STATUS OF THESIS

Title of thesis	Enhancement of Markov Random Field Mechanism to Achieve Fault-Tolerance in Nanoscale Circuit Design

I, JAHANZEB ANWER

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

____

____

The contents of the thesis will remain confidential for - years.

Remarks on disclosure:

____

____

	Endorsed by
____________________	____________________
Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor:
216, Islam Block, Azam Gardens, Multan Road, Lahore, Pakistan	Dr. Nor Hisham Bin Hamid
Date : 28-03-2011	Date : 28-03-2011

UNIVERSITI TEKNOLOGI PETRONAS

ENHANCEMENT OF MARKOV RANDOM FIELD MECHANISM TO ACHIEVE FAULT-TOLERANCE IN NANOSCALE CIRCUIT DESIGN

By

JAHANZEB ANWER

The undersigned certify that they have read, and recommend to the Postgraduate Studies Program for acceptance of this thesis for the fulfillment of the requirements for the degree stated.

Signature:	______________________________
Main Supervisor:	Dr. Nor Hisham Bin Hamid
Signature:	______________________________
Co-Supervisor:	Dr. Vijanth Sagayan Asirvadam
Signature:	______________________________
Head of Department:	Dr. Nor Hisham Bin Hamid
Date:	28-03-2011

ENHANCEMENT OF MARKOV RANDOM FIELD MECHANISM TO ACHIEVE FAULT-TOLERANCE IN NANOSCALE CIRCUIT DESIGN

By

JAHANZEB ANWER

A Thesis
Submitted to the Postgraduate Studies Programme
as a Requirement for the Degree of

MASTER OF SCIENCE
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
BANDAR SERI ISKANDAR,
PERAK

MARCH 2011

DECLARATION OF THESIS

Title of thesis	Enhancement of Markov Random Field Mechanism to Achieve Fault-Tolerance in Nanoscale Circuit Design

I, JAHANZEB ANWER

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

	Witnessed by
Signature of Author	Signature of Supervisor
Permanent address: 216, Islam Block, Azam Gardens, Multan Road, Lahore, Pakistan.	Name of Supervisor: Dr. Nor Hisham Bin Hamid
Date: 28-03-2011	Date: 28-03-2011

DEDICATION

I dedicate this research work to my parents ***Muhammad Anwar and Kalsoom Anwar*** and to my brothers ***Shoaib Anwar and Shakeeb Anwar***.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank ***Almighty Allah*** to give me the patience, ingenuity and environment to complete this research work. I would like to thank my supervisor, ***Dr. Nor Hisham B. Hamid,*** and co-supervisor, ***Dr. Vijanth Sagayan Asirvadam***, for their help and guidance, which they provided me with great patience and diligence. I would also like to thank ***Mr. Usman Khalid,*** my good friend, for his moral and practical support throughout the degree.

At the same time, I would like to thank my dear friends ***Mr. Aamir Amin***, ***Mr. Anas Mohammad Nazlee, Mr. Asif Ali Waggan, Mr. Asim Qureshi, Mr. Bilal Munir Mughal, Ms. Goay Xuan Hui, Mr. Muhammad Imran Khan, Ms. Mifrah Ahmed, Mr. Mubasshir Rehman, Mr. Narinderjit Singh, Mr. Nazabat Hussain, Mr. Sohail Safdar, and Ms. Varsha Jha.*** They have been a source of love and moral support during my studies at Universiti Teknologi PETRONAS (UTP). At last, I owe a deep debt of gratitude to ***Universiti Teknologi PETRONAS (UTP)*** for providing me the monetary resources and infrastructure to complete this research work.

ABSTRACT

As the MOSFET dimensions scale down towards nanoscale level, the reliability of circuits based on these devices decreases. Hence, designing reliable systems using these nano-devices is becoming challenging. Therefore, a mechanism has to be devised that can make the nanoscale systems perform reliably using unreliable circuit components. The solution is fault-tolerant circuit design. Markov Random Field (MRF) is an effective approach that achieves fault-tolerance in integrated circuit design. The previous research on this technique suffers from limitations at the design, simulation and implementation levels. As improvements, the MRF fault-tolerance rules have been validated for a practical circuit example. The simulation framework is extended from thermal to a combination of thermal and random telegraph signal (RTS) noise sources to provide a more rigorous noise environment for the simulation of circuits build on nanoscale technologies. Moreover, an architecture-level improvement has been proposed in the design of previous MRF gates. The re-designed MRF is termed as Improved-MRF.

The CMOS, MRF and Improved-MRF designs were simulated under application of highly noisy inputs. On the basis of simulations conducted for several test circuits, it is found that Improved-MRF circuits are 400 whereas MRF circuits are only 10 times more noise-tolerant than the CMOS alternatives. The number of transistors, on the other hand increased from a factor of 9 to 15 from MRF to Improved-MRF respectively (as compared to the CMOS). Therefore, in order to provide a trade-off between reliability and the area overhead required for obtaining a fault-tolerant circuit, a novel parameter called as ‘Reliable Area Index’ (RAI) is introduced in this research work. The value of RAI exceeds around 1.3 and 40 times for MRF and Improved-MRF respectively as compared to CMOS design which makes Improved-MRF to be still 30 times more efficient circuit design than MRF in terms of maintaining a suitable trade-off between reliability and area-consumption of the circuit.

ABSTRAK

Semakin dimensi *MOSFET* mengecil kepada skala nano, kebolehpercayaan litar berasaskan peranti ini semakin berkurangan. Oleh itu, merekabentuk sistem yang mempunyai kebolehpercayaan menggunakan peranti-nano menjadi semakin mencabar. Suatu mekanisma harus dicipta untuk menjadikan sistem-sistem skala nano berfungsi dengan bolehpercayaan menggunakan komponen litar yang tanpa bolehpercayaan. Penyelesaian masalah ini ialah rekabentuk litar dengan toleransi-kesalahan. *Markov Random Field* (*MRF*) ialah kaedah berkesan yang mencapai kebolehan toleransi-kesalahan dalam rekabentuk litar bersepadu. Kajian sebelum ini terhadap teknik tersebut menghadapi kekurangan dari had-had pada rekabentuk, simulasi dan tahap-tahap implementasi. Sebagai pembaikan, garis panduan toleransi-kesalahan MRF telah dibuktikan untuk litar contoh yang praktikal. Rangka kerja simulasi ditambah daripada suhu kepada kombinasi suhu dengan sumber gangguan *Random Telegraph Signal* (*RTS*) memberikan persekitaran gangguan yang lebih menyeluruh untuk simulasi litar yang dibina menggunakan teknologi skala-nano. Pembaikan pada tahap rekaan telah diusulkan di dalam rekabentuk gate-gate *MRF* terdahulu. *MRF* yang telah direkabentuk semula diistilahkan sebagai *Improved-MRF*.

Rekaan *CMOS*, *MRF* dan *Improved-MRF* telah disimulasikan dibawah aplikasi dengan input-input bergangguan tinggi. Simulasi dijalankan keatas beberapa litar ujian, didapati bahawa litar *Improved-MRF* adalah 400 berbanding litar *MRF* yang hanya 10 kali ganda lebih toleransi-gangguan daripada alternatif *CMOS*. Bilangan transistor bertambah dari factor 9 kepada 15 untuk *MRF* ke *Improved-MRF* (berdasarkan dengan *CMOS*). Imbangan antara kebolehpercayaan dan keluasan yang diperlukan untuk menghasilkan litar toleransi-kesalahan, satu parameter baru yang dipanggil “*Reliable Area Index* (*RAI*)” diperkenalkan di dalam kajian ini. Nilai *RAI* melebihi sekitar 1.3 dan 40 kali ganda untuk *MRF* dan *Improved-MRF* berbanding rekaan *CMOS* menjadikan rekabentuk litar *Improved-MRF* 30 kali ganda lebih efisyen berbanding *MRF* dalam mengekalkan keseimbangan yang bersesuaian antara kebolehpercayaan dan keluasan yang digunakan di dalam litar.

In compliance with the terms of the Copyright Act 1987 and the IP Policy of the university, the copyright of this thesis has been reassigned by the author to the legal entity of the university,

Institute of Technology PETRONAS Sdn Bhd.

Due acknowledgement shall always be made of the use of any material contained in, or derived from, this thesis.

© Jahanzeb Anwer, 2011
Institute of Technology PETRONAS Sdn Bhd
All rights reserved.

TABLE OF CONTENTS

STATUS OF THESIS i
APPROVAL PAGE ii
TITLE PAGE iii
DECLARATION iv
DEDICATION v
ACKNOWLEDGEMENTS vi
ABSTRACT vii
ABSTRAK viii
COPYRIGHT PAGE ix
LIST OF TABLES xiv
LIST OF FIGURES xv

CHAPTER 1 – INTRODUCTION 1

1.1 Motivation 1
1.2 Fault-Tolerant Circuit Design 3
1.3 Research Objectives 4
1.4 Scope of the Research Work 5
1.5 Contributions of the Thesis 6
1.6 Thesis Organization 6

CHAPTER 2 – FAULT-TOLERANCE IN INTEGRATED CIRCUIT DESIGN 8

2.1 An Overview of Fault-Tolerance 8
2.2 Types of Faults 9
    2.2.1 Permanent Faults 9
    2.2.2 Transient Faults 10

2.2.2.1 Noises Responsible for Transient Errors in Future Nanoscale Circuits.. 11
2.3 The Research on Fault-Tolerance... 13
2.3.1 Reliability-Evaluation Schemes.. 13
2.3.1.1 Bayesian Probabilistic Error Model............................. 15
2.3.1.2 Probabilistic Transfer Matrices (PTM) Model............. 16
2.3.1.3 Probabilistic Gate Model (PGM).................................. 16
2.3.1.4 Boolean Difference Error Calculator (BDEC)............. 17
2.3.1.5 Comparison of Reliability-Evaluation Schemes........... 17
2.3.2 Architecture Level Solutions.. 18
2.3.2.1 Redundancy... 18
2.3.2.2 Markov Random Field (MRF) Model.......................... 19
2.3.2.3 Comparing Redundancy and MRF.............................. 20
2.3.3 CAD Tools Development... 22
2.3.3.1 Multiple Environment and Multiple Error Simulation..... Tool for Analysis of Reliability (MEMESTAR)...........22
2.3.3.2 AutoPTMate.. 23
2.4 Markov Random Field.. 23

CHAPTER 3 – MARKOV RANDOM FIELD BASED CIRCUIT DESIGN......... 25

3.1 MRF Graph Theory...25
3.2 MRF Mathematical Model.. 27
3.2.1 Joint Probability.. 27
3.2.1.1 Detailed Computation Procedure................................ 28
3.2.1.2 Design Principle of Joint Probability........................... 32
3.2.1.3 Foundation of Joint Probability Principle..................... 33
3.2.2 Marginal Probability.. 34
3.2.2.1 Detailed Computation Procedure................................ 35
3.2.2.2 Marginal Probability Power Dissipation Principle....... 40
3.2.3 Combined Application of Joint and Marginal Probability................... Requirements... 41
3.3 MRF Implementation Model... 41

3.3.1 Mapping Design Principle of Joint Probability on Digital Hardware.......... 41

3.3.1.1 How the MRF Conversion Principles Enforce Maximum Joint Probability.......... 44

3.3.2 Mapping Marginal Probability Power Dissipation Principle on Digital Hardware.......... 44

3.4 Transistor-Count Comparison of CMOS and MRF-CMOS Designs.......... 46

3.5 Discussion.......... 46

CHAPTER 4 – VALIDATING MRF CIRCUIT'S PERFORMANCE.......... 48

4.1 Simulation Setting.......... 48

4.1.1 CMOS Technology Model.......... 48

4.1.2 Noise Models.......... 49

4.1.2.1 Thermal Noise.......... 49

4.1.2.2 Random Telegraph Signal (RTS) Noise.......... 51

4.1.2.3 Difference Between Thermal and RTS Noise Injection Models.......... 52

4.1.3 CMOS Inverter v/s MRF-CMOS Inverter.......... 53

4.1.4 CMOS NAND v/s MRF-CMOS NAND.......... 55

4.1.5 Simulations with Improved-MRF Design.......... 55

4.1.6 Quantifying MRF Noise-Tolerance.......... 55

4.1.7 Transistor-Count of CMOS, MRF and Improved-MRF Designs.......... 58

4.1.8 Circuit's Reliability v/s Transistor-Count.......... 58

4.1.9 CMOS Technology Independence of MRF Design.......... 60

CHAPTER 5 – CONCLUSION.......... 62

5.1 Synopsis of the Thesis.......... 62

5.2 Suggested Future Work.......... 64

REFERENCES.......... 65

APPENDIX A – LOGIC DIAGRAMS OF TEST CIRCUITS............................... 71
APPENDIX B – SIMULATIONS OF TEST CIRCUITS... 74
APPENDIX C – PUBLICATIONS... 80

LIST OF TABLES

Table 3.1 Logic Compatibility Function of NAND……………………………. 30
Table 3.2 Clique Energy Functions for NOT and NOR gates………………….. 31
Table 3.3 Node combinations having maximum joint probability……………... 32
Table 3.4 Probability Distribution Functions for inputs and cliques…………… 36
Table 3.5 Comparison of transistor-count in different circuit designs…………. 46
Table 4.1 CMOS 32 nm predictive technology parameters……………………..49
Table 4.2 Parameter setting of thermal noise function…………………………. 50
Table 4.3 RMS voltage variation (mV) of circuits against input variation of…….. 600 mV………………………………………………………….57
Table 4.4 Number of transistors used in CMOS, MRF and Improved-MRF …….. Designs…………………………………………………………..59
Table 4.5 RAI Values for CMOS, MRF and Improved-MRF designs……….... 60
Table 4.6 RMS Variation of Inverter output using different CMOS …………….. Technologies…………………………………………………………61

LIST OF FIGURES

Fig. 2.1 Fault-tolerance categories and their sub-divisions……………………14
Fig. 2.2 A conceptual idea of Bayesian Network error-probability……………… Calculation……………………………………………………… 15
Fig. 2.3 NAND's Probabilistic Transfer Matrix……………………………… 16
Fig. 2.4 Block diagram of Boolean Difference Error Calculator……………... 17
Fig. 2.5 The mechanism of redundancy……………………………………… 19
Fig. 2.6 (a) CMOS Inverter……………………………………………………..20
Fig. 2.6 (b) MRF-CMOS Inverter…………………………………………………20
Fig. 3.1 MRF Neighbourhood System……………………………………... 26
Fig. 3.2 (a) Test Circuit……………………………………………………………29
Fig. 3.2 (b) Its MRF dependence graph………………………………………... 29
Fig. 3.3 Joint Probability graph of Inverter……………………………………33
Fig. 3.4 (a) Inverter cascade……………………………………………………… 34
Fig. 3.4 (b) It's dependence graph……………………………………………... 34
Fig. 3.5 (a) Marginal probability graph of x_4 ……………………………………. 39
Fig. 3.5 (b) Marginal probability graph of x_5……………………………………. 39
Fig. 3.5 (c) Marginal probability graph of x_6……………………………………. 39
Fig. 3.5 (d) Marginal probability graph of x_7……………………………………. 39
Fig. 3.6 Marginal probability variation of x_7………………………………… 40
Fig. 3.7 (a) Inverter with AND gates……………………………………………...43
Fig. 3.7 (b) Inverter with NAND gates…………………………………………43
Fig. 3.8 MRF NAND gate……………………………………………………..44
Fig. 3.9 (a) Improved-MRF NOT…………………………………………………46
Fig. 3.9 (b) Improved-MRF NAND……………………………………………… 46
Fig. 4.1 Thermal noise representation………………………………………51
Fig. 4.2 RTS noise representation in NMOS drain current……………………52
Fig. 4.3 RTS noise effect in O/P v/s I/P waveform of inverter………………..52
Fig. 4.4 Thermal and RTS noise inclusion mechanisms ……………………...53

Fig. 4.5 Simulations of CMOS Inverter v/s MRF Inverter …………………… 54
Fig. 4.6 Simulations of CMOS NAND v/s MRF NAND ..…………………… 56
Fig. 4.7 Simulations of MRF Inverter v/s Improved-MRF Inverter...………... 57

CHAPTER 1

INTRODUCTION

This thesis explores the utilization of fault-tolerance principles in nanoscale circuit design by implementing probabilistic computation in digital hardware. The fault-tolerance criterion is described particularly for the future nanoscale MOSFETs which are not yet available though their performance can be predicted by their SPICE models. The motivation of this research is provided in Sec. 1.1. After providing the brief description of the issues related to nanoscale design, the problem statement (Sec. 1.2) and research objectives are formulated (Sec. 1.3). The scope of the research is followed in Sec. 1.4. Finally, the contribution and organization of the thesis will be explained in Sec. 1.5 and 1.6 respectively.

1.1 Motivation

The MOSFET (Metal oxide semiconductor field-effect transistor) is the basic element of integrated circuit design. With MOSFET scaling, the circuit design can be improved in certain ways. The premier advantage obtained is the increase in device density, which means that more transistors can be accommodated in the similar chip area e.g. Intel Processor-45nm (QX9650) chip contains 410 million transistors (in an area of 107 mm^2) as compared to Intel Processor-65nm (QX6850) chip which contains 341 million transistors (into a larger area of 143 mm^2) [1]. Therefore, the circuit designers can add more functional blocks while utilizing the same space.

Another major advantage with transistor scaling is the improvement in switching speed of the transistors. With downscaling of device dimensions, the gate capacitance is also scaled which results in decrease in the *RC* delay of the transistor [2, 3]. As a result, switching speed and operating frequency of the circuits increase as well e.g. processor and clock speeds of Intel microprocessors have an exponentially increasing trend [3].

The manufacturing cost of an integrated circuit drops with the transistor scaling. According to Eq. (1.1), the cost of an integrated circuit is directly proportional to the cost of a die [4].

$$\text{Cost of an Integrated Circuit} \propto \frac{\text{Cost of a Die}}{\text{Final Test Yield}} \tag{1.1}$$

where, the cost of a die is a function of its area as described in Eq. (1.2).

$$\text{Cost of a Die} = f\ (\text{Die Area})^4 \tag{1.2}$$

It is evident that by decreasing transistor dimensions, the die area decreases. According to Eq. (1.1) and Eq. (1.2), the cost of a die decreases with the area contraction thereby decreasing the cost of an integrated circuit too. Hence, increased device density, higher switching speed and decreased cost of an integrated circuit are the reasons that motivate researchers and semiconductor industries to continuously scale down transistor dimensions. At this point of time, the circuit design has entered into nanoscale era.

As the circuit design enters into the nanoscale regime, the literature reports the future demise of Moore’s Law [5-8] which originally stated that the *number of transistors in an integrated circuit will double every 18 months*. According to [9], this law becomes invalid as the transistor technology is approaching the atomistic dimensions. As a result, extra leakage currents appear, standard circuit design does not work and the noise voltage augments [9]. With the combined effect of signal (voltage) level *decrease* and noise level *increase*, the signal to noise ratio degrades. In order to build a reliable logic circuit, it has to maintain a certain amount of signal to noise ratio which will be violated in the implementation of future CMOS nanoscale technologies [9].

It is widely reported that the nano-devices are un-reliable as they are more sensitive to the radiation effects (e.g. radioactive decay and cosmic rays), high temperature, electromagnetic interference, parameter variations etc [9-16]. At the circuit design level, the reliability is distorted in the form of increased error-rate observed in the computing systems [11, 13, 14, 16]. The reliability degradation is undoubtedly, the outcome of transistor technology scaling which results in construction of un-reliable nano-devices [9-14, 16].

There are two possible solutions to deal with the un-reliability of nanoscale devices. As one option, such materials/devices are need to be formed that can work reliably at nanoscale dimensions. Research is already progressing on different physical theories for developing such equivalents of MOSFETs e.g. quantum, spin and magnetic theories [12]. The second approach is to make use of MOSFET while building a fault-tolerant circuit architecture i.e. designing reliable circuits using unreliable nano-devices [6, 11, 16]. Therefore, to make the most use of MOSFET, the research is progressing extensively towards fault-tolerant circuit design before considering MOSFET's alternatives.

Apart from fault-tolerance solutions, there is a need to develop reliability-evaluation techniques that can provide a measurement of the level of fault-tolerance of a circuit. In order to evaluate the reliability of large circuits, the computer-aided design (CAD) tools should also be developed that could automate the reliability-evaluation and fault-tolerant design processes. These softwares should be able to validate of fault-tolerance solutions as well. Hence, the reliability-evaluation techniques, fault-tolerance solutions and the CAD tools development will all be discussed in detail in Chapter 2.

1.2 Fault-Tolerant Circuit Design

The reliability of integrated circuits comes in question when the nanoscale technologies are used for the circuit design [6, 9-14, 16]. Following the trend of transistor scaling, the development of fault-tolerant systems is increasing as the circuit design is susceptible to increased error-rates for future nanoscale technologies [10, 16, 17-19]. The industries who want to develop high-speed computers/processors are also keen to utilize fault-tolerance mechanisms that can ensure reliable computation for their extremely reliable applications [20]. Hence, there is a need to develop a circuit design paradigm that can ensure reliable operation of digital devices using unreliable transistor technologies.

The fault-tolerance solutions consist of redundancy [21, 22] and Markov Random Field (MRF) [23] as the two major available options. Between these two schemes, MRF was proved to be better than redundancy in terms of reliability, error-handling capacity and area efficiency (as discussed in Chapter 2). A thorough literature review on MRF shows that this technique suffers from weaknesses at the design, simulation and implementation stages. At the design level, fault-tolerance rules derived from the mathematical model of MRF had not been validated by their application on a practical circuit example. At the simulation stage, the noise framework includes only thermal noise source as the potential source of errors though neglecting other kinds of deep sub-micron (DSM) noise*. Moreover, the transistor technology used in the simulations (conducted in the previous research) was 70 nm whereas more advanced technology models like 32 nm are now available [25]. At the implementation level, there is no relation developed between the marginal probability power dissipation principle and its application on digital design. The development of this relation was necessary so it could be used as a standard fault-tolerance principle by circuit designers. Hence, there is a possibility of improvements in this technique on the three levels as described. An Improved-MRF architecture could be developed based on these improvements. Therefore, these limitations direct us to formulate the research objectives in the following section.

1.3 Research Objectives

The purpose of this research is to improve the Markov Random Field fault-tolerance mechanism to develop a more noise immune circuit design as compared to the one in literature. Accordingly, the following objectives have to be attained.

---

*DSM technologies refers to those MOSFETs having physical gate length less than 100 nm but greater than 10 nm [24].

- To validate the MRF fault-tolerance rules stated in the previous literature (on a test circuit). The results obtained in the case study will be compared with the ones in literature.

- To extend the simulation framework from thermal to a combination of thermal and random telegraph signal (RTS) noises. The process would involve modelling these noises according to their mathematical models.

- To develop RTS noise models of transistors for the 32 nm technology which will be used to perform simulations required for noise-tolerance verification of MRF design.

