';
        ORAB=(2*ReqA-1)*(2*ReqB-1)*(OAB);
end

rab00=ra0*rb0+ORAB*sqrt(ra0*(1-ra0)*rb0*(1-rb0));
rab01=ra0*rb1+ORAB*sqrt(ra0*(1-ra0)*rb1*(1-rb1));
rab10=ra1*rb0+ORAB*sqrt(ra1*(1-ra1)*rb0*(1-rb0));
rab11=ra1*rb1+ORAB*sqrt(ra1*(1-ra1)*rb1*(1-rb1));

L1=RM1(ra0,ra1,rb0,rb1,rab00,rab01,rab10,rab11);
L2=RM(rA,rB,rA*rB);

X=[1-(KA+KB-KAB),KB-KAB,KA-KAB,KAB];    % The error-free input vector
Y=X*(L1*L2);                            % Error-free input vector thoughout dummy buffers

switch(x)
   case{1},       % x=1 NAND Gate
      RNAND=[rc   rc   rc   1-rc
           rc   rc   rc   1-rc
           rc   rc   rc   1-rc
           1-rc 1-rc 1-rc rc];

      RPNAND=[rc   rc   rc   1-rc];
      PC=Y*RPNAND';

      REQ=(L1*L2).*RNAND*[1 1 1 1]';

      Req1=X(1:3)*REQ(1:3)/sum(X(1:3));
      Req0=REQ(4);

   case{2},       % x=2 AND Gate
      RAND=[ rc   rc   rc   1-rc
           rc   rc   rc   1-rc
           rc   rc   rc   1-rc
           1-rc 1-rc 1-rc rc  ];

      RPAND=[1-rc   1-rc   1-rc   rc];
      PC=Y*RPAND';

      REQ=(L1*L2).*RAND*[1 1 1 1]';

      Req0=X(1:3)*REQ(1:3)/sum(X(1:3));
      Req1=REQ(4);
   case{3},       % x=3 NOR Gate
      RNOR=[ rc     1-rc   1-rc   1-rc
           1-rc   rc     rc     rc
           1-rc   rc     rc     rc
           1-rc   rc     rc     rc];

      RPNOR=[rc   1-rc   1-rc   1-rc];
      PC=Y*RPNOR';
```

```
    REQ=(L1*L2).*RNOR*[1 1 1 1]';

    Req0=X(2:4)*REQ(2:4)/sum(X(2:4));
    Req1=REQ(1);
  case{4},       % x=4 OR Gate
    ROR=[  rc     1-rc   1-rc   1-rc
          1-rc   rc     rc     rc
          1-rc   rc     rc     rc
          1-rc   rc     rc     rc];

    RPOR=[1-rc   rc   rc   rc];
    PC=Y*RPOR';

    REQ=(L1*L2).*ROR*[1 1 1 1]';

    if (sum(X(2:4))==0)
       Req1=mean(REQ(2:4));
    else
    Req1=X(2:4)*REQ(2:4)/sum(X(2:4));
    end

    Req0=REQ(1);
  case{7},       % x=7 XOR Gate
    RXOR=[ rc     1-rc   1-rc   rc
          1-rc   rc     rc     1-rc
          1-rc   rc     rc     1-rc
          rc     1-rc   1-rc   rc];

    RPXOR=[1-rc   rc   rc   1-rc];
    PC=Y*RPXOR';

    REQ=(L1*L2).*RXOR*[1 1 1 1]';

    Req0=[X(1),X(4)]*[REQ(1),REQ(4)]'/sum([X(1),X(4)]);
    Req1=[X(2),X(3)]*[REQ(2),REQ(3)]'/sum([X(2),X(3)]);
end

end
```

Appendix C: Copyright Information

In reference to IEEE copyright material which is used with permission in this thesis, the IEEE does not endorse any of University of Windsor's products or services. Internal or personal use of this material is permitted.

Appendix D: List of Publications

- Ran Xiao and Chunhong Chen, "Single-Electron Tunneling based Turnstiles: Modeling and Applications," in *Proceedings of 2013 IEEE International Conference on Nanotechnology (IEEE-Nano'13)*, Beijing, China, August 2013.
- Ran Xiao and Chunhong Chen, "Statistical Delay Modeling for Single-Electron based Circuits," revised and resubmitted to *IEEE Transaction on Nanotechnology*.
- Chunhong Chen and Ran Xiao, "A Fast and Recursive Model for Reliability Analysis with Combinational Circuits," submitted to *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*.
- Ran Xiao, Chunhong Chen and H. K. Kwan, "Implementation of the Conscience Mechanism using Single-Electron Transfer in Competitive Learning," submitted to *ISCAS 2014*.

VITA AUCTORIS

NAME:	Xiao, Ran
PLACE OF BIRTH:	Beijing, China
YEAR OF BIRTH:	1989
EDUCATION:	High School affiliated to Renmin University of China, Beijing, China, 2007
	Beijing Institute of Technology, B.Sc., Beijing, China, 2011
	University of Windsor, M.Sc., ON, Canada, 2013