- To propose improvement in the previous MRF circuit architecture based on the power dissipation principle of MRF. This improvement would enhance the MRF design of basic logic gates to a more noise-immune architecture design.

In this thesis, the above-mentioned objectives will be obtained sequentially in order to reach up to the final MRF proposed architecture.

1.4 Scope of the Research Work

The scope of the research is limited to the following parameters.

- The noise-tolerance principle of MRF would be investigated at the design and simulation levels only. The layout and fabrication of MRF circuits will not be covered due to the time constraints of this research work.

- The design mechanism of MRF is limited to combinational circuits only.

- The noise-tolerance mechanism is designed to counter thermal and random telegraph signal noises only being the major sources of errors in future deep submicron devices. Other nanoscale noise sources have not been modelled in this dissertation.

- The main focus of this thesis would be the MRF design of universal gates only. The strategy is to make the logic gates individually noise-tolerant so that when they will be used as parts of a bigger circuit, the overall circuit would automatically remain noise-tolerant. Although the design of universal gates only is shown in this thesis, bigger circuits like decoder, multiplexer and adders are simulated as well and their noise tolerance is quantified in the results chapter (Chapter 4).

1.5 Contributions of the Thesis

The major contributions of this research work are as follows.

- The Improved-MRF technique which has been introduced in this thesis has the potential to replace the previous MRF scheme in terms of attaining better trade-off between circuit's reliability and area-efficiency.

- The computation procedures of MRF's mathematical framework have been described in detail (with implementation on a practical circuit example) which makes it easy to understand the principle of probabilistic computation for digital circuits.

- The thesis provides the researchers a way to model thermal and RTS noise sources with implementation on the software Cadence. This is a useful contribution as the noise inclusion in the digital circuit design is not the built-in part of Cadence. Moreover, the RTS noise models that have been developed in this research could help in conducting advanced (nanoscale) circuit simulations by researchers working in the field of integrated circuit design.

1.6 Thesis Organization

The thesis is divided into five chapters.

- Chapter 1 aims to provide the motivation for conducting research on fault-tolerant circuit design followed by objectives and contributions of this research work.

- Chapter 2 sums up the literature review of major fault-tolerance techniques which are proposed or in use today. By the end of this chapter, we will filter out Markov Random Field as the technique for onwards research (based on the critical analyses of all the techniques discussed in this chapter).

- Chapter 3 is based on describing mathematical and implementation models of Markov Random Field. It consists of detailed computation procedures for mathematical analysis based on the outlines provided in the literature. The results obtained by this computation procedure will be matched with the results proposed in the literature.

- Chapter 4 lists the simulations performed on the previous and the Improved-MRF designs. The purpose is to prove the worth of noise immune design in the presence of target noise sources.

- Chapter 5 provides the conclusion of this thesis.

CHAPTER 2

FAULT-TOLERANCE IN INTEGRATED CIRCUIT DESIGN

In the previous chapter, a strong motivation was provided to conduct the research on fault-tolerance in integrated circuit design. This chapter is destined to give an overview of the major fault-tolerance techniques and the noise models proposed. By the end of this chapter, a technique will be chosen that best serves the fault-tolerance purpose and has the potential to be carried on towards critical analysis and research.

2.1 An Overview of Fault-Tolerance

Fault-tolerance has the implementation at either the software or hardware level. Software fault-tolerance refers to design the software programs that can tolerate the software design faults i.e. programming errors [20]. Hardware fault-tolerance, on the other hand, aims to recover from hardware faults which can be further split into component or system level fault-tolerance [20]. At the system-level, the target is to protect the system from outside noise disturbances like radiation from electromagnetic interference or packaging materials. The component level fault-tolerance aims to develop a circuit design that can tolerate the faults produced by components of a circuit like transistors or interconnects. The area of focus in this thesis would be based on achieving fault-tolerance at the component level. Unless stated otherwise, the term ‘fault-tolerance’ in this thesis would refer to component level fault-tolerance.

Fault-Tolerance in integrated circuit refers to design it in such a way that the circuit network runs satisfactorily in the presence of faults or signal noise. Hence, the fault-tolerance is implemented either to avoid errors (by absorbing signal noise) or as error-recovery mechanisms (that ensure the correct functioning of the circuit in case the error is detected) [20, 23, 24]. As the CMOS technology downscales, the error-occurrence in the system becomes undeterminable* [22-24, 26]. This non-deterministic behaviour of nano-circuits is addressed by the branch of mathematics known as ‘probability’. The reason of using probabilistic analysis is the random (or probabilistic) nature of signal noise and errors. Therefore, the fault-tolerance mechanisms in this chapter are based on probabilistic computation. Before proceeding towards fault-tolerance mechanisms, the origin of faults (that occur in a digital circuit) has to be understood so that the methodologies can be designed to heal the errors caused by these faults.

2.2 Types of Faults

A fault is a hardware defect which could occur at the gate or transistor level. An error is the manifestation of the fault [27]. The faults in an integrated circuit could be either permanent or transient.

2.2.1 Permanent Faults

The permanent faults results from hardware malfunctioning in which the device halts to operate correctly. These errors arise due to manufacturing defects or faults appeared in the hardware due to repeated use of the circuit [28]. The errors caused by the permanent faults are called ‘hard errors’. If these faults/errors occur during manufacture, they can be detected by the initial testing of the chip but if they appear during the usage of the chip, the erroneous circuit has to be replaced [29]. Hard errors are usually reproducible, consistent and easy to isolate [29].

The major techniques designed to tolerate these errors are redundancy (discussed in 2.3.1.1) and reconfiguration [21, 30]. In reconfiguration, defect-tolerance is achieved through detection of faulty components during an initial defect map phase (defect mapping is the process of finding defective locations in a chip) and excluding them during actual configuration.

---

*The word system in this context does not refer to system-level fault-tolerance but to indicate a portion of a large circuit.

2.2.2 Transient Faults

These faults or errors, as the name implies, are transient in nature and disappear after a short period of time either by themselves or with the application of the error-recovery mechanisms. To explain the possibility of self-fading of transient errors, a simple example of two-input AND gate is used. Suppose both of the inputs are at logic *0,* being the correct inputs. If for a short span of time, one of its inputs becomes erroneous and switches to logic 1, the output would remain the same (or error-free). But since the self-fading is not the common case, error-recovery systems must be designed to ensure correct logic operation of the circuit. Errors caused by transient faults are called as either transient or soft errors.

Environmental conditions like radiation effects, temperature, altitude and humidity all cause transient errors [29]. The outside disturbances like power jitter, electromagnetic interference (EMI) and ionization (due to cosmic rays or alpha particles from packaging materials) are the sources of these errors as well. A considerable amount of research on packaging materials and techniques to isolate circuits from environmental effects has already been conducted and successfully implemented [31].

The transient errors which are the result of circuit design issues include charge sharing [4, 32], charge leakage [4, 32], power source [32], crosstalk [33], thermal and random telegraph signal (RTS) noises. The architectural-level solutions that counter the charge sharing, charge leakage, power source and crosstalk noises have been successfully implemented [4, 32, 33]. But the solutions are not designed to counter thermal or RTS noises as their small magnitudes do not affect the circuit operation involving relatively high dimension transistor technologies of today.

The transient errors have a high probability of occurrence than permanent ones [34]. They give rise to single event upset (SEU) errors which cause bit flips at the circuit nodes. SEU errors are divided into detected and undetected soft errors. The undetected errors are called as silent data corruption (SDC) and the detected errors which are un-recoverable are called as (DUE) for detected unrecoverable errors. To quantify SEU errors, the term soft error rate (SER) is used. The unit of SER is failure-in-time (FIT) which represents number of errors in billion hours [27, 35]. Hence, the SER is measured in terms of SDC FIT and DUE FIT. The industry declares the SER along with its products e.g. IBM targeted 114 SDC FIT, 4,566 system-kill DUE FIT and 11,415 processor-kill DUE FIT for Power4 processors [35].

2.2.2.1 Noises Responsible for Transient Errors in Future Nanoscale Circuits

The noise sources responsible for transient errors were discussed in the previous section. Recall that the thermal and RTS are the noise sources which are left un-attended by the circuit designers and industries. The reason is the small magnitude of these noises as compared to the signal voltage levels. With the downscaling of transistor technologies, the power supply voltages are also decreasing at a fast pace e.g. 5 V (for $1\,\mu\text{m}$ technology) drops down to 1.2 V (for 50 nm technology) [36]. Hence, with the decrease in signal voltage levels, these noises are expected to cause significant downgraded performance in the future deep submicron technologies due to continuous dropping in signal to noise level strength. Therefore, in this research, the main focus would be towards providing a methodology to counter these two noise sources. Before providing solution, the nature and origin of these particular types of noises are discussed.

(a) Thermal Noise

Thermal noise (or Johnson-Nyquist noise) is generated in a circuit due to the random motion of charge carriers [37]. Its existence is inevitable as this noise will always be present in a circuit operating at temperatures above 0 K as the charge carriers cannot move at 0 K. The thermal noise voltage can be calculated by Eq. (2.1) [37].

$$v_n = \sqrt{\frac{kT}{C}} \tag{2.1}$$

where *k* is the Boltzmann constant, *T* is the temperature and *C* is the capacitance of the node. It means that the thermal noise can be approximated by just having the knowledge of node temperature and capacitance. And since temperature cannot reach a value of zero, thermal noise will always be present in a circuit. In order to conduct circuit design simulations, the thermal noise has to be modelled accurately in software tools. As a contribution, the thermal noise is modelled by a Gaussian noise source placed at the output of each gate/module which represents the total thermal noise coming out of this stage [23]. Similarly, Poisson noise model has been proposed in [38] to model the thermal noise as variation of load capacitor charge. The variation in charge (proportional to the output current) is simulated against time samples and the results are in agreement with the Gaussian noise model of this noise. A number of such modelling attempts have been made by circuit designers but the research to accurately model this noise (in nanoscale designs) is still underway.

(b) Flicker and Random Telegraph Signal Noise

The flicker (1/f) and random telegraph signal (RTS) noise in MOSFETs results from trapping or detrapping of charge carriers near the $Si\text{-}SiO_2$ interface [24, 39]. The trapped carriers limit the mobility of free carriers near the interface by Coulombic scattering [24] causing fluctuations in the MOSFET drain current. As the gate length of MOSFET decreases, the noise variation become discretised and called as random telegraph signal (RTS) noise. It is agreed that the superposition of many RTS noise sources generate flicker noise [24]. Hence, for very small MOSFETs (particularly belonging to deep submicron technologies) are believed to experience RTS noise while the larger MOSFETs ($> 5\text{-}10\,\mu m^2$) encounter flicker noise [24].

In [24, 40], noise models for both NMOS and PMOS have been developed for flicker and RTS noises. These models have been programmed in hardware description language, VerilogA and simulated by integrating their models in Cadence simulation software with the toolbox called as ‘Analog Design Environment’. For flicker noise modelling, sum-of-sinusoids method whereas for RTS noise, Monte Carlo based technique is used [40]. These models automatically add flicker or RTS noise in output current of both PMOS and NMOS (based on the mathematical models of these noises). Although these models have been designed for CMOS 350 nm and 35 nm technologies, they can be extended to other technology model simulations by modifying the programming code. In real operation of the circuit, the flicker or RTS noise frequency is quite low but for simulation purposes, an overestimate of this noise has to be provided so as to obtain reliable simulation results for future predictive CMOS technology models.

2.3 The Research on Fault-Tolerance

The research on fault-tolerance can be divided into three categories. The first category contains reliability-evaluation schemes which calculate the reliability (or output error probability) of a circuit [26, 41-44]. After developing a measure of reliability calculation, the research progresses into modifications proposed on the architecture level. Thus, the second category is named as architecture-level solutions [22-24]. In this category, the fault-tolerant design techniques are proposed, the efficiency of which can evaluated by the techniques proposed in the first category. The final approach is about developing CAD (computer aided design) based tools that can accurately simulate and provide the reliability report of a target digital circuit [45, 46]. Fig. 2.1 shows the fault-tolerance categories and their sub-divisions. Now, the fault-tolerance design schemes (with their sub-divisions) and their scope will be discussed followed by a general discussion on their applicability and effectiveness.

2.3.1 Reliability-Evaluation Schemes

Reliability is the measure of the percentage of time along which the circuit is supposed to work error-free. The premier goal towards the fault-tolerant design should be modelling noise in the nanoscale circuits. Based on these noise models, mathematical analysis could be developed that can calculate the reliability of a circuit or a system in consideration. Thus, the reliability-evaluation schemes provide a validation framework for the measurement of noise-tolerant capability of a circuit. Bayesian networks, Probabilistic transfer matrices, Probabilistic gate model and Boolean difference error calculator as the four proposed reliability-evaluation schemes covered in this category. A short comparison will be provided after briefly describing the mechanism of each of these schemes.

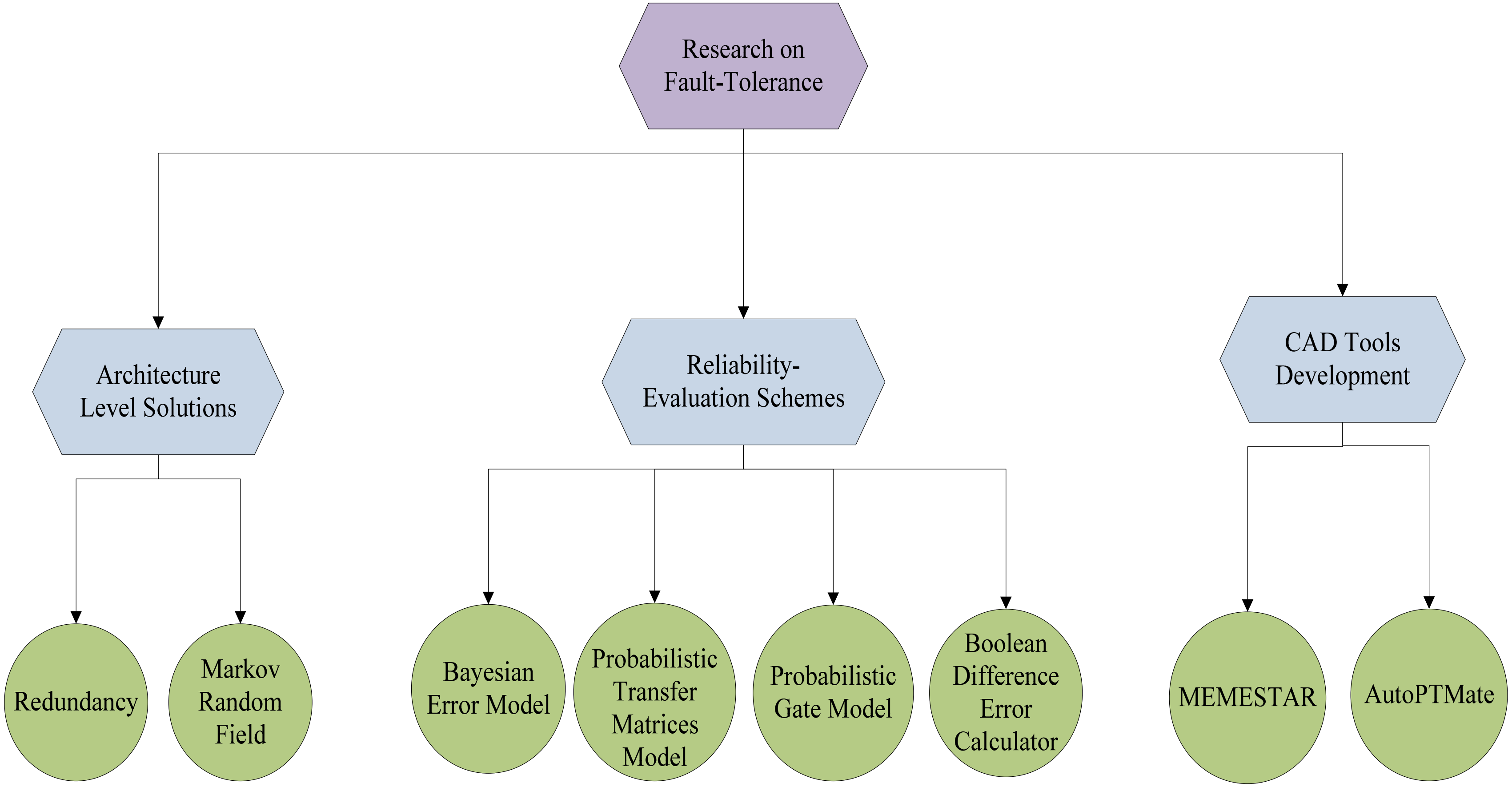

Research on Fault-Tolerance
Architecture Level Solutions
Reliability-Evaluation Schemes
CAD Tools Development
Redundancy
Markov Random Field
Bayesian Error Model
Probabilistic Transfer Matrices Model
Probabilistic Gate Model
Boolean Difference Error Calculator
MEMESTAR
AutoPTMate

Fig 2.1: Fault-Tolerance categories and their sub-divisions

2.3.1.1 Bayesian Probabilistic Error Model

Bayesian networks error model [26] computes the error-probability of the circuit by comparing difference between the error-free and error-encoded circuit outputs. Thereafter, if a mismatch is observed between the two output values; the comparator linking the two systems will output logic 1. The probability of comparator output being in logic 1 provides the error probability of the circuit. This concept has been illustrated in Fig. 2.2. In this figure, *a*, *b* and *c* are the inputs whereas *A'* and *B'* are the error-free copies of *A* and *B* respectively. As shown, the outputs from the actual circuit and its ideal copy circuit are matched using a XOR gate (comparator). *E1*=1 or *E2*=1 are the indications of an error.

The output-error probabilities have been calculated for exact and approximate inference schemes. The authors in [26] have designed an algorithm, logic induced probabilistic error model (LIPEM), and used software tools HUGIN and SMILE, for error-probability computation. For small benchmark circuits (e.g. LGSynth'93), exact inference scheme whereas for large benchmark circuits (e.g. ISCAS'85), approximate inference scheme has been used.

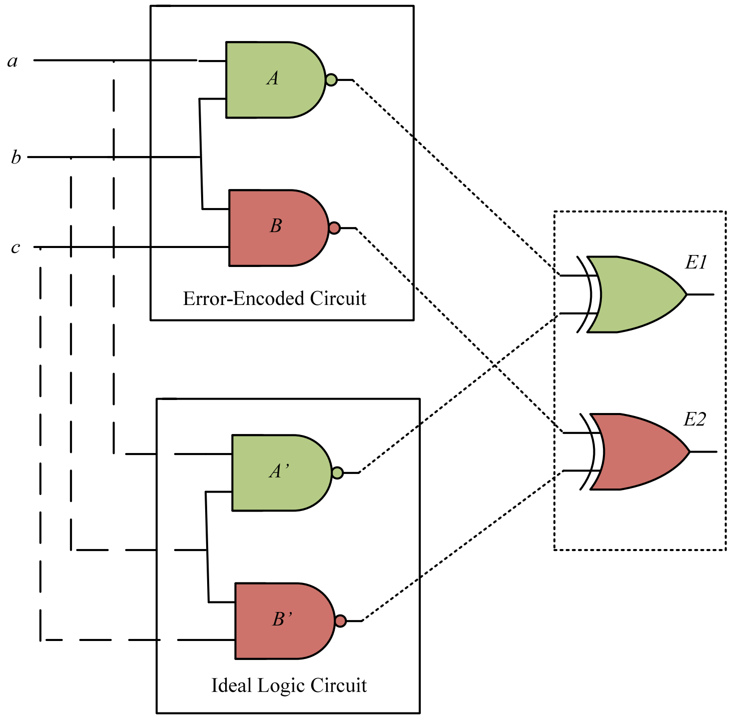


Fig 2.2: A conceptual idea of Bayesian Network error-probability calculation [26]

2.3.1.2 Probabilistic Transfer Matrices (PTM) Model

This model computes the output error probability by calculating PTM for each gate in a circuit [41]. Here, the concept of PTM calculation for NAND gate is explained using Fig. 2.3. In this figure, *p* denotes the ‘gate error probability’. For each input set, the probability is *p* for an incorrect and 1-*p* for the correct output. Hence, the PTM is a matrix representation of error probabilities for all node combinations in a network. After calculating PTMs of all gates in the circuit, the circuit is divided into stages and the PTM of the entire circuit is formed by a method that involves computing tensor products and matrix multiplications. Finally, the reliability of a circuit can be found by Eq. (2.1) [42].

$$Reliability\,(v, M, J) = \|\, v\,(M.{*}J)\,\| \tag{2.1}$$

where *v* is the input vector, *M* is the PTM of the entire circuit and *J* is the identity transfer matrix, ITM. Hence, the output error probability can be found by subtracting the reliability value from unity.

$$\begin{array}{cc} & \text{O/Ps} \\ & \begin{array}{cc} 0 & 1 \end{array} \\ \text{I/Ps} \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{bmatrix} p & 1-p \\ p & 1-p \\ p & 1-p \\ 1-p & p \end{bmatrix} \end{array}$$

Fig 2.3: NAND’s Probabilistic Transfer Matrix [41]

2.3.1.3 Probabilistic Gate Model (PGM)

Like PTM and Bayesian models, PGM [43] is based on calculating error probability of a circuit. The gate error models used in this method have been developed by Von Neumann approach and used in the PGM computation. The inputs and outputs in this scheme are considered to be independent of each other. Overall reliability (and hence output error probability) is calculated by multiplying reliabilities of each output of a circuit. Results obtained by this method have been compared with those of PTM and the output error probabilities are found to be in close comparison for both techniques. An algorithm has also been developed by the authors in [43] that automates the PGM computation process. With the development of PGM algorithm, it is now possible to calculate the reliability of large circuits.

2.3.1.4 Boolean Difference Error Calculator (BDEC)

The BDEC [44] is another error probabilistic model that claims to be better in efficiency, execution time and memory usage than PTM. The concept of BDEC is explained in Fig. 2.4. According to this figure, the inputs required by the calculator are p_i (probability of i^{th} input being in logic 1), e_i (error probability of input i), f (logic equation of the gate) and e_g (gate error probability). The output e_z is calculated by a complex mathematical model involving differential equations whereas the software SIS and MATLAB have been used for simulation purposes [44]. The research work in [44] compares BDEC with PTM and PGM, reporting close comparison of results obtained for all three techniques.

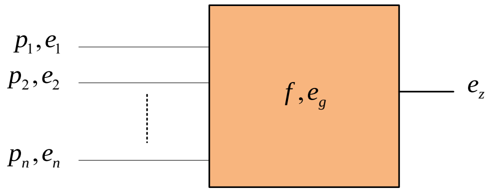


Fig 2.4: Block diagram of Boolean Difference Error Calculator [44]

2.3.1.5 Comparison of Reliability-Evaluation Schemes

Output error probability is the direct indication of fault-tolerance capability of a circuit. The output error-probabilities (for benchmark circuits) calculated by the four techniques are comparable to each other. In terms of execution time comparison, PGM though shows no results whereas BDEC is a timing efficient method is orders of magnitude than PTM and Bayesian [26, 41, 42, 44]. The next step is to automate the reliability-evaluation methods by developing their generic mathematical models followed by integrating them into software. Among the techniques described in this section, only PTM is automated i.e. a tool has been developed in [46] that calculates the reliability of a given circuit by taking the circuit netlist as its input. The authors of the remaining three techniques have also claimed to automate their reliability-evaluation methods in future as well [26, 43, 44].

2.3.2 Architecture-Level Solutions

As compared to the reliability-evaluation schemes, this category does not focus on reliability measurement. Instead, it aims to design a reliable system. The fault-tolerance capability of the techniques that come under this category can be proven by the reliability-evaulation techniques as well (though this validation does not come under the scope of this thesis). The major fault-tolerance schemes that lie in this category are redundancy and Markov Random Field.

2.3.2.1 Redundancy

Redundancy is the basic approach to design a fault-tolerant circuit [21, 22]. It works by replicating each gate in the circuit (or that portion of the circuit probable of being in error) and then taking the output from the majority output decision of the original and copied gates. Hence, if a single gate (or a circuit module) in the redundant combination is faulty, the output is not affected. Redundancy can be split into static redundancy (fault-masking) and dynamic redundancy (dynamic recovery).

In static redundancy [20], same function is computed by identical units and their outputs are voted to remove the error generated by the faulty module. The simple form of static redundancy is triple modular redundancy (TMR) which triplicates the original module and the output is decided by the ‘majority decision module’. This concept is illustrated in Fig. 2.5, whereby the original circuit has been placed with two identical copies. The decision module samples the results of three modules and outputs the result which has the highest instances received from the three modules. Hence, the TMR system is designed to counter single module error only.

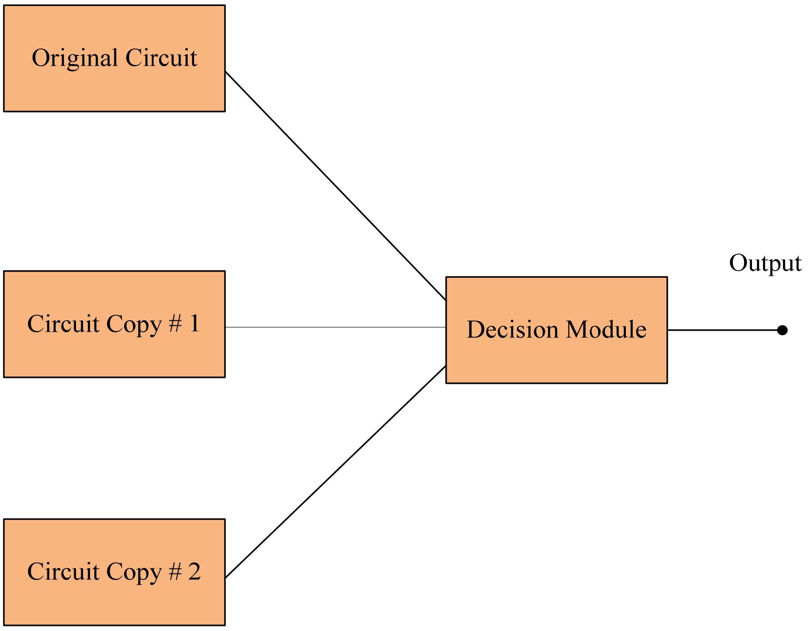
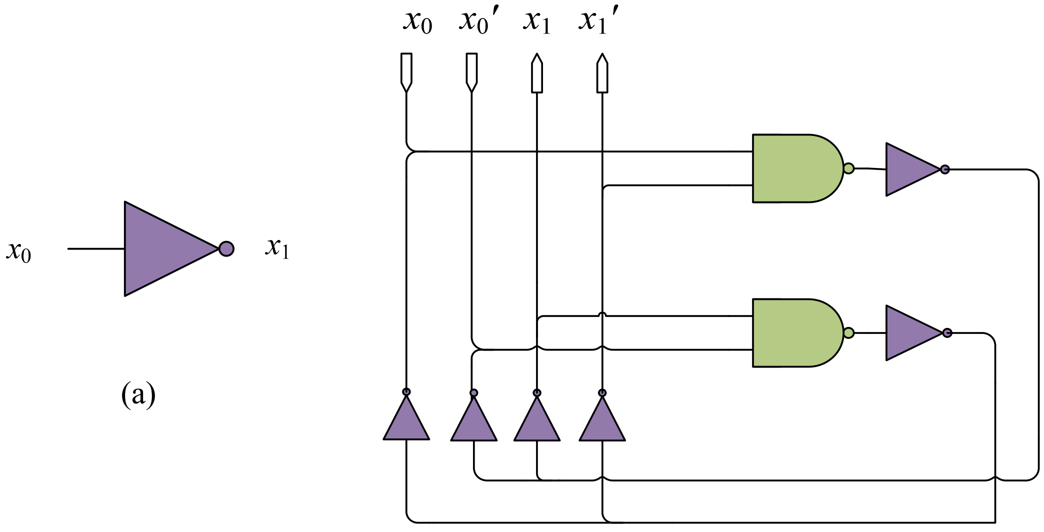


Fig 2.5 The mechanism of triple modular redundancy

Dynamic redundancy is based on incorporating more than one module for the similar logic function but the second or further copies are powered only when the running module provides a faulty output. There are automatic mechanisms that detect the fault and switch in the spare module. Followed by the functional module installation, software actions (rollback, initialization, retry, and restart) are performed which are necessary to restore and continue the computation [20]. Although this mechanism is more hardware-efficient than voted systems, its disadvantage is in terms of delay occurred during the resumption of computation.

2.3.2.2 Markov Random Field (MRF) Model

Markov Random Field (MRF) model has been introduced in [23, 47] to perform reliable circuit operation under the effect of thermal noise. To deal with this noise, MRF equivalents [23, 47] of universal gates have been proposed that can very well isolate the effect of thermal noise in the circuit and prevent it from affecting the final output. The final output comes out to be clean as if there were no noise in the circuit. Fig. 2.6 illustrates the difference between CMOS and MRF-CMOS inverter, as an example. The detailed modelling and MRF design procedure will be provided in Chapter 3.



(a)

(b)

Fig 2.6 (a) CMOS Inverter (b) MRF-CMOS Inverter

According to this model, a Gaussian noise source is added at the output of each logic gate/stage. This noise source is accounted for the thermal noise generated from all the components in this stage. In this way, the thermal noise effect of the current stage is supposed to be countered by the MRF equivalent of the next circuit stage. In other words, the thermal noise effect in MRF is modelled by placing a noise source at the input of the gate/stage which represents the noise originating from the previous stage. A more analytical model of thermal noise is rather tedious to develop; this thermal noise model is considered to be adequate for simulation purposes. Simulations [23, 47, 48] show that injecting thermal noise in a circuit causes many unnecessary bit reversals in a simple CMOS gate as compared to almost noiseless output of MRF-CMOS gate. The drawback of this technique is the noticeable increase in the number of transistors required for a simple circuit. But, for improved circuit reliability, high transistor count is the price a circuit designer has to pay.

2.3.2.3 Comparing Redundancy and MRF

In this section, a comparison is provided between the architecture-level techniques.

The major difference between the two techniques is that the redundancy is based upon making copies of the same gate/module whereas the MRF believes in changing the gate/circuit design. Hence, in the redundant combination, the noise which is the cause of transient error in a single gate can affect its copies as well. In contrast, MRF restructures the gate design so that it absorbs the noise within the circuit. The noise tolerance simulations comparing redundancy and MRF shows that the MRF gates experience very little distortion in the outputs whereas the redundant gates pose a lot of bit reversals and errors for the injection of similar amount of noise [48]. Hence, it can be safely said that the MRF is a more reliable technique between the two architecture-level options.

The majority decision module (called as voter) is assumed to work perfectly in order to ensure correct functioning of the redundant architectures. Whereas, it is evident that if the voter halts to operate, the redundant architecture collapses. By using Monte Carlo simulations, it has been proved that imperfect voting circuitry greatly reduces the TMR system reliability [49]. In order to achieve an efficient TMR system in the presence of noise, higher orders of redundancy are required which demands extra hardware [50]. Moreover, the crosstalk noise degrades the voter's mechanism as well [51]. As compared to redundancy, MRF does not involve any decision module, thereby avoiding the system to rely on any single circuit module.

Redundancy is designed to counter no more than fixed number of errors in a redundant combination. TMR is designed to counter single error per module. Higher orders of redundancy like CTMR (cascaded triple modular redundancy) can increase the error-handling capability though increasing the circuit complexity exponentially. MRF, on the other hand, is not designed to handle a particular number of errors. Instead, it is designed so that the error doesn't occur as it absorbs the noise within the modified gate architecture.

The number of transistors required for an efficient redundant system exceeds the requirement for an MRF system e.g. 2nd order CTMR inverter requires 54 transistors as compared to 34 transistors for the MRF alternative. The higher orders of redundancy require even more transistors. But the first two orders of redundancy i.e. 0th and 1st order require fewer transistors than MRF though they lack in providing the similar reliability as the MRF inverter offers. The area consumption of voters is an extra overhead in a redundant system.

Hence, on the basis of above-mentioned comparisons, MRF can be proved to be better in reliability, error-handling capability and area efficiency as compared to redundancy. Therefore, MRF is expected to replace redundancy in future as the research on MRF is in the initial phase only [23] whereas redundancy is used as a fault-tolerance scheme since long [20-22, 52].

2.3.3 CAD Tools Development

This category is analogous to the reliability-evaluation schemes except that the computation is performed using software tools. Hence, the reliability of rather complex circuits can be calculated with considerable ease with the use of computer-aided design tools. Note that the schemes covered in the first category can be merged into this category provided they will be automated in future. Only two tools have been developed that can calculate and plot reliability graphs which include MEMSTAR (Multiple environment and multiple error simulation tool for analysis of reliability) and AutoPTMate.

2.3.3.1 Multiple Environment and Multiple Error Simulation Tool for Analysis of Reliability (MEMESTAR)

MEMSTAR [45] is a simulation framework that determines the reliability of the circuit under different area and operating environments. It has the capability to take into consideration effects of multiple faults. A special case-study was carried out in this work (for LGSynth'93 benchmark circuits) that plots the circuit failure rate as the gate failure rate increases (from 0 to 10%) and also with the number of fault injections introduced per trial.

The MEMSTAR was implemented using VPI (Verilog Procedural Interface) extensions to Verilog HDL (Hardware Description Language). The programming code was simulated under the Cadence Logic Design and Verification Package v5.1.

The uniqueness of this software tool is the measurement of reliability-evaluation under multiple circuit parameter environments. These parameters include transistor length or width, threshold voltage and power supply voltage as few examples. Hence, it is an efficient tool developed to estimate the performance of future CMOS technologies.

2.3.3.2 AutoPTMate

AutoPTMate [46] is the tool that automates the process of PTM (Probabilistic transfer matrices) computation. It takes the circuit description in a form of netlist, breaks the circuit intro stages and calculates the PTM for each stage. The implementation of Eq. (2.1) provides the reliability of the circuit output. With the automatic computation, the reliability plots i.e. circuit error-probability against gate error-probability can also be drawn with considerable ease. With these plots, a circuit designer can locate and design robustly, those gates/modules of a circuit which are more susceptible to errors. The tool has been programmed in PERL scripting language which generates the MATLAB m-file as an output. Upon running the m-file in MATLAB, the reliability of the test circuit can be directly obtained.

2.4 Markov Random Field

The reliability-evaluation and the CAD tools development, as shown in the previous sections are used to calculate the reliability of the circuits. In contrast, architecture-level solutions are based upon designing a fault-tolerant system and not just measuring the fault-tolerance (in the form of circuit reliability) of test circuits. From a circuit designer’s point of view, the category that goes beyond mathematical models i.e. architecture-level techniques is selected for further research. Moreover, due to the high research literature volume for each category, the scope of this research has to be kept limited up to one category i.e. architecture-level solutions.

Within the architecture-level solutions, there are two techniques that promise a fault-tolerant circuit design. It has been proved in Sec. 2.3.1.3 that the Markov Random Field (MRF) is superior to redundancy for higher reliability, error-handling capability and area efficiency [21-23, 49-51]. Hence, the scope of this research is reserved up to Markov Random Field only.

After conducting a literature review on MRF, it is found that this technique has a room for improvement at the design, simulation framework and implementation levels [23, 47]. At the design stage, the fault-tolerance rules derived from the mathematical model of MRF had not been validated on a practical circuit example. Therefore, the premier target of this research is to perform a case study in which the mathematical model of MRF will be mapped on a test circuit and the fault-tolerance rules will be derived accordingly.

At the simulation level, the noise framework is kept limited to thermal noise only whereby it has been found in literature that Random Telegraph Signal (RTS) noise also affects the noise-immunity of nanoscale circuits [24, 39, 40]. Therefore, the noise framework will be extended from thermal noise to a combination of thermal and RTS noises. Another limitation with the previous MRF research i.e. Nepal et all [23] is that the simulations in this research are based upon 70 nm technology model whereas more downscaled CMOS technology models like 32 nm have been developed (by the Nanoscale Integration and Modeling (NIMO) group of Arizona State University) and are available as open source SPICE programs [25].

At the implementation level, there is no relation developed between the marginal probability power dissipation principle and its implementation on the digital design in Nepal et all [23]. By giving this relation a form of fault-tolerance principle, the architecture of previous MRF design can be modified in order to propose a better noise-immune circuit design. Therefore, in this research, the MRF circuit design will be modified to provide an improved MRF model.

The methodology of this research work is to improve to previous MRF based circuit design technique (in [23]) on the design, simulation and implementation stages. The modifications proposed in this section are implemented and discussed in detail in chapter 3. The worth of the previous and modified MRF designs will be compared with the help of simulations conducted in Spectre Circuit Simulator (in Cadence) in chapter 4.

CHAPTER 3

MARKOV RANDOM FIELD BASED CIRCUIT DEISGN

Markov Random Field (MRF) is a branch of probability theory, the existence of which dates back to the early 70's [53, 54]. This technique has been used to solve problems in the fields of computer vision, artificial intelligence or image processing, as few examples. In these research areas, MRF serves as a suitable modelling technique due to its convenience to model spatially correlated features e.g. describing dependence among image pixels in computer vision problems [55]. For digital circuits and systems, this technique was used to develop fault-tolerant circuit architecture [23, 47]. The work presented in this thesis is an extension of the research conducted on the digital circuit design application of MRF. This section is initiated by a brief overview of MRF theory followed by an explanation of its mathematical model and implementation on digital circuits.

3.1 MRF Graph Theory

Consider the graph in Fig. 3.1. Each point or site is a random variable x_i, from the set of random variables $X = \{x_1, x_2, x_3, x_4, x_5\}$. These variables are connected to each other via edges. All the variables connected to a specific site (via edges), x_i, make up the *neighbourhood* of this site.

In order for any random variable set to form an MRF system, it has to satisfy the following two rules [21].

1) Positivity

$$p(x) > 0, \quad \forall x \in X \tag{3.1}$$

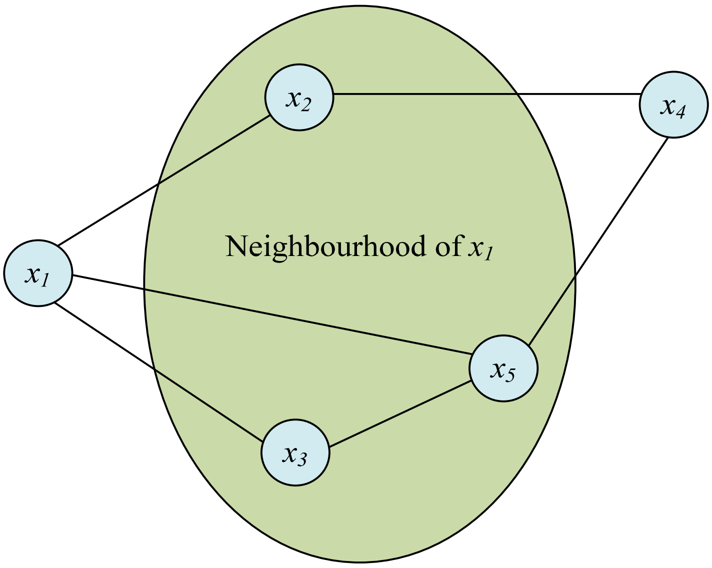


Fig 3.1: MRF Neighbourhood System

This rule implies that the probability of each variable, x_i should be greater than zero which is actually the mathematical proof of the variable's existence (in a graphical representation of a mathematical problem).

2) Markovianity

$$p\left(x_i \mid \{X - x_i\}\right) = p(x_i \mid N_i) \tag{3.2}$$

This rule states that any site or variable is influenced only by its neighbours with whom it is connected by edges (since edges show the statistical dependencies among nodes). This concept has been shown in Fig. 3.1 with reference to the node x_1. According to this principle, conditional probability of x1 is conditioned upon the variables in its immediate neighbourhood only i.e. x_2, x_3 and x_5.

A group of two or more variables forms a clique if each variable in it is connected to all other variables by edges. In Fig. 3.1, the sets $\left\{x_1, x_2\right\}$ and $\left\{x_1, x_3, x_5\right\}$ form first and second order cliques respectively. The values obtained by these variables are called 'labels'. For digital systems, each variable can have only two labels, 0 or 1 (unless the mathematical model of the system is probabilistic where intermediate node states i.e. between 0 and 1 also have probability of occurrence greater than or equal to 0).

For mapping digital circuits onto MRF system, the circuit nodes are considered as sites and show their dependence on each other by using edges. An example of translating digital circuit onto its MRF graph is shown in the following section.

3.2 MRF Mathematical Model

The mathematical model of MRF consists of two statistical terms, joint and marginal probability. For each term, a brief introduction followed by its detailed computation procedure is provided. By the end of this section, fault-tolerance design rules will be derived from the conclusions obtained by these analyses. Kindly note that the mathematical analyses performed in this section are the case studies (for a test circuit) conducted for the validation of fault-tolerance rules proposed in [23, 47].

3.2.1 Joint Probability

The joint probability of MRF network, according to Hammersley-Clifford theorem [56, 57] can be written as,

$$P(X) = \frac{1}{Z}\prod_{c \in C} e^{\frac{-U_c}{kT}} \tag{3.3}$$

where X is the set of all nodes in the MRF network, C is the set of cliques and U_c is the clique energy function. The term Z is called 'normalization constant' which is required to normalize the probability function to $\{0, 1\}$. The term kT is the thermal energy which controls the sharpness of the joint and marginal probability distribution graphs. Since the joint probability is the function of all node states in a network, it is responsible for the correct or incorrect operation of the network. It is the probability calculated for a specific node label combination and since (in real-time operation) each node has two possible states (0 or 1), there are a total of 2^n possible values of joint probability where 'n' is the number of nodes in a network.

The system represented by MRF (as a dependence graph) can be decomposed into cliques. Since these cliques are independent of each other, the joint probability of each can be calculated separately. At the end, individual results will be multiplied to calculate the joint probability of the whole system. Nepal et al [23] found that the *correct logic states* are those that maximize the joint probability of the overall network. Note that the correct logic states refer to those logic states which are achieved in a circuit state without error. In the following section, the validation of this finding is provided along with the procedure to compute and maximize the joint probability.

3.2.1.1 Detailed Computation Procedure

For the purpose of joint probability computation, a test circuit (M3 module of C432 interrupt controller) from [58] is used. Fig. 3.2 shows its logic diagram and dependence graph. To create a dependence graph, the inputs, intermediate and output wires will be labeled and represented as nodes with their respective dependence on each other according to MRF theory.

The Eq. (3.3) is used for joint probability computation. The equation requires us to identify all the cliques in the network and compute their energy functions, U_c, before proceeding to the $\prod$ function evaluation. Hence, the observed cliques are $\{x_3, x_4\}$, $\{x_2, x_4, x_5\}$, $\{x_0, x_1, x_5, x_6\}$ and $\{x_6, x_7\}$.

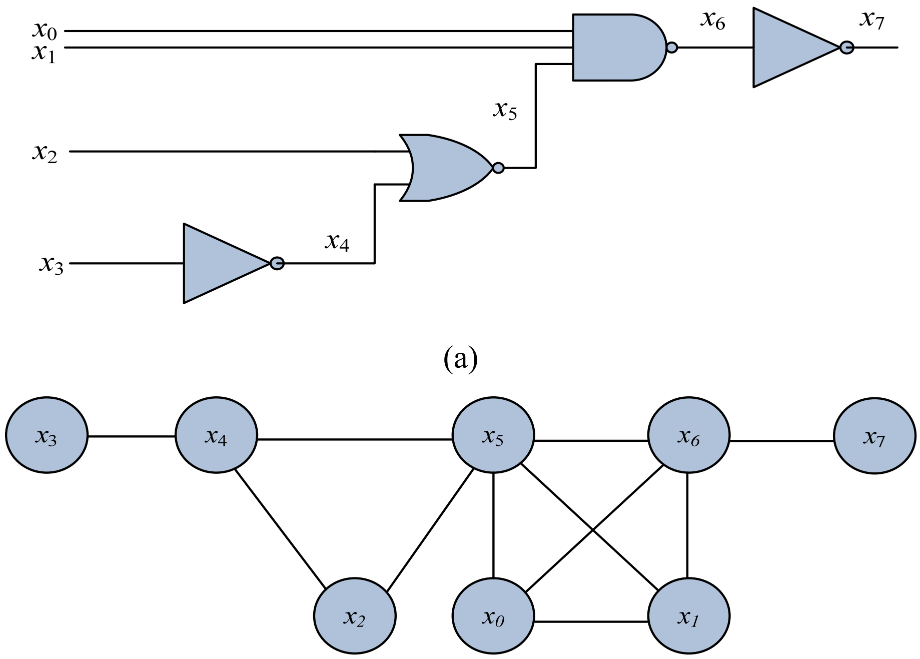
The formula for evaluating clique energy function is derived from [57] and its general form can be expressed in Eq. (3.4).

$$U_c(x_0, x_1, \ldots\ldots, x_i) = -\sum_i f_i(x_0, x_1, \ldots\ldots, x_i) \tag{3.4}$$

Before proceeding towards the computation, the significance of the following points has to be understood.

- The function, *f* in Eq. (3.4) contains the valid minterms in the logic compatibility function i.e. for which *f*=1. The logic compatibility function of a logic gate is the same as its truth table except for the additional function *f*, which shows that whether the output of the logic gate is correct (*f*=1) or erroneous. The logic compatibility function of NAND gate, with reference to Fig. 3.2 is shown in Table 3.1.

- The negative sum in Eq. (3.4) accounts for the principle that the total logic energy of valid minterms should be less than invalid ones for the correct functioning of the logic element. This principle was proved in [22, 47] where the probabilities are physically related to the entropy of computation.



(a)

(b)

Fig 3.2: (a) Test circuit (b) Its dependence graph

Table 3.1: Logic Compatibility Function of NAND

x_0	x_1	x_5	x_6	f
0	0	0	1	1
0	0	0	0	0
0	0	1	1	1
0	0	1	0	0
0	1	0	1	1
0	1	0	0	0
0	1	1	1	1
0	1	1	0	0
1	0	0	1	1
1	0	0	0	0
1	0	1	1	1
1	0	1	0	0
1	1	0	1	1
1	1	0	0	0
1	1	1	0	1
1	1	1	1	0

- For the sake of relating logic energy to thermal energy, the logic variables are treated as algebraic variables and logic operations are converted to algebraic operations [22]. The main Boolean to algebraic conversion used in the upcoming computation is expressed as:

$$X' \rightarrow (1-X) \tag{3.5}$$

Now, the steps for evaluating U_c of NAND gate will be outlined, as an example.

$U_c = -$(Sum of valid minterms ($f = 1$) in the logic compatibility function (Table 3.1))

$$= -\begin{bmatrix} x_0'x_1'x_5'x_6 + x_0'x_1'x_5x_6 + x_0'x_1x_5'x_6 + x_0'x_1x_5x_6 + x_0x_1'x_5'x_6 + x_0x_1'x_5x_6 \\ + x_0x_1x_5'x_6 + x_0x_1x_5x_6' \end{bmatrix}$$

Now, applying boolean simplification,

$$= -\begin{bmatrix} x_0'x_1'x_6(x_5 + x_5') + x_0'x_1x_6(x_5 + x_5') + x_0x_1'x_6(x_5 + x_5') + x_0x_1x_5'x_6 \\ + x_0x_1 \; x_5x_6' \end{bmatrix}$$

$$= -\left[x_0'x_1'x_6 + x_0'x_1x_6 + x_0x_1'x_6 + x_0x_1x_5'x_6 + x_0x_1x_5x_6'\right]$$

$$= -\left[x_0'x_6(x_1' + x_1) + x_0x_1'x_6 + x_0x_1(x_5'x_6 + x_5x_6')\right]$$

Applying Boolean to Algebraic conversion,

$$= -\left[(1-x_0)x_6 + x_0x_6(1-x_1) + x_0x_1(x_6(1-x_5) + x_5(1-x_6))\right]$$

$$= -\left[x_6 + x_0x_1x_5 - 2x_0x_1x_5x_6\right]$$

$$= 2x_0x_1x_5x_6 - x_0x_1x_5 - x_6$$

Similarly, the U_c of NOT (index 1 & 2 having x_3 and x_6 as inputs respectively) and NOR gates were calculated (with reference to Fig. 3.2(a)) and listed in Table 3.2. It can be observed that, for each clique energy function, one of its terms contains all the variables associated with that gate, another term includes output and the rest of the terms are subsets of the inputs of this gate.

Table 3.2: Clique Energy Functions for NOT and NOR gates

NOT 1	$U_c = 2x_3x_4 - x_3 - x_4$
NOR	$U_c = x_2x_4 + 2x_4x_5 + 2x_2x_5 - 2x_2x_4x_5 - x_2 - x_4 - x_5$
NOT 2	$U_c = 2x_6x_7 - x_6 - x_7$

Using the clique energy functions, their respective exponentials are evaluated and the exponential results are multiplied at the end to evaluate the overall joint probability of the test circuit. This methodology is manifested in the following computation of joint probability (with reference to Fig. 3.2(a) and Eq. (3.3)).

$$\begin{aligned} P\ (x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7) &= \frac{1}{Z}\left(e^{-U_c(NOT\ 1)/\,kT}\,.\,e^{-U_c(NAND)/\,kT}\,.\;e^{-U_c(NOR)/\,kT}\,.\,e^{-U_c(NOT\ 2)/\,kT}\right) \\ &= \frac{1}{Z}\exp\begin{bmatrix}(x_2+x_3+2x_4+x_5+2x_6+x_7-x_2x_4-2x_2x_5-2x_3x_4 \\ -2x_4x_5-2x_6x_7+x_0x_1x_5+2x_2x_4x_5-2x_0x_1x_5x_6)\,/\,kT\end{bmatrix} \end{aligned}$$

Following the joint probability calculation, the node label combinations will be determined that maximize its value (following the joint probability rule proposed in [23]). The simplified form of $P(x_0, x_1, ..., x_7)$ shows that the power of its exponential has to be maximum to obtain the maximum value of this function i.e. for the maximum value of numerator of the power $(x_2 + x_3 + 2x_4 + x_5 + 2x_6 + x_7 - x_2x_4 - 2x_2x_5 - 2x_3x_4 - 2x_4x_5 - 2x_6x_7 + x_0x_1x_5 + 2x_2x_4x_5 - 2x_0x_1x_5x_6)$.

The MATLAB is used to determine the value of this power's numerator for its 256 (=2^8) possible node combinations. It can be observed that the maximum value of the numerator is '4' and it exists for sixteen combinations of node labels shown in Table 3.3. These combinations are the same as the sixteen combinations of this circuit's truth table; which shows that the joint probability is maximum for correct logic combinations (since the truth table lists only correct label combinations i.e. combinations with no errors). For rest of the combinations, its value is always lower.

Although only the two logic states (0 & 1) have been considered for all nodes, the logic states between 0 and 1 also have probability of occurrence greater than 0 (since MRF is a probabilistic framework). The reason for considering these two values only is that the real-operation logic values are used but if intermediate values are considered, the joint probability would still be less than maximum (for the infinite intermediate-label combinations. To justify this principle, a 3-dimentional joint probability graph of inverter has been constructed as shown in Fig. 3.3. In this figure, the joint probability has attained maximum value for only correct logic combinations of {x_0, x_1) i.e. {0, 1} and {1, 0}, as expected.

Table 3.3: Node combinations having maximum joint probability

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0	1	1	0	0	1	0
0	1	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	1	1	1	0	0	1	0
0	0	0	0	1	0	1	0
0	1	0	0	1	0	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0
0	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0
1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0
0	0	0	1	0	1	1	0
0	1	0	1	0	1	1	0
1	0	0	1	0	1	1	0
1	1	0	1	0	1	0	1

3.2.1.2 Design Principle of Joint Probability

From the joint probability analysis, it can be concluded that for the perfect logic operation of a circuit i.e. with no errors at any nodes of the circuit, the circuit should be designed, as such to ensure at all times, that the joint probability of the circuit remains maximum.

Therefore, with the proposed computation procedure, a principle of a fault-tolerant circuit design is derived which is in agreement with the joint probability requirement stated in Nepal et al [23]. Following the validation of this principle, the next target is to devise the circuit design requirements that enforce the maximum joint probability of the circuit which will be addressed in Sec. 3.3.

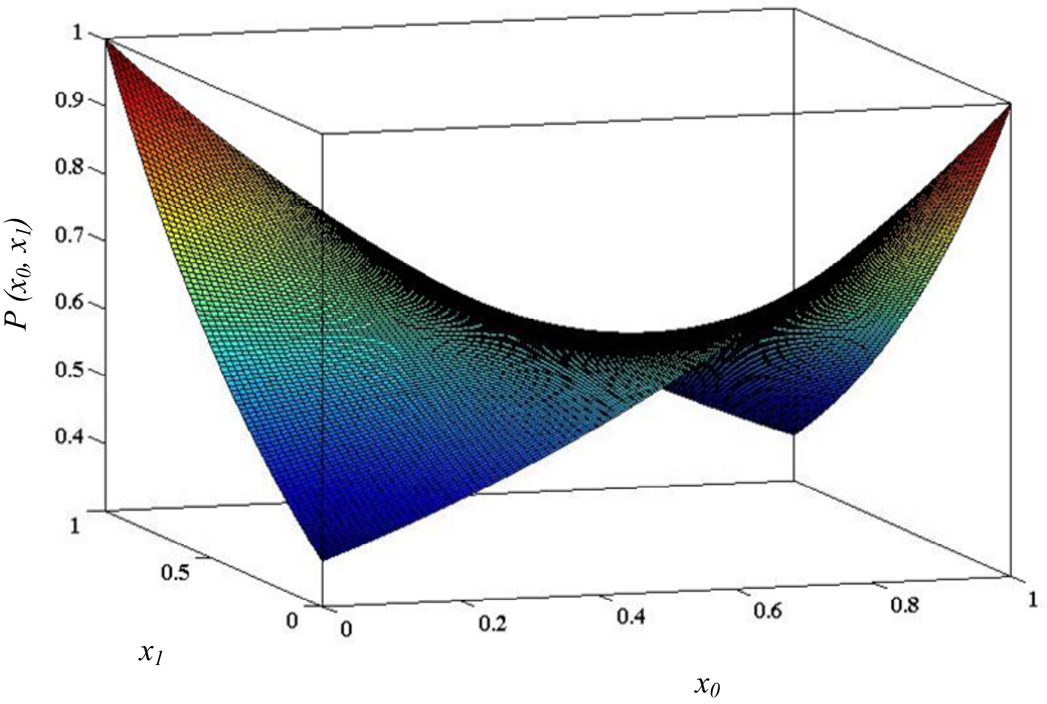


Fig 3.3 Joint Probability graph of Inverter

3.2.1.3 Foundation of Joint Probability Principle

The question on the origination of maximum joint probability principle still needs to be answered i.e. from where this principle actually came from. The answer to this question lies in the J. Pearl's literature [57]. For the sake of simplicity, the origination of this principle is explained by using a simple circuit example in Fig. 3.4(a).

The circuit is an inverter cascade. The nodes in the figure are labeled *A*, *B* and *C* whereas its dependence graph in shown in part (b). Note that *B*=1 implies that *A*=*C*=0. Hence, in order to ensure that *B*=1*,* the condition of *A*=*C*=0 has to be ensured as well.

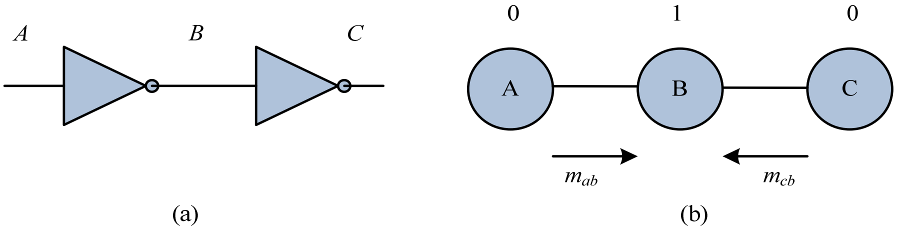


Fig 3.4 (a) Inverter cascade (b) Its dependence graph

The above concept has been used in designing message-passing algorithms in the fields of artificial intelligence and computer networking algorithms. Since every node in the system has some probability measure associated with it; its value is maximum only when the neighbouring nodes' values are in compliance with it e.g. the node *B* has the maximum probability of being in logic 1 if its neighbouring nodes stay at logic 0. This probability has been calculated at node *B* by using messages coming from the neighbouring nodes i.e. m_{ab} and m_{cb}. These messages actually inform the node-updating algorithms about their present states. Since, every node in the network is receiving as well as sending messages to its neighbouring nodes, there is a probability measure computed on each node. This probability measure, if stay maximum on each node indicates that the network is working fine and without error.

Therefore, the above concept has been mapped on circuit networks where the concept of a clique is used in place of a single node. It can be observed from the joint probability Eq. (3.3) that the clique energy Eq. (3.4) has to remain minimum to obtain the maximum value of the clique's joint probability. And for all correct logic combinations, the clique energy always stays minimum. As a result, the overall circuit's joint probability (which is the multiple of clique probabilities) remains maximum as well.

3.2.2 Marginal Probability

The calculation of marginal probability requires fixing values of one or more variables in the function and summing it over non-fixed variables. For discrete two-random variable case, the marginal probability function is written as $p(X=x)$ [59] i.e.

$$p(X=x)=\sum_{y} p\left(X=x,Y=y\right)=\sum_{y} p(X=x\,|\,Y=y)*p(Y=y) \tag{3.6}$$

where $p(X=x, Y=y)$ is the joint distribution of X & Y, while $p(X=x|Y=y)$ is the conditional distribution of X given Y.

Marginal probability is a function of a single node state unlike joint probability. For a multi-variable case, the dependence of each node is gradually removed from the joint probability (of a particular clique) unless the output node of that clique is reached. The resulting single variable function links the probability of occurrence of all node-states to their logic values (between 0 and 1 inclusive).

Since the inputs of a logic circuit have defined probabilities of being in logic state 0 or 1; the intermediate and output nodes (together termed as hidden nodes) have the probabilities which have to be calculated. For this purpose, Pearl's belief propagation algorithm [47, 60] is used. This algorithm computes the marginal probabilities of intermediate and output nodes by marginalizing each node step by step unless the desired node is reached. The use of this statistical term is three-fold. It helps us to

- determine the most probable logic state for any node in the network.
- observe the variation of any node state's probability with temperature variation.
- understand the principle of probabilistic computation with reference to MRF.

The significance of these properties will be well understood after going through the following computation procedure of marginal probability.

3.2.2.1 Detailed Computation Procedure

This analysis is conducted on the same test circuit (Fig. 3.2) that is used for the joint probability case. As there are eight nodes, eight marginal probability functions should be evaluated. Since the probability distribution of inputs has been provided by the user; for simplicity, it can be assumed that all of the inputs are equally likely to be in logic states 0 or 1. For computing probabilities of remaining four nodes (including three intermediate and one output) hidden nodes, the belief propagation algorithm is used [47].

The marginal probability computation proceeds in the following steps.

- The first step is to assign probability distribution functions (PDF) to all inputs and cliques as shown in Table 3.4. In the process of computing marginal probability of output, x_7, the probabilities of all the intermediate nodes (x_4, x_5 and x_6) will also be calculated, as the belief propagation algorithm does not allow evaluating the probability of a given node without the knowledge of probability functions of its dependent nodes.

Table 3.4: Probability Distribution Functions for inputs and cliques

Input	PDF	Clique	PDF
x_0	$f_0\,(s_0)$	$\{x_3,x_4\}$	$f_4\,(x_3,x_4)$
x_1	$f_1\,(s_1)$	$\{x_2,x_4,x_5\}$	$f_5\,(x_2,x_4,x_5)$
x_2	$f_2\,(s_2)$	$\{x_0,x_1,x_5,x_6\}$	$f_6\,(x_0,x_1,x_5,x_6)$
x_3	$f_3\,(s_3)$	$\{x_6,x_7\}$	$f_7\,(x_6,x_7)$

- Initially, $p(x_7) = f_0 f_1 f_2 f_3 f_4 f_5 f_6 f_7$.
- The inputs will be eliminated first followed by intermediate nodes unless the output node is reached. In eliminating one node, two of the functions of that node eliminate and one new function forms. So, for each step, one function from $p(x_7)$ decreases unless only one function is left which would be dependent only on x_7.

The marginal probability of the output, $p(x_7)$, will be computed by the following seven steps.

Step1: Eliminate x_3

Eliminated : $f_3(s_3), f_4(x_3,\ x_4)$

New : $f_8(x_4)$

$$p(x_4) \ = \sum_{x_3 \in (0,1)} \frac{1}{Z_1} e^{-U_c\ (NOT\ \ 1)/\ kT} = \frac{1}{Z_1}\left(e^{x_4/kT} + e^{(1-x_4)\ /kT}\right) = f_8(x_4)$$

$$\Rightarrow p(x_7) \ = f_0 f_1 f_2 f_5 f_6 f_7 f_8$$

Step2: Eliminate x_2

Eliminated : $f_2(s_2), f_5(x_2, x_4, x_5)$

New : $f_9(x_4, x_5)$

$$p(x_5 \mid x_4) = \sum_{x_2 \in (0,1)} \frac{1}{Z_2} e^{-U_c\ (NOR)\ /\ kT} = \frac{1}{Z_2}\left(e^{(x_4+x_5-2x_4x_5)\ /kT} + e^{(1-x_5)\ /kT}\right) = f_9(x_4, x_5)$$

$$\Rightarrow p(x_7) \ = f_0 f_1 f_6 f_7 f_8 f_9$$

Step3: Eliminate x_4

Eliminated : $f_8\left(x_4\right),\, f_9\left(x_4, x_5\right)$

New : $f_{10}\left(x_5\right)$

$$p\left(x_5\right)=\sum_{x_4 \in\ (0,1)} \frac{1}{Z_3}\left[p\left(x_5 \mid x_4\right) *\, p\left(x_4\right)\right]=\frac{1}{Z_3}\left(e^{x_5\ /kT}+3 e^{\left(1-x_5\right)\ /kT}+\ e^{\left(1+x_5\right)\ /kT}+\ 3 e^{\left(2-x_5\right)\ /kT}\right)$$

$$= f_{10}\left(x_5\right)$$

$$\Rightarrow p\left(x_7\right)\ = f_0 f_1 f_6 f_7 f_{10}$$

Step4: Eliminate x_0

Eliminated : $f_0\left(s_0\right), f_6\left(x_0, x_1, x_5, x_6\right)$

New : $f_{11}\left(x_1, x_5, x_6\right)$

$$p\left(x_6 \mid x_1, x_5\right)\ =\sum_{x_0 \in\ (0,1)} \frac{1}{Z_4}\, e^{-U_c\ (NAND)\ /kT}=\frac{1}{Z_4}\ \left(e^{\left(x_6\right)\ /kT}\ +\ e^{\left(x_6+x_1 x_5-2 x_1 x_5 x_6\right)\ /kT}\right)= f_{11}\left(x_1, x_5, x_6\right)$$

$$\Rightarrow p\left(x_7\right)\ = f_1 f_7 f_{10} f_{11}$$

Step5: Eliminate x_1

Eliminated : $f_1\left(s_1\right), f_{11}\left(x_1, x_5, x_6\right)$

New : $f_{12}\left(x_5, x_6\right)$

$$p\left(x_6 \mid x_5\right)=\sum_{x_1 \in\ (0,1)} \frac{1}{Z_5} f_{11}\left(x_1, x_5, x_6\right)=\frac{1}{Z_5}\left(e^{\left(x_5+x_6-2 x_5 x_6\right)\ /kT}\quad+\ 3 e^{\left(x_6\right)\ /kT}\right)= f_{12}\left(x_5, x_6\right)$$

$$\Rightarrow p\left(x_7\right)\ = f_7 f_{10} f_{12}$$

Step6: Eliminate x_5

Eliminated : $f_{10}\left(x_5\right), f_{12}\left(x_5, x_6\right)$

New : $f_{13}\left(x_6\right)$

$$p\left(x_6\right)=\sum_{x_5 \in\ (0,1)} \frac{1}{Z_6}\left[p\left(x_6 \mid x_5\right) * p\left(x_5\right)\right]$$

$$=\frac{1}{Z_6}\ \left(28 e^{\left(1+x_6\right)\ /kT}+\ 13 e^{\left(x_6\right)\ /kT}\ +15 e^{\left(2+x_6\right)\ /kT}+\ 3 e^{\left(1-x_6\right)\ /kT}\ +4 e^{\left(2-x_6\right)\ /kT}+\ e^{\left(3-x_6\right)\ /kT}\right)$$

$$= f_{13}\left(x_6\right)$$

$$\Rightarrow\ p\left(x_7\right)\ = f_7 f_{13}$$

Step7: Eliminate x_6

Eliminated : $f_7\left(x_6, x_7\right),\, f_{13}\left(x_6\right)$

New : $f_{14}\left(x_7\right)$

$$p\left(x_7\right)\ =\ \sum_{x_6 \in\ (0,1)} \frac{1}{Z_6}\ \left[\ p\left(x_7 \mid\ x_6\right)\ *\ p\left(x_6\right)\ \right]$$

$$
\begin{aligned}
&= \frac{1}{Z_6}\begin{pmatrix} 31e^{(1+x_7)}\,{/}^{kT} + 19e^{(2+x_7)}\,{/}^{kT} + e^{(3+x_7)}\,{/}^{kT} + 3e^{(1-x_7)}\,{/}^{kT} + 17e^{(2-x_7)}\,{/}^{kT} + 29e^{(3-x_7)}\,{/}^{kT} \\ +15e^{(4-x_7)}\,{/}^{kT} \end{pmatrix} \\
&= f_{14}(x_7) \\
&\Rightarrow p(x_7) = f_{14}
\end{aligned}
$$

Following the calculation of node probability functions (in steps 1, 3, 6 and 7), they are plotted using MATLAB to observe their probability distribution with respect to their logic values. In Fig. 3.5, probability distribution graphs (for kT=0.05) are shown; the discussion of which is provided as follows.

- The $p(x_4)$ graph (Fig. 3.5 (a)) shows that there is an equal probability of getting either logic state 0 or 1. By comparing it with inverter's truth table it can be seen that its output has equal probability of being in logic state 0 or 1. Hence, the marginal probability results are verified with the real operation of the inverter.

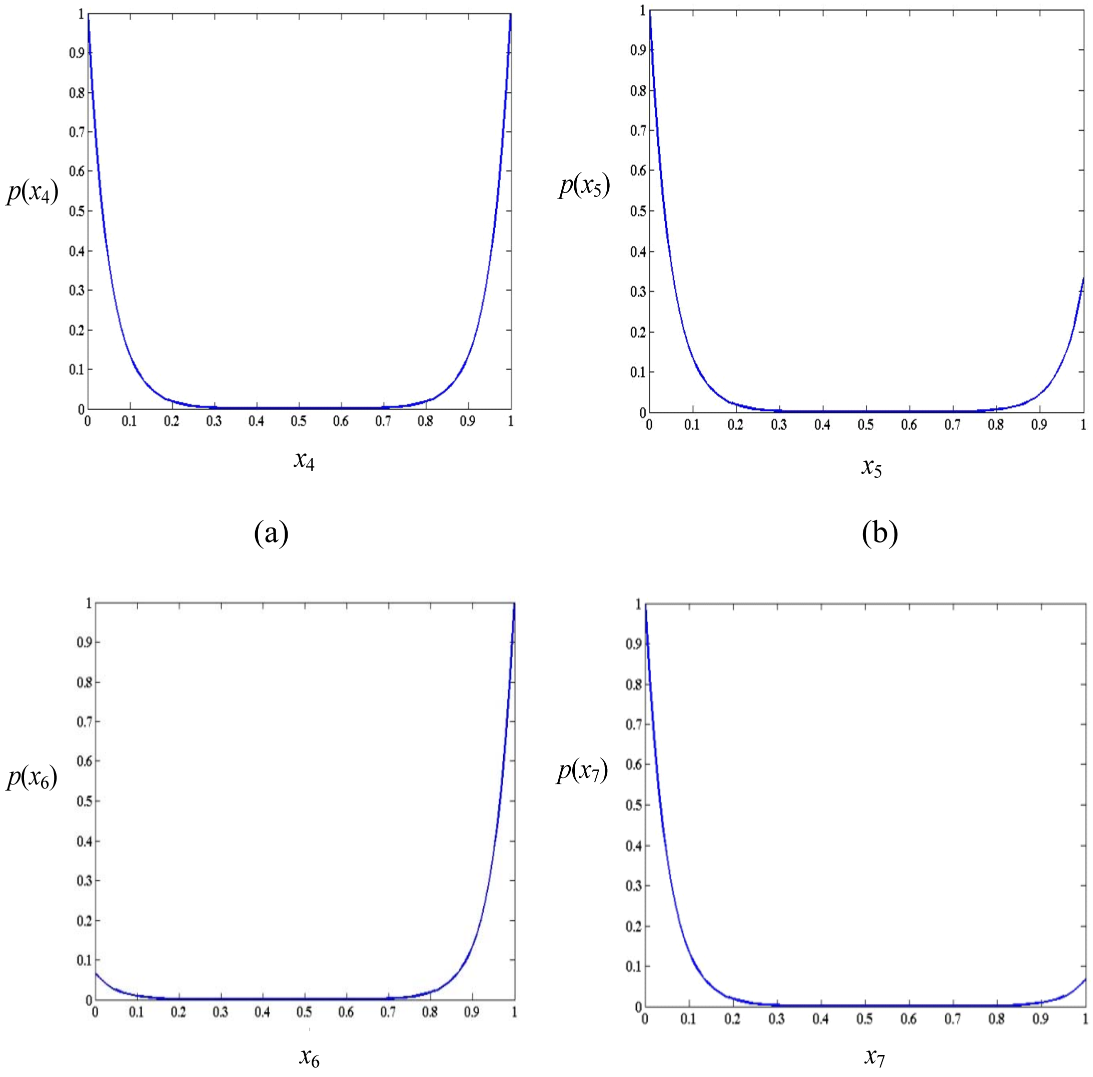
- Similarly, the $p(x_7)$ graph (Fig. 3.5 (d)) shows that the probability of achieving logic 0 at this node is almost sixteen times the probability for logic 1. From the truth table of this circuit, it can also be observed that x_7 goes to logic 1 only once in sixteen node combinations which proves the authenticity of the marginal probability results as they are in agreement with the truth table analysis.

- The probability of intermediate states between 0 and 1 is negligible. Note that since the MRF is a probabilistic framework, the intermediate node state probabilities are greater or equal to zero.

Fig. 3.6 analyzes the probability distribution with respect to temperature variation. The observations are:

- By increasing thermal energy kT, the marginal probability graph moves upward and the probability of intermediate logic states start increasing thus making logic circuit more probable of achieving these states. And since in ideal case, the probability of intermediate states should be zero; the probability of error increases in nano-computation. Since the heat dissipation in the circuit increases the temperature of the system, the marginal probability graph gradually moves upward and if the heat removal system is not efficient, the error probability of the circuit nodes increases continuously.

- Moreover, the noise margin also decreases as a consequence of increasing *kT* rendering digital circuits more prone to error.



(a) (b)

(c) (d)

Fig 3.5: Marginal probability graphs of (a) x_4 (b) x_5 (c) x_6 (d) x_7

The term *kT* in this analysis expresses the amount of energy inherent in the thermal excitations and used here to control the ratio of logic to thermal energy [47].

The values of kT are actually selected in contrast to the unit logic energy e.g. kT=0.1 means that the logic energy is ten times the thermal energy because only normalized logic energy is considered in this analysis.

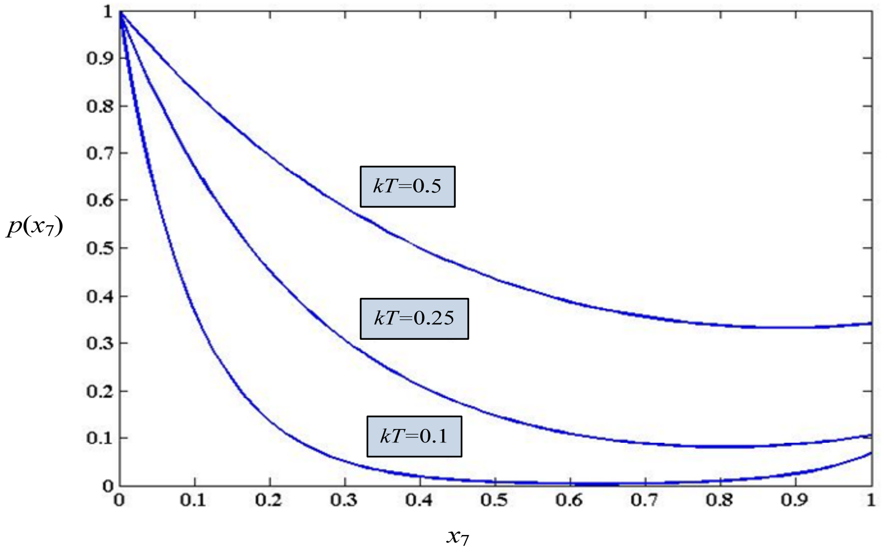


Fig 3.6: Marginal probability variation of x_7

3.2.2.2 Marginal Probability Power Dissipation Principle

The key to design a fault-tolerant circuit is to ensure minimum power dissipation (in the integrated circuit) with a good heat removal system which keeps the probability of intermediate states close to zero and maintain sufficient noise margin as well.

This principle is based upon the dependence of intermediate logic states on the temperature. As the power dissipation increases, the temperature of the system increases. Hence, the probability of occurrence of intermediate logic states also increases which poses more probability of bit reversals. Therefore, the implementation of this principle ensures a sufficient noise margin by limiting the power dissipation.

The marginal probability graphs plotted in this section describe the probabilistic nature of MRF network which was claimed in literature [47]. Moreover, the above analysis has validated temperature-dependence of probability distribution of nodes as reported in [47]. The relation of power dissipation to marginal probability principle is though novel and was not explicitly stated in the previous MRF design mechanism.

3.2.3 Combined Application of Joint and Marginal Probability Requirements

In order to achieve the fault-tolerant operation of nanoscale circuits, the principles of joint and marginal probability have to be followed. The enforcement of joint probability design principle requires the architecture level changes which will be discussed in the following section. In contrast, the marginal probability principle poses a precautionary measure in terms of ensuring minimum power dissipation in the circuit. The marginal probability principle does not require the architecture level changes (as in joint probability requirement) but this principle can be used to find out the limitation in the prior MRF design [47] which will be presented in Sec. 3.3.2.

3.3 MRF Implementation Model

In this section, the procedure to map the design principles of joint and marginal probabilities on digital hardware is shown. This section also describes as how to create simple MRF circuits from CMOS counterparts.

3.3.1 Mapping Design Principle of Joint Probability on Digital Hardware

In the previous section, the rule of error-free computation was described i.e. the enforcement of maximum joint probability of the system. Since, the joint probability of a system is an overall multiple of the joint probabilities of all cliques; it stays maximum as long as the joint probability of each gate (or clique) remains maximum. Note that each gate is associated with its own clique. Accordingly, each gates is designed in a way (in [23]) that its output never goes wrong no matter how much noise disturbance is introduced at its input. So, when the outputs of the all gates stay correct, the logic states of intermediate nodes never go into error. As a result, the joint probability of the overall circuit automatically remains maximum.

To convert the basic CMOS gates into MRF-CMOS gates, following two rules are proposed in [23]*.

*From now on, the terms MRF and MRF-CMOS will be used interchangeably since the MRF is still the revised architecture of CMOS logic.

Rule 1:

> "Each logic state, s_i, should be represented as a bi-stable storage element, taking on logical values of 0 and 1 with equal probability. The probability for any other signal value should be low."

Rule 2:

> "The constraints of each logic graph clique should be enforced by feedback to the appropriate storage elements, implementing the logic compatibility functions to maximize the joint probability of the correct logical values."

The first design rule requires us to provide the original signal as well as its complement for all inputs and outputs. The second design rule requires us to enforce the constraints of the logic graph cliques. These constraints are the minterms contained in the clique energy function (or the valid minterms from the logic compatibility function). These constraints can be represented by an AND gate for each minterm. Following the second design principle, the outputs of AND gates are directed as feedback to the logic states (originals and complements) contained in the minterm.

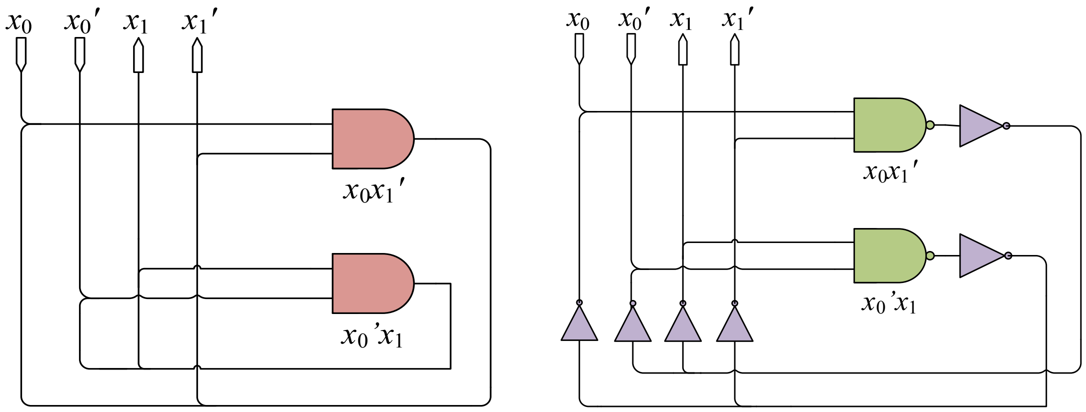
Let us design the NOT gate with these rules. The clique energy function of inverter given as Eq. (3.7) is used as follows.

$$U_c = -\left(x_0' x_1 + x_0 x_1'\right) \tag{3.7}$$

By following the two rules stated above, MRF Inverter is created in Fig. 3.7(a). In the testing phase, lets apply x_0=0 (and hence x_0'=1). The output, x_1 equals to 0 (if previously x_1 was at logic 0) or equals to 1 if x_1 was previously 1. Hence, the output latches into the correct state only when x_1 was previously at logic 1. Therefore, this design suffers from the dependence of output's next state on previous state.

To deal with this problem, NAND gates can be used instead of AND gates with the NAND outputs driven to the complemented form of the states contained in each minterm e.g. for minterm x_0x_1', NAND output is derived as a feedback to x_0' and x_1. Moreover, to provide the effect of a buffer, two inverters are added in each of the feedback paths [49]. The revised circuit is shown in Fig. 3.7(b). The new circuit works independently of previous node states.

The MRF circuits can also be optimized by the clique energy function simplification [47] (which is the simple Boolean logic simplification). Without this simplification, the circuit requires more transistors e.g. 30 transistors are required for a simplified version as compared to *36* transistors and few more interconnects for the original minterm implementation. The noise immunity of the circuit remains the same for both cases. For NAND and NOR gates, the simplified equations would be $\left[U_c = -\left(\left(x_0' + x_1'\right)x_2 + x_0 x_1 x_2'\right)\right]$ and $\left[U_c = -\left(\left(x_0 + x_1\right)x_2' + x_0' x_1' x_2\right)\right]$ respectively. Following the same design principles, the NAND MRF equivalent is constructed in Fig. 3.8 and NOR MRF in Fig 3.9.



(a) (b)

Fig 3.7: Inverter with (a) AND gates (b) NAND gates

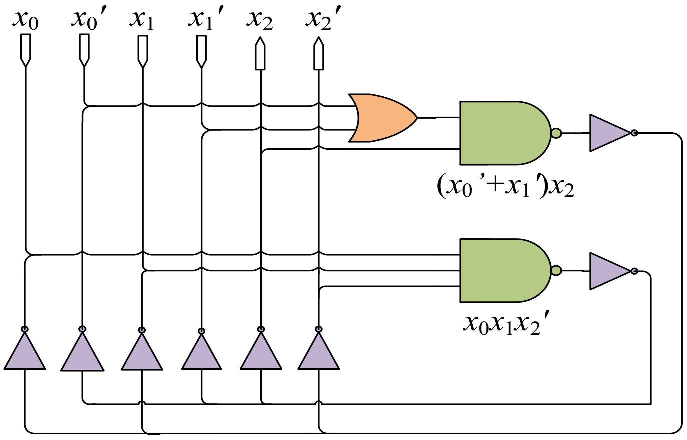


Fig 3.8: MRF NAND gate

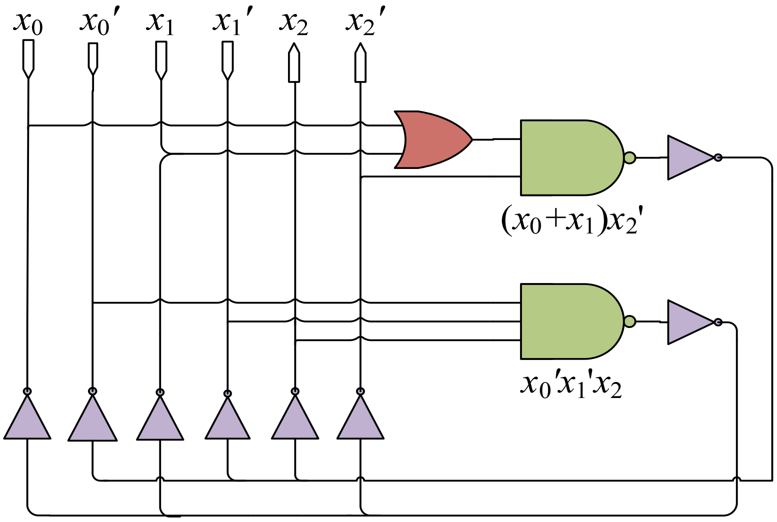


Fig 3.9: MRF NOR gate

3.3.1.1 How the MRF Conversion Rules Enforce Maximum Joint Probability?

Now the discussion will be provided on the mechanism by which the conversion principles stated in this section enforce joint probability principle.

The first conversion rule is used to enforce the correct logic states by enforcing whole of the correct logic combination of the gate/clique. For this reason, both of the original and complemented form of inputs and outputs are required. The idea is to align the output corresponding to the given inputs so that noise disturbance cannot flip the bit values as they are re-enforcing each other. This methodology is clearly in conjunction with the message passing example explained in Sec. 3.2.1.3.

The second design rule uses the feedback mechanism to enforce the correct output corresponding to inputs. If the input reverts due to noise, whole of the logic combination of the gate or clique reverts, resulting in diversion to the wrong (but still maximum joint probability) combination. This is prevented by using the feedback mechanism.

3.3.2 Mapping Marginal Probability Power Dissipation Principle on Digital Hardware

The marginal probability principle targets minimum power dissipation in the circuit. MRF circuit designed using the joint probability principle is already an example of limiting power dissipation as it limits the distortion caused by the circuit noise. The lesser the signal variation in the circuit, the lesser would be the power dissipation of the circuit.

The MRF design proposed in [47] still lacks in presenting the minimum possible power dissipation architecture. Since the contacts between the inputs and feedback paths serve as the fan-in points having resistance, the distortion in the digital circuit increases. This resistance can be removed by adding up AND gates on these joints with its two inputs being the input of the MRF gate and the feedback path. The reason behind replacing contact resistance with AND gate (composed of transistors) is that the transistor behaves as a switch in digital circuit i.e. in ideal case, it has a zero on-state resistance. Although in real operation, it still poses a non-zero resistance though the magnitude of this resistance is much lower than the contact resistance. Therefore, this modification is proposed (in this thesis) as an extra conversion rule in addition to the two rules proposed in [23] and mentioned in Sec. 3.3.1.

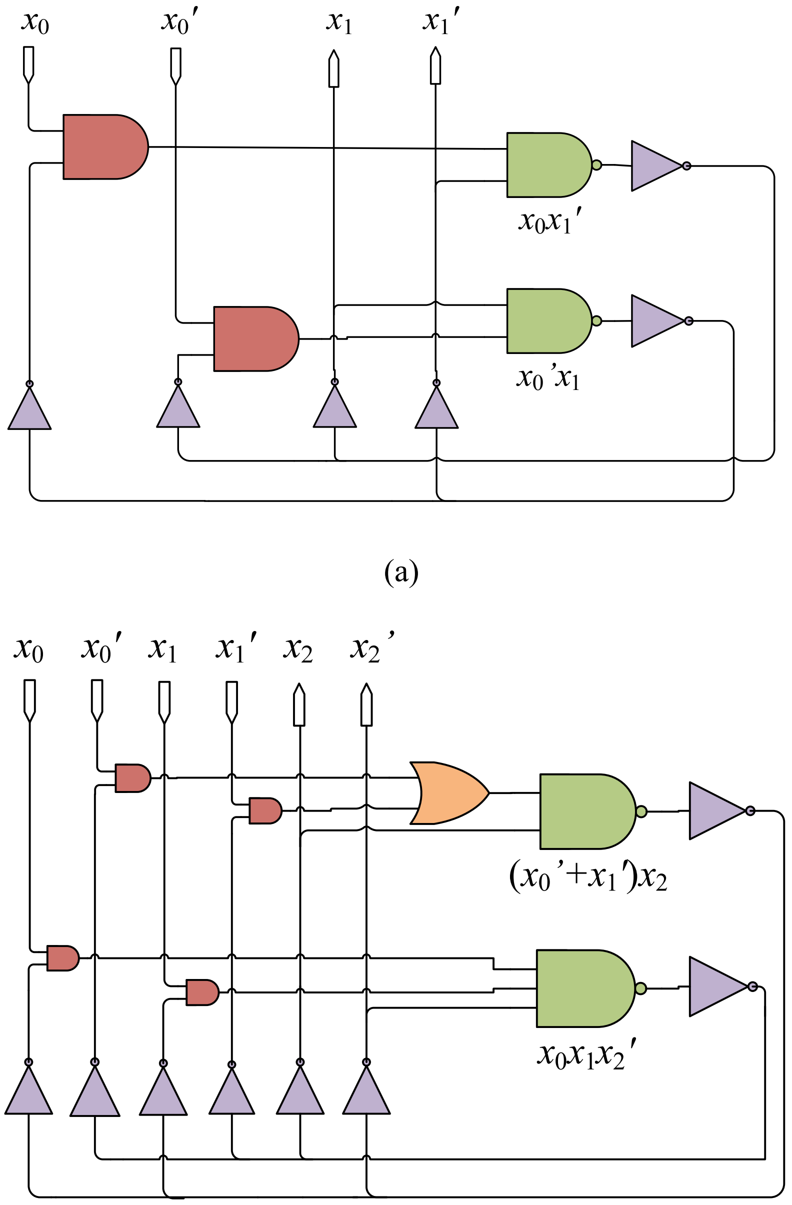
> **Rule 3:**
>
> Replace the joints connecting inputs and feedback loops with AND gates with both the input and feedback loop being the two inputs of the AND gate.

Using the third conversion rule, the MRF logic gates are renamed as ‘Improved-MRF’ logic gates. The Improved-MRF NOT and NAND gates are shown in Fig. 3.9. The noise immunity improvement obtained with following this extra conversion rule will be proved by simulations carried out in the next chapter.

3.4 Transistor-Count Comparison of CMOS and MRF-CMOS Designs

A comparison of transistor-count in the three design methodologies is shown in Table 3.5. With reference to this table, the NAND gate size, for example, increased from

CMOS to Improved-MRF, by a factor of 15 which may seems alarming but if the area consumption improvement obtained from scaling transistor dimensions is considered, the overall area-efficiency would still be maintained e.g. the 4-transistor logic gate (NAND or NOR) has an area of 0.35 μm^2 in 2010 which is expected to shrink up to 0.01 μm^2 in 2024 i.e. a decrease of chip size by almost 35 times [61]. Therefore, the area overhead of MRF would still remain less than the area-efficiency obtained though scaling transistor dimensions.



(a)

(b)

Fig 3.9: Improved-MRF (a) NOT (b) NAND

Table 3.5 Comparison of transistor-count in different circuit designs

	CMOS	MRF	Improved-MRF	MRF/ CMOS	Improved-MRF/ CMOS
Inverter	2	22	34	11.0	17.0
NAND	4	36	60	9.00	15.0
NOR	4	36	60	9.00	15.0

3.5 Discussion

The huge mathematical work involved in MRF theory gives an impression that the MRF computation would become very complex as bigger circuits are used. Fortunately, this would not happen since the clique-independence helps us to compute individual gate joint probabilities instead of necessarily finding the whole network joint probability. The maximization of each gate's joint probability ensures maximum probability of the whole network automatically.

To prove the noise-immunity improvement of MRF logic elements over simple CMOS counterparts, noisy inputs will be applied for the purpose of conducting simulations. Thus, the noisy signal construction and the results of simulations will all be covered in the following chapter.

CHAPTER 4

VALIDATING MRF CIRCUIT'S PERFORMANCE

To prove the worth of MRF gates constructed in the previous chapter and analyze their noise-tolerance capability, Spectre circuit simulator (in Cadence) is used with its interface known as Analog Design Environment.

4.1 Simulation Setting

The process of simulations was initiated by setting up the parameter environment. The input signal construction requires developing noise models (thermal and RTS) using the VerilogAMS hardware description language (HDL). The noise models are applied to the simulation framework after which the simulation criteria are set for rigorous analysis. Note that the task of user-noise interface development is necessary as the noise analysis for digital circuits is not a built-in part of the software Cadence. The following sub-sections list the CMOS technology model used for analysis followed by the detailed procedure for development of noise models.

4.1.1 CMOS Technology Model

The CMOS technology model used is 32 nm bulk-CMOS (at Temp=27℃). The NMOS and PMOS characteristic files for this technology were generated from Berkeley's Predictive Technology Models (PTM) [25]. The 32 nm technology was used because it is the latest and most downscaled version of BSIM4 (Berkeley Short Channel IGFET* Model, Version 4) available on this website. Note that the 22 nm technology is also available on this website though it does not provide the desired I-V

---

*Insulated gate field-effect transistor

characteristics, hence it was not selected for the analysis. The main features of these transistors are shown in Table 4.1.

Table 4.1: CMOS *32 nm* predictive technology parameters

L_g = Transistor Gate Length	32 nm
L_{eff} = Effective channel length	12.6 nm
V_{th} = Threshold voltage	0.16 V (for NMOS) and -0.16 V (for PMOS)
V_{dd} = Supply voltage	0.9 V
T_{ox} = Oxide thickness	1 nm

4.1.2 Noise Models

In this section, the thermal and RTS noise sources will be modelled. The combined effect of both noises will form the simulation setup for injecting input noise in the CMOS and MRF-CMOS gates.

4.1.2.1 Thermal Noise

To represent the thermal noise, a Gaussian noise source is placed at the input of each MRF gate. Another possible approach was to calculate the thermal noise data (from the literature and formulae) for noise originating from every transistor and interconnect. But instead, a simplified model of thermal noise is used representing the noise as a lumped source placed at each input of gate [47]. Hence, this noise source is accounted for the thermal noise generated from all the components in the previous circuit gate/stage. In this way, the thermal noise effect of the previous stage is supposed to be absorbed by the MRF equivalent of the current stage. To generate the Gaussian/thermal noise data, a MATLAB-based Gaussian noise function (derived from [62]) is used. This function can be represented in Eq. (4.1) and the parameters' values are shown in Table 4.2.

$$\text{Magnitude (Thermal Noise)} = f\,(\text{Mean, Standard Deviation, Nominal Voltage}) \quad (4.1)$$

Table 4.2: Parameter setting of thermal noise function

Mean	0 V
Standard Deviation	0.3 V
Nominal Voltage	0 V

The Eq. (4.1) calculates the thermal noise based on the statistical parameters of Gaussian noise function. These parameters include mean, standard deviation and nominal voltage of the noise (waveform). For this analysis, the noise is assumed to be stationary i.e. it has a zero mean so that the variation of noise above and below the mean value are balanced. The high value of standard deviation i.e. 0.3 V was selected based on the empirical noise data generation for which the logic voltage levels were observed to cross the acceptable noise margin at numerous times.

The nominal voltage is an offset value of the voltage which simply adds a DC voltage level to the noise data. Since the offset values required for the noise model would be voltage of the logic levels i.e. 0 V for logic 0 and 0.9 V for logic 1, a nominal voltage of 0 V is applied so that it does not disrupt the noise inclusion process. Moreover, it is assumed that each input of the gate is subjected to similar noise magnitude; therefore, the same noise data for all input noise sources will be used.

To apply the noise data (generated from the MATLAB noise function) at the input of MRF gate, the VPWLF (voltage piecewise linear file) function is used which is provided for the external noise data inclusion in Cadence. The VPWLF source adds the noise sample to the input voltage present at each time sample. For this analysis, the input waveform generated has a period of 20 microseconds, with noise frequency of 100 samples per microsecond which generates a highly noisy input signal waveform shown in Fig. 4.1.

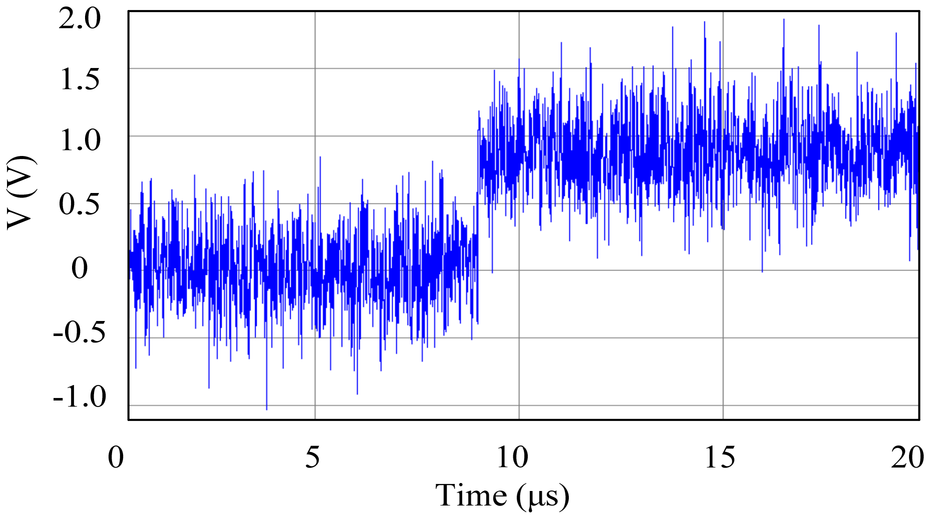


Fig 4.1. Thermal Noise representation

4.1.2.2 Random Telegraph Signal (RTS) Noise

The RTS noise is generated by using the noisy transistor models developed by the Monte Carlo simulation method [24, 40]. According to this method, a noise source is attached from drain to source terminals adding the RTS noise in the current flowing between these two terminals. Since these models have been previously programmed for the 90 nm technology [24], the Verilog code (of the RTS models) have been modified to reflect the 32 nm technology parameters which are shown in Table 4.1. Moreover the code has been transformed from VerilogA to VerilogAMS HDL being the more advanced hardware description language in current Cadence versions. The NMOS I-V characteristics for a range of V_{GS} have been plotted as shown in Fig. 4.2. The current variation in this figure accounts for the random trapping or detrapping of charge carriers near the Si-SiO$_2$ interface which is called as RTS noise [24]. When a trap receives an electron, the output current decreases and hence the voltage drops whereas the reverse case happens when the trap releases an electron.

Since the voltage is derived from current, the RTS noise effect can be seen in the voltage waveform of inverter (as example) as shown in Fig. 4.3. It can be observed from the figure that the voltage level is preserved at the higher energy state due to release of electrons. The reverse case happens when the trap captures an electron thus resulting in the current and hence voltage level drop.

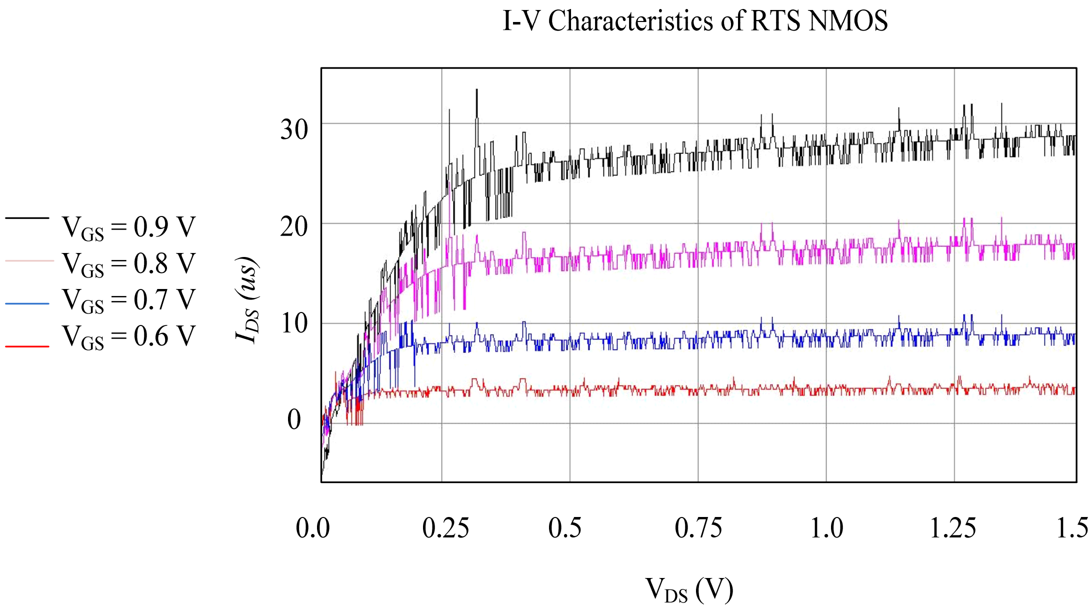

I-V Characteristics of RTS NMOS

Fig 4.2: RTS noise representation in NMOS drain current

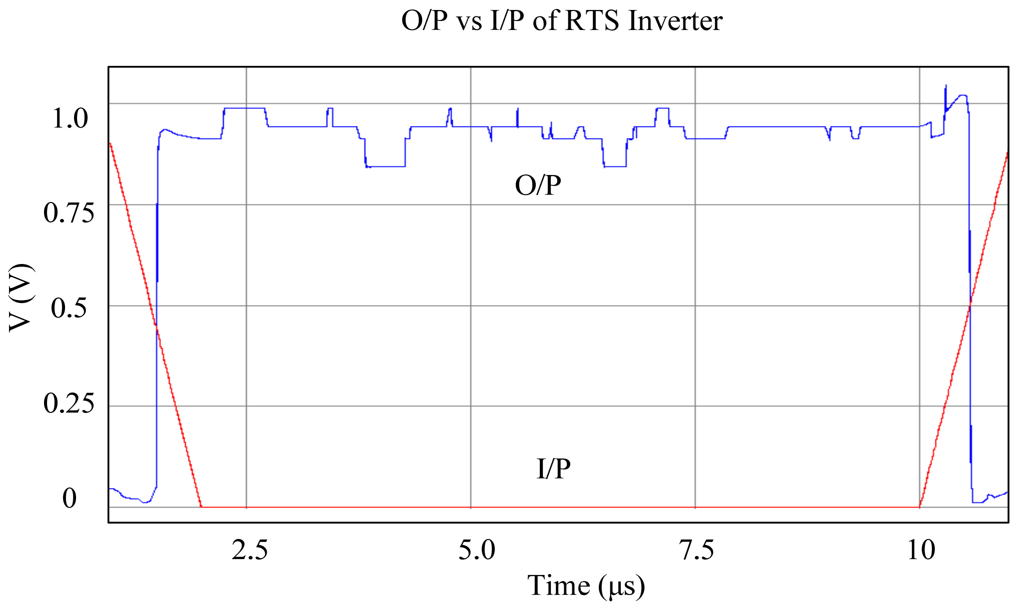

O/P vs I/P of RTS Inverter

Fig 4.3: RTS noise effect in O/P v/s I/P waveform of inverter

4.1.2.3 Difference Between Thermal and RTS Noise Injection Methods

The difference between thermal and RTS noise modelling is the way these noises are injected in the circuit. For thermal noise representation, the noise is injected (as a lumped source) at the input to represent the previous circuit/stage noise. For RTS model, the noise is added at the output (drain current) of each transistor. The mechanism of noise addition is illustrated for a simple CMOS inverter as shown in Fig. 4.4. This method of noise modelling cause variations in drain (to source) current under combined effect of both thermal and RTS noises.

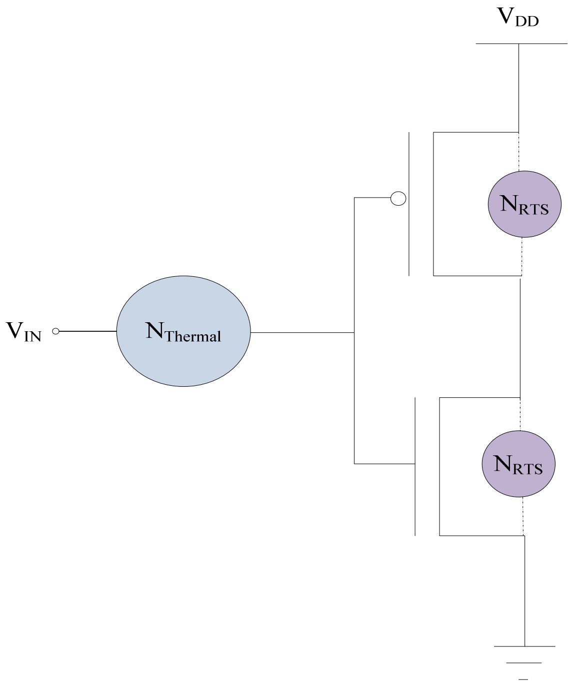


Fig 4.4: Thermal and RTS noise inclusion mechanisms

4.1.3 CMOS Inverter v/s MRF-CMOS Inverter

The first analysis was performed for simple CMOS and MRF-CMOS inverters. The results are shown in Fig. 4.5. The voltage levels used are 0 V and 0.9 V (equal to power supply voltage) for logic 0 and 1 respectively.

It can be observed from the figure that for extremely noisy input signal, the CMOS inverter output is very unstable whereas the MRF inverter output shows very little distortion at both logic levels. This clearly explains the worth of noise immune MRF design. The MRF, on one hand, not only eliminates the possibility to encounter bit reversals but also offers very low output distortion i.e. the distortion in the output still enables the signal interpreter to clearly distinguish between logic 0 and 1.

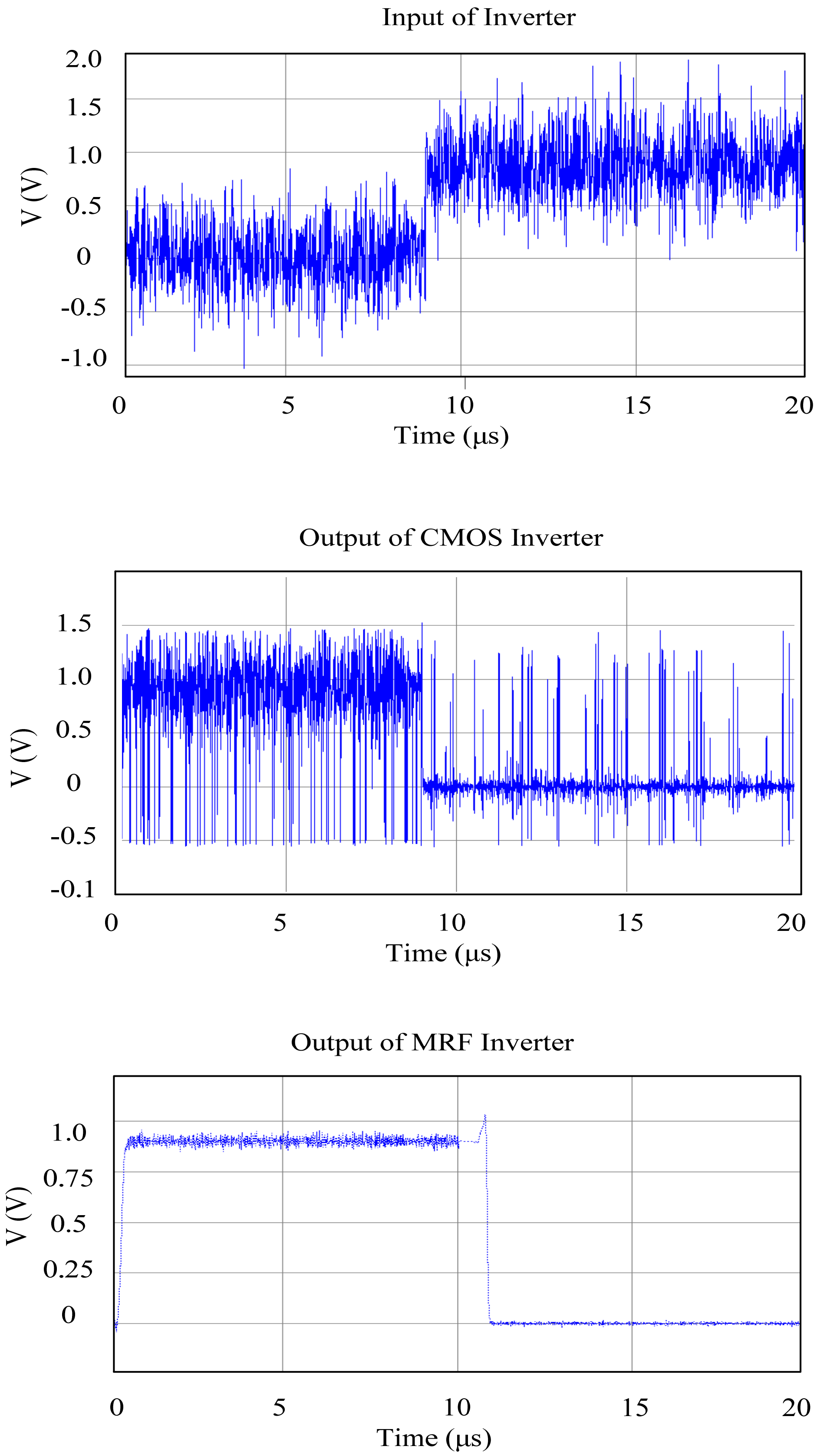


Fig 4.5: Simulations of CMOS Inverter v/s MRF Inverter

4.1.4 CMOS NAND v/s MRF-CMOS NAND

The results for the NAND (2-input) gate are shown in Fig. 4.6. It is evident that the output of the simple CMOS NAND shows enormous distortion whereas the MRF NAND is hardly affected by the input noise. Hence, by using MRF NAND in place of CMOS NAND would account for zero bit errors which therefore maintain the maximum joint probability of NAND gate and the overall circuit using MRF gates.

4.1.5 Simulations with Improved-MRF Design

In this section, the noise analysis of the Improved-MRF inverter will be performed (designed in Sec. 3.3.2). Following the Conversion Rule 3, which accounts for adding AND gates at the joints of inputs and feedback loops, the output waveform gets smoother and looks like the ideal waveform that can be expected to obtain without noise addition which is manifested in Fig. 4.7. The implementation of this rule costs few more transistors (depending on the number of fan-in joints) which are actually the extra hardware to be used as a tradeoff for extra reliability.

4.1.6 Quantifying MRF Noise Tolerance

Besides the noise-immune behaviour depicted by the simulation waveforms, there is need to measure the output distortion so that the specific factor can be calculated by which the MRF technique is more noise-tolerant than CMOS. But, the quantification of noise-tolerance of digital signals in terms of distortion level could not be carried out by the Cadence Analog Design Environment. Hence, the statistical measure, root-mean-square (RMS) variation is used to measure the average distortion in output voltage. The RMS variation method was selected as it is the standard way to measure the signal variation particularly when the signal waveform has discrete samples [63-65]. The RMS output voltage variation of seven sample circuits (whose logic diagrams and simulation results are shown in Appendix A and B respectively) for their CMOS, MRF and Improved-MRF alternatives is presented in Table 4.3. The output variation is calculated against the input voltage variation of 600 mV.

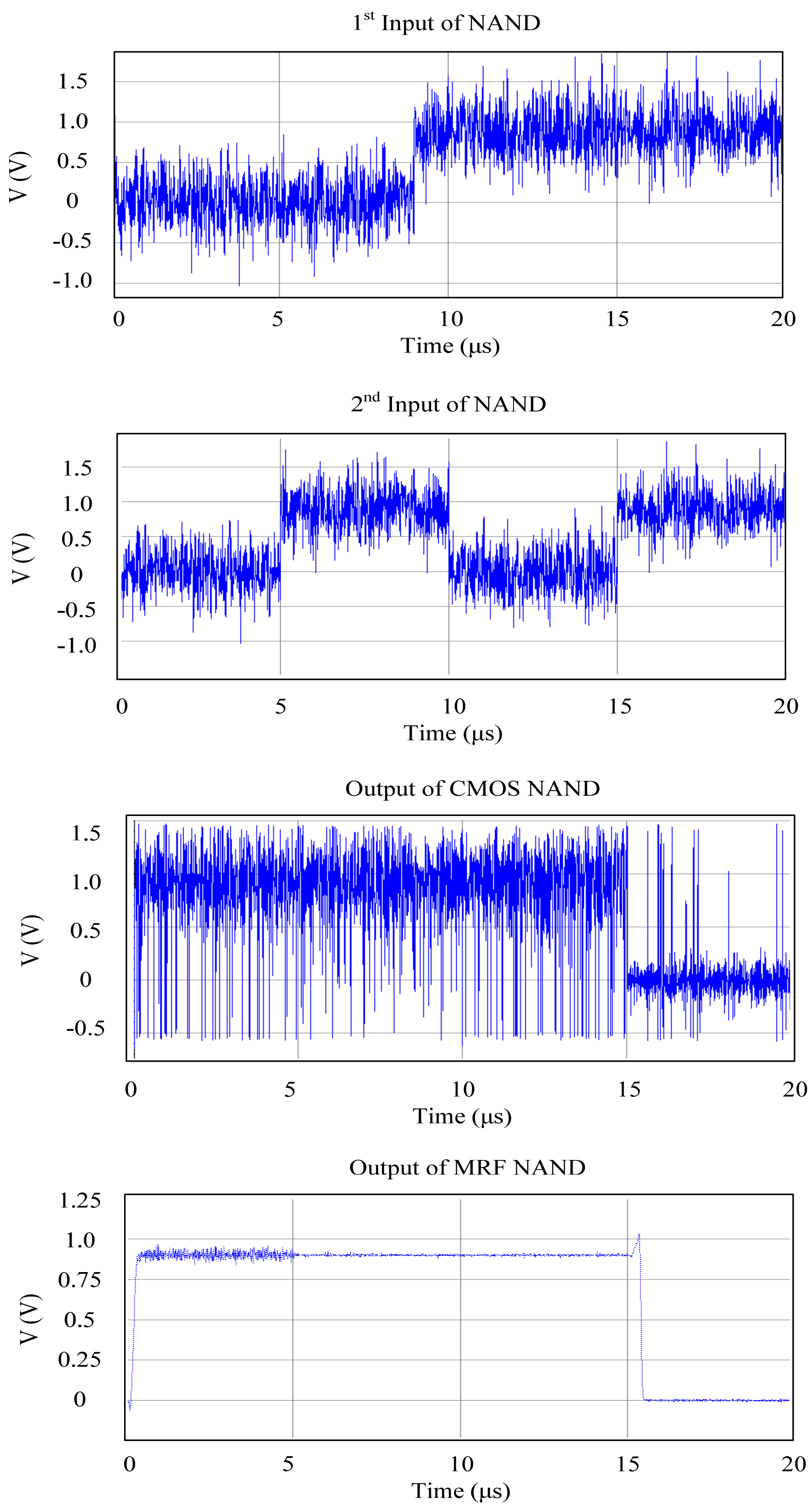
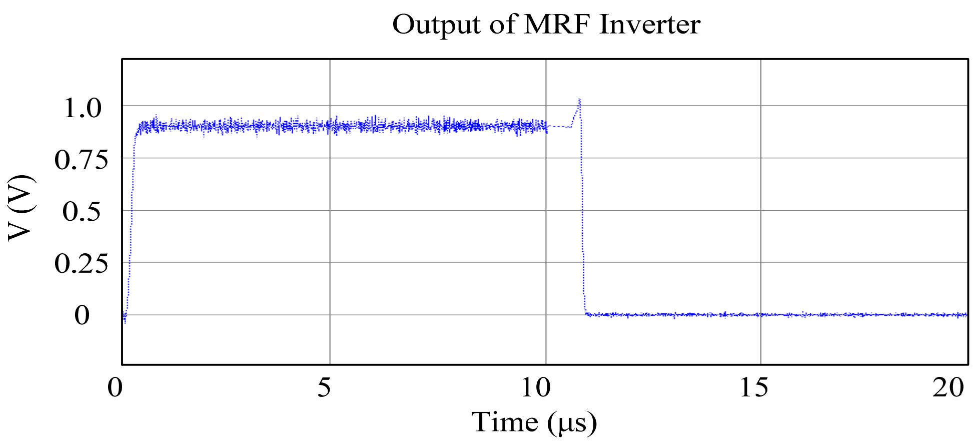


Fig 4.6: Simulations of CMOS NAND v/s MRF NAND


Output of MRF Inverter

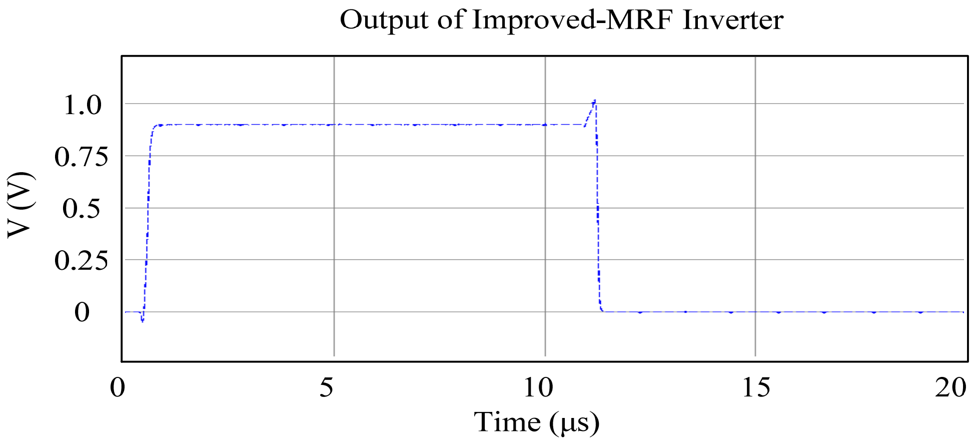

Output of Improved-MRF Inverter

Fig 4.7 Simulations of MRF Inverter v/s Improved-MRF Inverter

Table 4.3: RMS voltage variation (mV) of circuits against input variation of 600 mV

	CMOS	MRF	Improved-MRF	CMOS/ MRF	CMOS/ Improved-MRF
Inverter	389.1	25.37	0.705	15.33	551.9
NAND	473.9	40.92	0.653	11.58	725.7
NOR	489.6	47.01	0.727	10.41	673.4
C17	307.2	30.28	0.396	10.14	775.7
Dec 2x4	353.7	28.36	0.624	12.47	566.8
Mux 4x1	333.1	20.23	0.428	16.46	778.3
Full Adder	407.5	17.44	0.943	23.36	432.1

The results in Table 4.3 show that the output voltage variation for MRF gates is atleast 10 times less than the CMOS designs or in other words, the MRF gates are nearly 10 times more noise-tolerant than the CMOS alternatives (as noise-tolerance is inversely proportional to output voltage variation). Likewise, Improved-MRF design is more than 400 times noise-efficient as compared to its CMOS counterparts. Hence, the MRF design technique could ensure the noise efficiency attained for a circuit design even if the circuit is operated under highly noisy conditions.

4.1.7 Transistor-Count of CMOS, MRF and Improved-MRF Designs

The numbers of transistors used in the CMOS, MRF and Improved-MRF circuit designs have been tabulated in Table 4.4. As shown in the table, the number of transistors in MRF design exceeds the CMOS by atleast 10 times whereas this factor reaches up to 14.9 times for Improved-MRF design. On recalling the device-density improvement obtained by the nanoscale design (as discussed in Sec. 3.4), it was shown that the area of 4-transistor logic gate, for example is scaled down by 35 times in a span of 14 years (as shown by the ITRS website [61]). On the other hand, the implementation of fault-tolerant architecture demands an increase of almost 15 times in the number of gates. Hence, it can be concluded that the improved device density is still achievable by the implementation of MRF architecture.

4.1.8 Circuit's Reliability versus Transistor-Count

There is no fixed criterion found in the literature that relates the reliability of a circuit to its transistor-count. Note that the term reliability is the antonym for the noise distortion. The lesser the signal or noise distortion, the higher is the reliability of the circuit. Therefore, a relation between the circuit's reliability and the transistor-count will be developed based on the understanding of these two parameters and their respective effects on digital circuits. For this purpose, a factor called as Reliable-Area Index (RAI) is introduced. The high value of this index refers to an efficient circuit design that maintains an acceptable tradeoff between circuit's reliability and area consumption. Firstly, the circuit's area consumption is inversely proportional to this index. The reason is that the smaller circuit is always area-effective (as discussed in Sec. 1.1). Secondly, the reliability of the circuit is directly proportional to this index (in order to provide this index a high value). These two conditions make us form the Eq. (4.2).

$$\text{Reliable Area Index (RAI)} = \frac{\text{Reliability}}{\text{Circuit's Area Consumption}} \tag{4.2}$$

Since, the output signal variation (RMS variation) is the reverse of the reliability of the circuit and the transistor-count is analogous to the circuit's area consumption, therefore, the reliable area index is reformed as shown in Eq. (4.3).

$$\text{Reliable Area Index (RAI)} = \frac{1}{\text{(Transistor Count) (RMS Output Variation)}} \tag{4.3}$$

The values of RAI are tabulated for all the target circuits in Table 4.5. It can be observed from the table that the MRF is more than 1.1 times whereas the Improved-MRF is atleast 29.4 times higher RAI as compared to the CMOS alternatives. The two observations obtained from this table are as follows.

Table 4.4: Number of transistors used in CMOS, MRF and Improved-MRF designs

	CMOS	MRF	Improved-MRF	MRF/ CMOS	Improved-MRF/ CMOS
Inverter	2	22	34	11.0	17.0
NAND	4	36	60	9.00	15.0
NOR	4	36	60	9.00	15.0
C17	24	216	360	9.00	15.0
Dec 2x4	28	276	444	9.86	15.9
Mux 4x1	38	406	658	10.7	17.3
Full Adder	62	610	924	9.84	14.9

Table 4.5: RAI Values for CMOS, MRF and Improved-MRF designs

	CMOS	MRF	Improved-MRF	MRF/ CMOS	Improved-MRF/ CMOS
Inverter	1.285	1.792	41.71	1.39	37.1
NAND	0.527	0.679	25.52	1.29	48.4
NOR	0.511	0.591	22.93	1.16	44.9
C17	0.136	0.153	7.015	1.13	51.6
Dec 2x4	0.101	0.128	3.609	1.27	35.7
Mux 4x1	0.079	0.121	3.551	1.53	44.9
Full Adder	0.039	0.094	1.148	2.41	29.4

(a) As the circuit size becomes large (ranging from inverter to full-adder), the RAI keeps decreasing (for each design technique) which means that the efficiency of the circuit design would decrease with the circuit size anyway.

(b) The RAI always exceeds from CMOS to Improved-MRF thereby concluding MRF to be always superior to CMOS design regardless of circuit size.

4.1.9 CMOS Technology-Independence of MRF Design

To investigate the effect of using different CMOS technologies on the noise-tolerance principle of MRF design, equivalent circuits of inverter (for CMOS, MRF and Improved-MRF) were simulated for three sample CMOS technologies i.e. 600 nm, 180 nm and 32 nm (with the typical power supply voltages of 4.5 V, 3.3 V and 0.9 V respectively). In order to compare the noise-tolerance capability of target technologies, the same noisy signal is used as an input for each technology simulations. The results obtained are shown in Table 4.6.

Table 4.6: RMS Variation of Inverter output using different CMOS technologies

	CMOS	MRF	Improved-MRF	MRF/ CMOS	Improved-MRF/CMOS
600 nm (4.5 V)	10.54 mV	1.564 µV	45.54 nV	6739	$2.31\text{x}10^5$
180 nm (3.3 V)	54.47 mV	14.71 µV	85.66 nV	3704	$9.33\text{x}10^8$
32 nm (0.9 V)	389.1 mV	25.37 mV	0.705 mV	15.34	552

The observations derived from Table 4.6 are three fold.

(a) The noise-distortion is reduced in a similar fashion for each CMOS technology i.e. the noise distortion is reduced from CMOS to MRF which further reduces from MRF to Improved-MRF. Therefore, the noise-tolerance mechanism of MRF technique is CMOS-technology-independent.

(b) The factor by which the noise-immunity differs among the three design mechanisms are based on each CMOS technology e.g. the MRF is noise-tolerant than CMOS scheme on the order of 6739, 3704 and 15.34 times for 600 nm, 180 nm and 32 nm technologies respectively.

(c) As the technology scales down i.e. from 600 nm to 32 nm, the noise-tolerance capability of CMOS, MRF and Improved-MRF all decrease. It can be observed that for 600 nm, the noise variation of 10.54 mV is small enough to affect the logic 1 voltage of 4.5 V; hence the MRF design is not really a need for high-dimension technologies. Instead, a variation of 389.1 mV for logic 1 voltage of 0.9 V (for 32 nm technology) poses strong chances of bit errors for this technology. That is why, the MRF technique is particularly designed for use with deep submicron technologies.

CHAPTER 5

CONCLUSION

Based on the literature review and analysis of the technique i.e. Markov Random Field, a summary of important findings and contributions made towards the fault-tolerant design of nanoscale circuits is presented.

5.1 Conclusion

The benefits achieved from MOSFET scaling are improved device density, higher switching speed and decreased cost of an integrated circuit. By the time, circuit design enters into the nanoscale era particularly deep sub-micron design, the reliability of digital circuits come into question. The reason is the increased transient error-rate. Since the reliability of electronic applications cannot be sacrificed on the cost of availing above-mentioned benefits of nanoscale circuits; the circuit designers seek for a solution to this problem i.e. how to make use of unreliable nanoscale devices to design a reliable system. The solution is fault-tolerant circuit design.

The research on fault-tolerance can be divided into three categories i.e. reliability-evaluation schemes, architecture-level techniques and CAD tools development. Among these categories, the architecture-level option was selected as the other two categories fall beyond the scope of the research work in this thesis. Between the options available for architecture-level solutions i.e. redundancy and Markov Random Field (MRF), MRF was selected being the superior model in terms of reliability, error-handling capability and area efficiency as compared to redundancy.

The previous research on MRF lacks at the design, simulation framework and implementation levels. At the design level, computing procedures for the mathematical model of MRF are proposed based on the general outlines found in the previous MRF research. The mathematical analysis ended up with the development of fault-tolerance rules which were verified by conducting a special case study. The fault-tolerance rules, when compared to the MRF literature were found to be in total agreement.

At the simulation level, noise framework was extended from thermal to a combination of Random Telegraph Signal (RTS) and thermal noises. The reason for injection of these two particular types of noises only is their highest vulnerability to affect the future nanoscale technologies. At the implementation stage, an architecture-level improvement is also proposed that further improves the noise immunity of the circuit. The resulting logic gate designs have been proposed to fall under the novel category i.e. Improved-MRF design.

The logic components thus developed have been simulated in Cadence Analog Design Environment. Under the application of thermal and RTS noise sources, the output of CMOS, MRF and Improved-MRF gates were observed. The CMOS output shows numerous bit reversals whereas the MRF gates show a very little distortion in the output levels with no bit reversals. Improved-MRF gates are found to obtain nearly ideal outputs with un-noticeable distortion and zero bit errors as well. Therefore, among the three techniques, Improved-MRF was found to be highest noise-tolerant circuit design.

The noise-immune design of MRF has been described for the universal gates i.e. NAND, NOR and NOT. Since in normal CMOS design, every circuit is composed of universal gates, therefore, any circuit constructed using MRF design technique would be noise tolerant and reliable. The simulations performed for some sample large circuits showed that the Improved-MRF circuits are atleast 430 times more noise-tolerant than their CMOS alternatives. The tradeoff for the MRF design is the increase in transistor count by a factor of 17 for an inverter and 15 times for NAND and NOR gates. The increased transistor count, if compared to the significant decrease in the transistor dimensions (in Sec. 3.4) still promises area efficiency which will be achieved by utilizing future nanoscale technologies.

5.2 Suggested Future Work

The research work in this thesis could be extended in the following areas.

- The MRF design can be implemented at the layout and fabrication level. For this purpose, layout tools available in Cadence simulation software can be utilized.

- The fault-tolerance capability of MRF circuits can be evaluated by the reliability-evaluation techniques. The reason for not using these techniques for reliability measurement of MRF circuits (in this research work) is the absence of any fixed criteria to select the CMOS or MRF gate error probabilities. The mathematical models of reliability-evaluation techniques use arbitrary gate error-probability values for their simplified models available at this stage. By the time, the reliability-evaluation techniques get mature, they will be able to verify fault-tolerance capability of MRF.

- The MRF design, so far, is limited to only combinational circuits. Its methodology can be extended to sequential circuits as well. The combinational and sequential circuit conversion schemes can lead to MRF system design e.g. an MRF-based processor.

- The reliability-evaluation techniques are based on mathematical models only. Their automation via developing software toolbox is still pending. This software toolbox could be able to input circuit description in the form of netlist in order to perform the reliability-evaluation technique and provide the output error probability of the circuit. At this time, an initial process of integrating probabilistic gate model (PGM) technique with the software Xilinx ISE 8.1i is in progress under our research group.

REFERENCES

[1]. S. Wasson, “Intel’s Core 2 Extreme QX9650 Processor,” TechReport Website. [online] Available: http://techreport.com/articles.x/13470. [Accessed: Jan 2010]

[2]. P. V. Voorde, MOSFET Scaling into the Future, Article 12, Hewlett-Packard Journal, 1997.

[3]. S. Rusu, “Trends and Scaling in VLSI Technology Scaling Towards 100nm (Invited Paper),” in European Solid-State Circuits Conference (ESSCIRC), 2001.

[4]. J. M. Rabaey, Digital Integrated Circuits: A Design Perspective, 2nd Edition, Prentice Hall Publishers, Dec 2002.

[5]. D. C. Brock, Understanding Moore's law: Four Decades of Innovation, Chemical Heritage Foundation, 2006.

[6]. M. L. Shooman, Reliability of Computer Systems and Networks: Fault Tolerance, Analysis and Design, John Wiley and Sons, 2002.

[7]. R. Whitaker, The End of Privacy: How Total Surveillance is Becoming a Reality, The New Press, 1999.

[8]. D. Page, A Practical Introduction to Computer Architecture, Springer, 2009.

[9]. G. Moore, A. Grove, C. Barrett, L. Vadasz, T. Hoff, D. Frohman and F. Faggin, “Moore’s Law: An Intel’s Perspective,” Tech.rep, 2005. [online] Available: ftp: //download. intel. com/ museum/Moores_Law. [Accessed: Sep, 2009]

[10]. C. Huang, Robust Computing with Nanoscale Devices: Progresses and Challenges, Springer, 2010.

[11]. M. Stanisavljevic, A. Schmid and Y. Leblebici, Reliability of Nanoscale Circuits and Systems: Methodologies and Circuit Architectures, Springer, 2010.

[12]. T. Ryhänen, M. A. Uusitalo, O. Ikkala and A. Kärkkäinen, Nanotechnologies For Future Mobile Devices, Cambridge University Press, 2010.

[13]. G. Casati and D. Matrasulov, Complex Phenomena in Nanoscale Systems, Springer, 2009.

[14]. J. C. Wooley and H. Lin, Catalyzing Inquiry at the Interface of Computing and Biology, National Academies Press, 2005.

[15]. K. Nikolic, A. Sadek and M. Forshaw, “Architectures for Releiable Computing With Unreliable Nanodevices,” in IEEE Conference on Nanotechnology (IEEE-NANO), 2001.

[16]. M. Tehranipoor, Emerging Nanotechnologies: Test, Defect-Tolerance and Reliability, Volume 37 of Frontiers in Electronic Testing, 2008.

[17]. H. Iwai, “Future of Nano CMOS Technology,” in Proc. International Workshop on Electron Devices and Semiconductor Technology, IEDST 2007, China.

[18]. J. M. P. Cardoso and P. C. Diniz, Compilation Techniques for Reconfigurable Architectures, 2008.

[19]. M. Horowitz, “Scaling, Power and the Future of CMOS,” in Proc. 20th International Conference on VLSI Design (VLSID), 2007.

[20]. D. A. Rennels, Fault-Tolerant Computing, Encyclopedia of Computer Science, Editors: A. Ralston, E. Reilly and D. Hemmendinger, 1999.

[21]. S. Ahuja, G. Singh, D. Bhaduri and S. K. Shukla, “Fault and Defect-Tolerant Architectures for Nano-computing,” in Bio-Inspired and Nanoscale Integrated Computing, Mary Eshaghian-Wilner, Wiley, 2009.

[22]. 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. K. Shukla and R.I. Bahar, Springer, 2004.

[23]. K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson and A. Zaslavsky, “Designing Nanoscale Logic Circuits Based on Markov Random Fields,” Journal of Electronic Testing: Theory and Applications, vol 23, pp. 255–266, Jun 2007.

[24]. N. H. Hamid, “Modelling Noisy MOSFETs,” in Can Deep-Sub-Micron Device Noise be Used As The Basis for Probabilistic Neural Computation ?, PhD Dissertation, The University of Edinburgh, 2006.

[25]. Predictive Technology Models Website: http://ptm.asu.edu/ [Accessed: Jan 2010]

[26]. T. Rejimon, K. Lingasubramanian and S. Bhanja, “Probabilistic Error Modeling for Nano-Domain Logic Circuits.” in IEEE Transactions on Very Large Scale Integration (VLSI) Systems. vol 17, no 1, USA, Jan 2009.

[27]. I. Koren and C. M. Krishna, Fault-Tolerant Systems, Morgan-Kaufman Publishers, San Francisco, CA, 2007.

[28]. T. Lehtonen, J. Plosila and J. Isoaho, “On Fault-Tolerance Techniques towards Nanoscale Circuits and Systems,” TUCS Technical Report, No. 708, Aug 2005.

[29]. “Transient Error Protection: The Smarter Approach to Uptime,” Whitepapers, Stratus Technologies [online] Available: www.stratus.com/pdf/whitepapers/ TransientErrorProtection.pdf. [Accessed: Dec 2009]

[30]. D. Bhaduri and S. Shukla, “Reliability Analysis of Fault-Tolerant Reconfigurable Architectures,” FERMAT, Tech.rep. 2004-15, 2004.

[31]. W. K. Chen, The VLSI Handbook, 2nd Edition, CRC Press, 2007.

[32]. M. A. Elgamel and M. A. Bayoumi, Interconnect Noise Optimization in Nanometer Technologies, Springer (Science and Technology), 2006.

[33]. N. Ekekwe, “Interconnection Noise Sources and Reductions in Nanometer CMOS,” Ezine Articles [online] Available: http://ezinearticles.com/ Interconnection-Noise-Sources-and-Reductions-in-NanometerCMOS&id =4473484. [Accessed: Oct 2009]

[34]. J. Sosnowski, “Transient Fault-Tolerance in Digital Systems,” in Proc. IEEE Micro, vol 14(1), USA, 1994.

[35]. J. Yan and W. Zhang, “Evaluating Instruction Cache Vulnerability to Transient Errors,” in Proc. 2006 workshop on Memory performance: Dealing with Applications, Systems and Architectures, pp 21-28, USA, 2006.

[36]. “Scaled CMOS Technology-Reliability Users Guide,” NASA Tech.rep. [online] Available: http://trsnew.jpl.nasa.gov/dspace/bitstream/2014/40765/1/ 08-014.pdf. [Accessed: Sep 2009]

[37]. S. R. Delbruck and T. Mead, “White Noise in MOS Transistors and Resistors,” in IEEE Circuits and Devices Magazine, vol 9, no. 6, 1993.

[38]. H. Li, J. Mundy, W. R. Patterson, D. Kazazis, A. Zaslavsky, and R.I. Bahar, "A Model for Soft Errors in the Subthreshold CMOS Inverter," in Proc. Workshop on System Effects of Logic Soft Errors, Nov. 2006.

[39]. K. K. Hung, P. K. Ko, C. Hu and Y. C. Cheng, “A Unified Model for the Flicker Noise in Metal Oxide Semiconductor Field Effect Transistors,” in IEEE Transactions on Electron Devices, vol 37, no. 3, Mar 1990.

[40]. N. H. Hamid, A. F. Murray, S. Roy, “Time-Domain Modeling of Low-Frequency Noise in Deep-Submicrometer MOSFET,” in IEEE Transactions on Circuits and Systems- I: Regular Papers, vol 55, no. 1, Feb 2008.

[41]. S. Krishnaswamy, G. F. Viamontes, I. L. Markov and J. P. Hayes, “Accurate Reliability Evaluation and Enhancement via Probabilistic Transfer Matrices,” in Proc. Design, Automation and Test in Europe, pp: 282-287. 2005.

[42]. S. Krishnaswamy, G. F. Viamontes, I. L. Markov and J. P. Hayes, “Probabilistic Transfer Matrices in Symbolic Reliability Analysis of Logic Circuits,” in ACM Transactions on Design Automation of Electronic Systems (TODAES**), v**ol 13, no. 1, Jan 2008.

[43]. J. Han, E. Taylor, J. Gao and J. Fortes, “Faults, Error Bounds and Reliability of Nanoelectronic Circuits,” in Proc. 16th International Conference on Application-Specific Systems, Architecture and Processors (ASAP’05), pp. 247-253, 2005.

[44]. N. Mohyuddin, E. Pakbaznia and M. Pedram, “Probabilistic Error Propagation in Logic Circuits Using the Boolean Difference Calculus.” in Proc. 26th International Conference on Computer Design, ICCD, pp. 7-13, USA, 2008.

[45]. C. J. Hescott, D. C. Ness and D. J. Lilja, “MEMESTAR: A Simulation Framework for Reliability Evaluation over Multiple Environments,” in Proc. 8th International Symposium on Quality Electronic Design, pp 917-922, 2007.

[46]. A. Beg and W. Ibrahim, “On Teaching Circuit Reliability,” in IEEE 38th Frontiers in Education Conference, USA, 2008.

[47]. K. Nepal, “Markov Random Field,” in Designing Reliable Nanoscale Circuits Using Principles of Markov Random Fields, PhD dissertation, Brown University, RI 02912, US, 2007.

[48]. K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson and A. Zaslavsky, “MRF Reinforcer: A Probabilistic Element for Space Redundancy in Nanoscale Circuits,” in IEEE Micro, IEEE Computer Society Magazine, vol. 26, no. 5, 2006.

[49]. R.E. Lyons and W. Vanderkulk, “The Use of Triple-Modular Redundancy to Improve Computer Reliability,” IBM Journal of Research and Development, vol. 6, no. 2, Apr. 1962, pp. 200-209.

[50]. N. Pippenger, “Reliable Computation by Formulas in the Presence of Noise,” IEEE Transactions of Information Theory, vol. 34, no. 2, pp. 194-197, Mar. 1988.

[51]. M. Favalli and C. Metra, “TMR Voting in the Presence of Crosstalk Faults at the Voter Inputs,” in IEEE Trans. Reliability, vol. 53, no. 3, pp. 342-348, Sept. 2004.

[52]. G. W. Brown, P. Thai, “Redundant Memory Circuit and Method of Programming and Verifying the Circuit”. US Patent 4577294, Mar 1986.

[53]. F. Spitzer, Markov Random Fields and Gibbs Ensembles, The American Mathematical Monthly, vol. 78, no. 2, pp. 142-154, Feb 1971.

[54]. R. Kindermann, Markov Random Fields and their applications, American Mathematical Society, Rhode Island, 1980.

[55]. S. Z. Li, Markov Random Field Modeling in Computer Vision. Berlin: Springer - Verlag, 1995.

[56]. J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” Journal of the Royal Statistical Society, series B, vol. 36, no. 2, pp. 192-236, 1974.

[57]. J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, USA, 1988.

[58]. I. C. Wey, Y. G. Chen, C. Yu, J. Chen and A. Y. Wu, "A 0.13μm Hardware Efficient Probabilistic-Based Noise-Tolerant Circuit Design and Implementation With 24.5dB Noise-Immunity Improvement," in Proc. IEEE Asian Solid-State Circuits Conference (ASSCC), pp. 295-298, Korea, 2007.

[59]. B. S. Everitt, The Cambridge Dictionary of Statistics, Cambridge University Press, 2002.

[60]. J. Yedidia, W. Freeman, and Y. Weiss, "Understanding Belief Propagation and its Generalizations," in Exploring Artificial Intelligence in the New Millennium, G. Lakemeyer and B. Nebel, Morgan Kaufmann, 2003.

[61]. "International Technology Roadmap for Semiconductors, ORTC Tables," 2009 Edition. [online] Available: http://www.itrs.net/Links/2009ITRS [Accessed: Feb 2010]

[62]. E. Taylor, "GenerateGuassianInputNoise.m". [Online] Available: http://ertaylor.wordpress.com/generategaussianinputnoisem [Accessed: Oct, 2009].

[63]. D. F. Hendry, M. S. Morgan, The Foundations of Econometric Analysis, Cambridge University Press, 1997.

[64]. K. Williston, Digital signal processing: World Class Designs, Newnes, 2009.

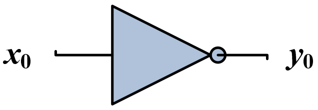
[65]. Books Llc, Statistical Deviation and Dispersion: Standard Deviation, Variance, Interquartile Range, Algorithms for Calculating Variance, Kurtosis, General Books LLC, 2010.

APPENDIX A

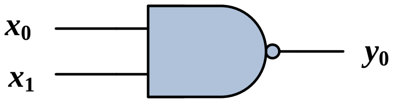
LOGIC DIAGRAMS OF TEST CIRCUITS

- The subscripts ‘*x*’ and ‘*y*’ denote inputs and outputs respectively.

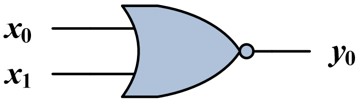
a) Inverter (NOT)



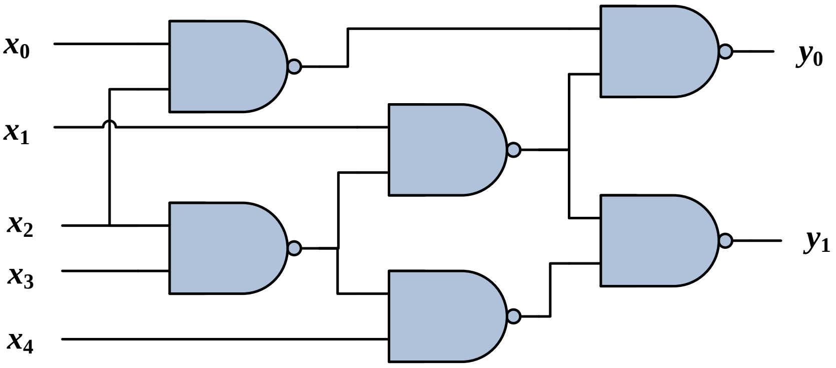
b) NAND



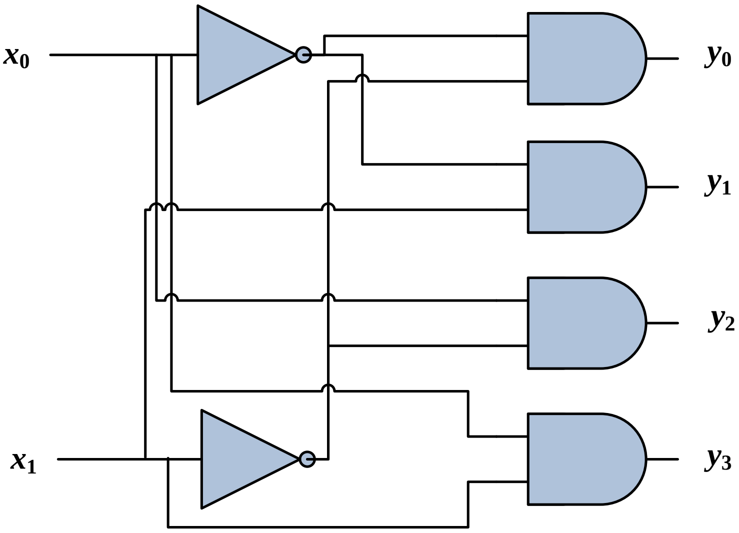
c) NOR



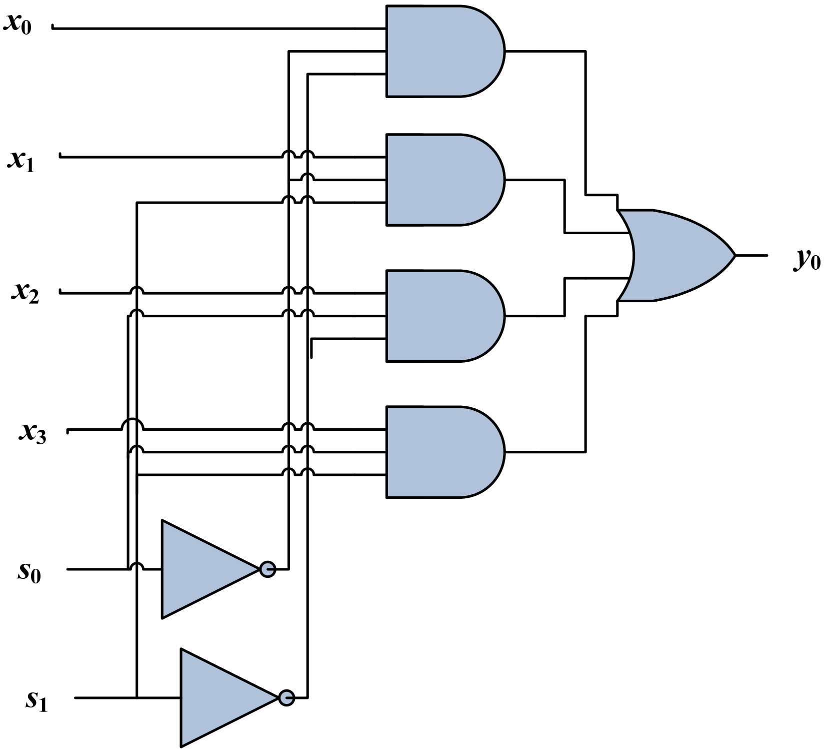
d) C17 (Benchmark Circuit)



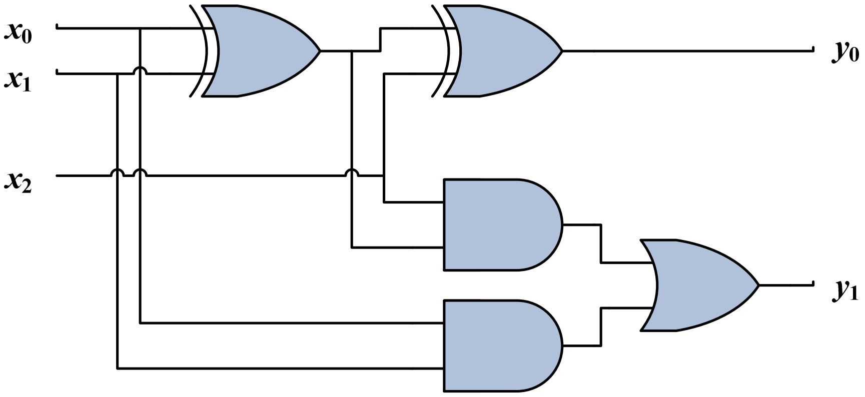
e) Dec 2x4 (2 to 4 Line Decoder)



f) Mux 4x1 (4 to 1 Line Multiplexer)



g) Full Adder



APPENDIX B

SIMULATIONS OF TEST CIRCUITS

In this appendix, the simulation results showing the difference in output waveforms of CMOS, MRF and Improved-MRF test circuits is shown. Before proceeding towards the waveforms, the significance of following points has to be understood.

- The test circuits used in Tables 4.3-4.5 have been simulated under the effect of noisy input signal shown in Fig 1.
- For multiple-input circuits, the signal waveform in Fig B1 is modified (by changing its time period) and applied to the inputs other than the one utilizing signal in Fig 1.
- For multiple-output circuits, the output port which has the longest path length from the corresponding input was considered for analysis.

The simulations for the circuits in Tables 4.3-4.5 (except for the Inverter and NAND gate) have been shown in Fig B2-B5.

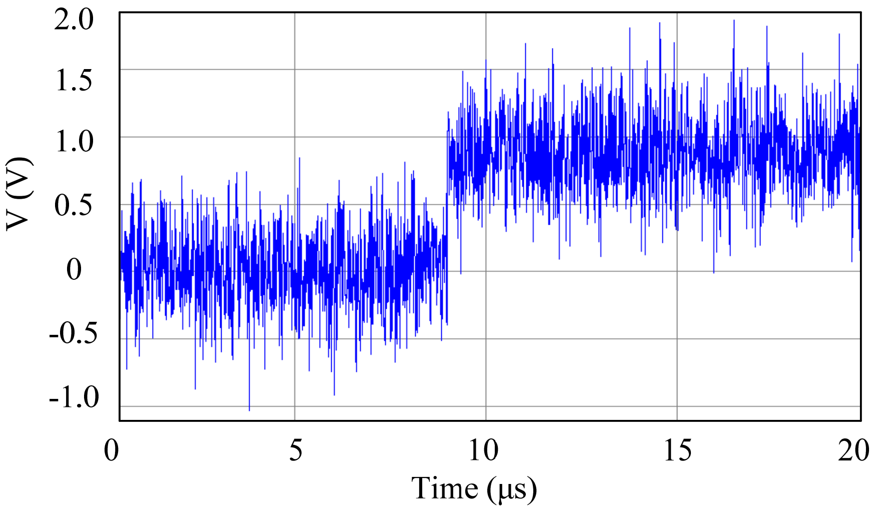
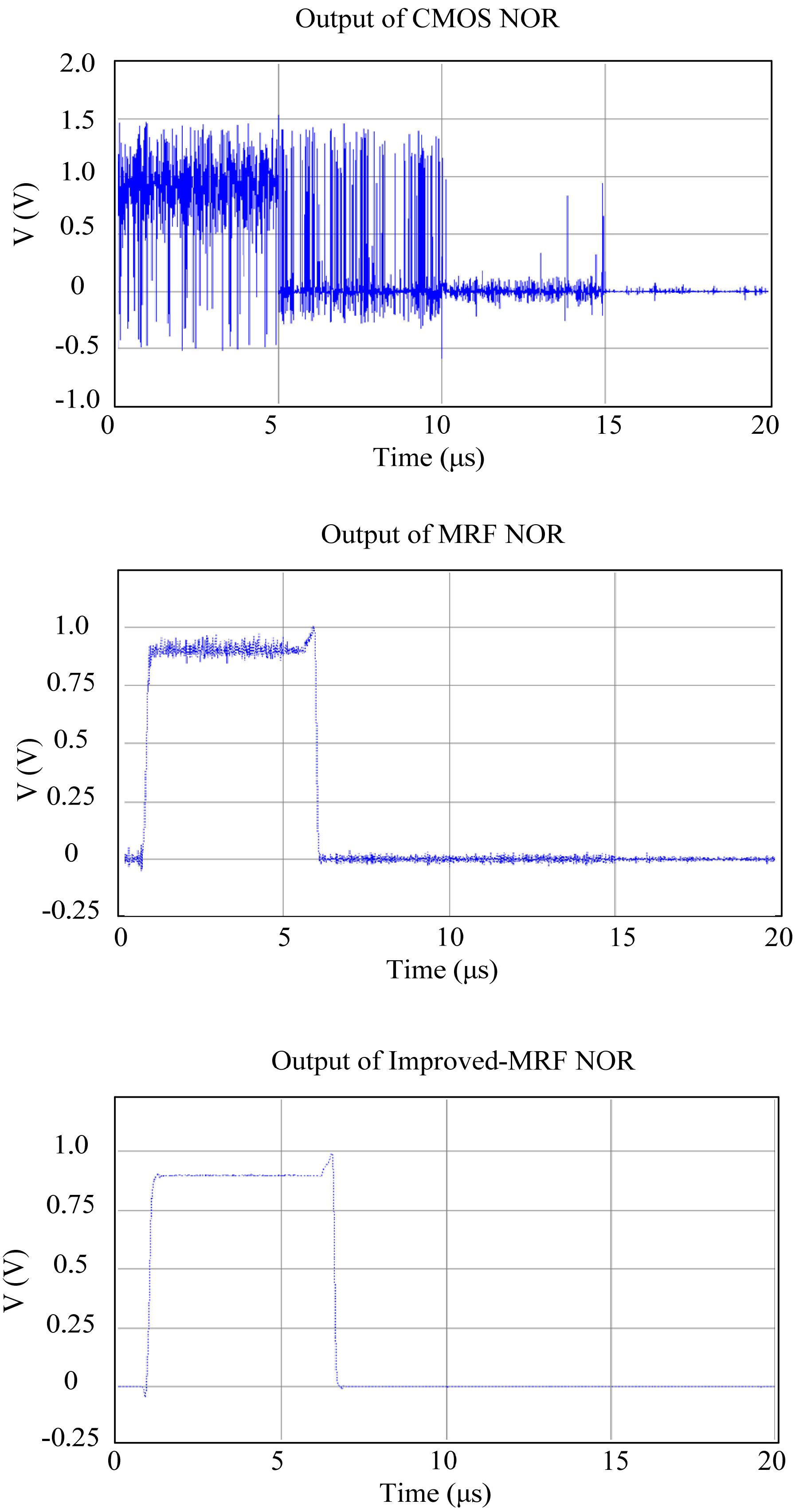


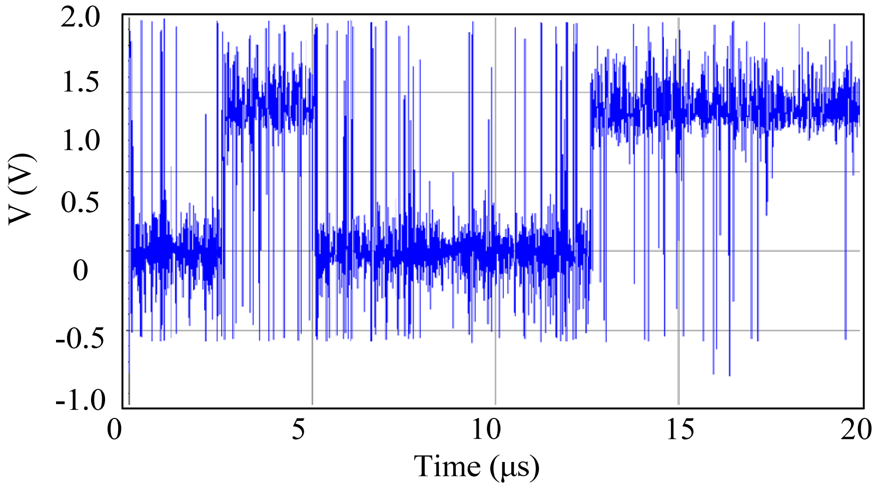
Fig B1: Noisy input signal used for 1-input gate

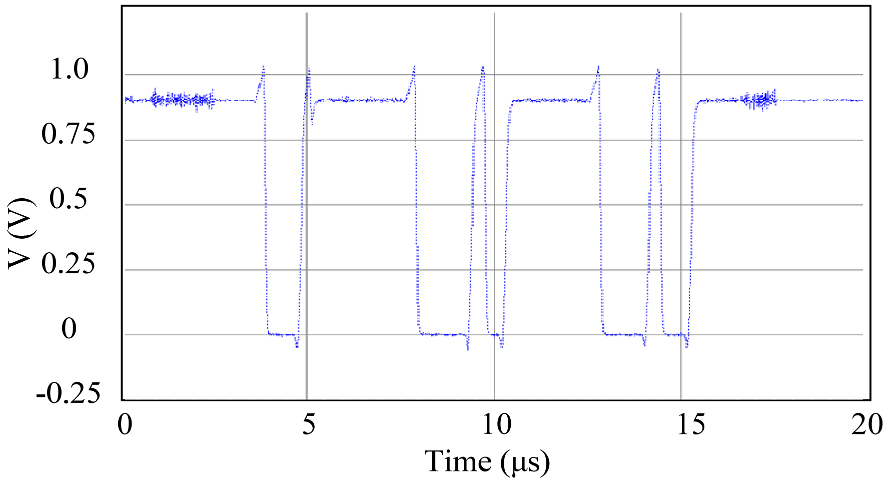

Output of CMOS NOR

Output of MRF NOR

Output of Improved-MRF NOR

Fig B2: Output waveforms of CMOS, MRF and Improved-MRF NOR


Output of CMOS C17


Output of MRF C17

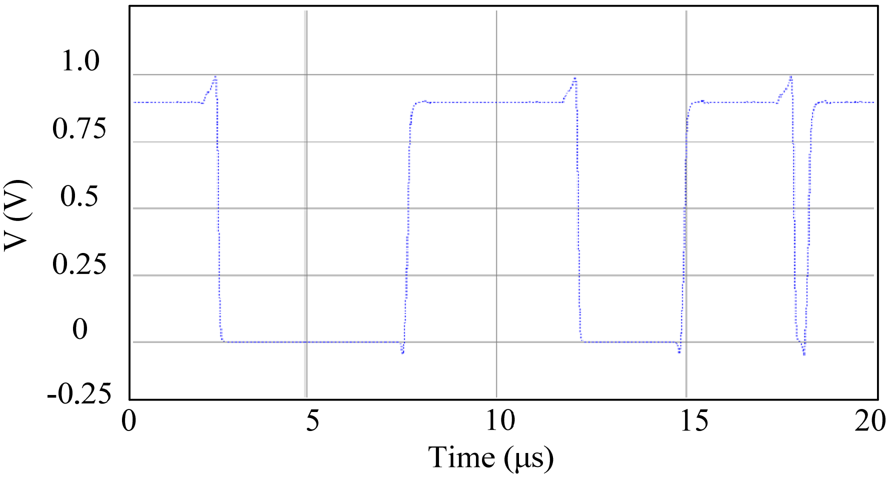
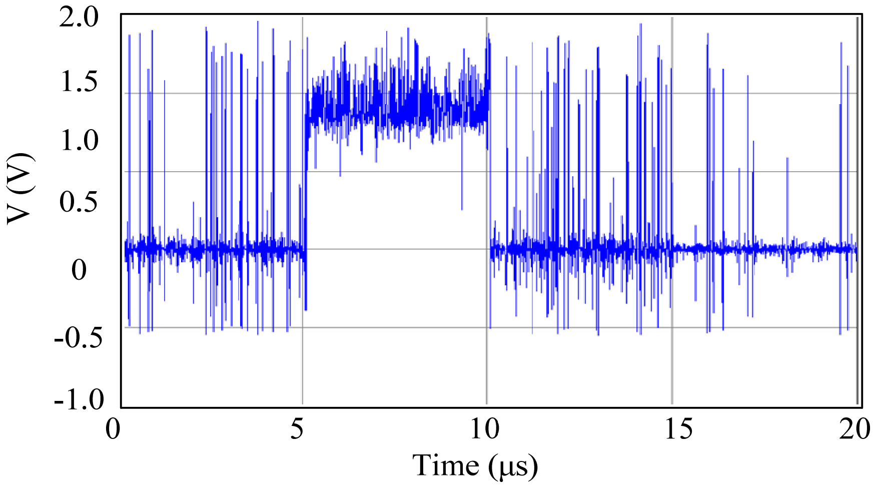
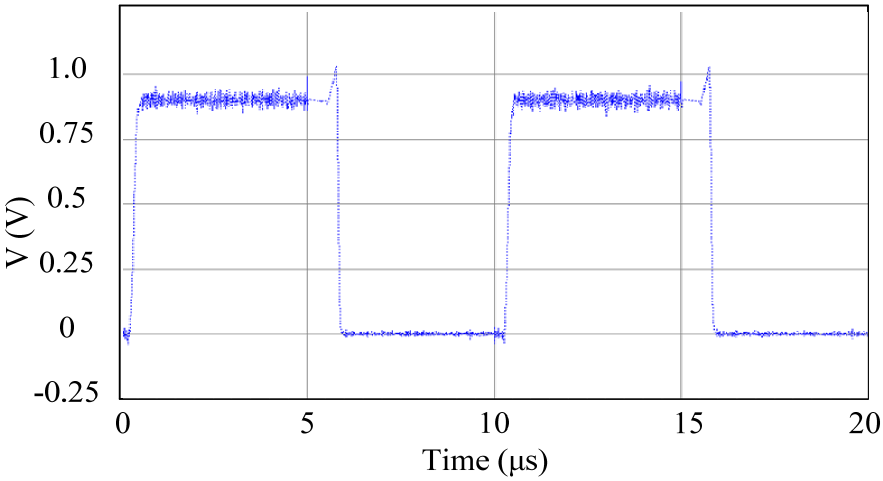

Output of Improved-MRF C17

Fig B3: Output waveforms of CMOS, MRF and Improved-MRF C17 Circuit

Output of CMOS Dec2x4



Output of MRF Dec2x4



Output of Improved-MRF Dec2x4

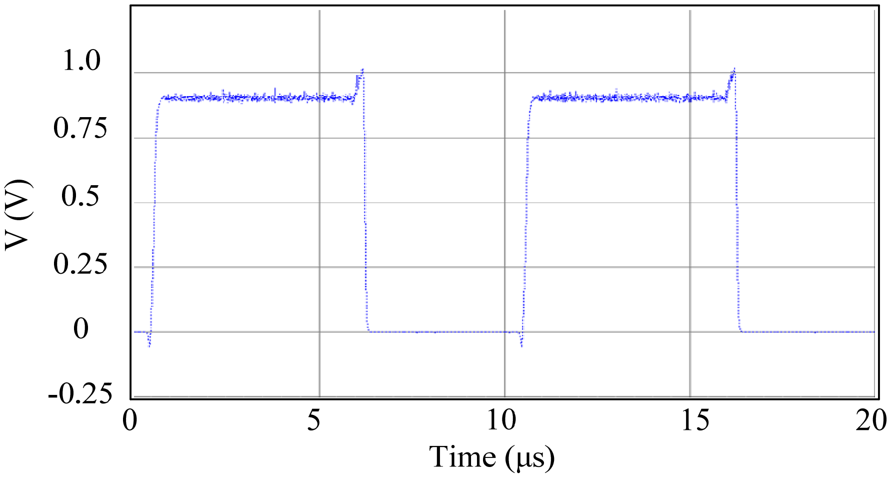


Fig B4: Output waveforms of CMOS, MRF and Improved-MRF Decoder (2x4)

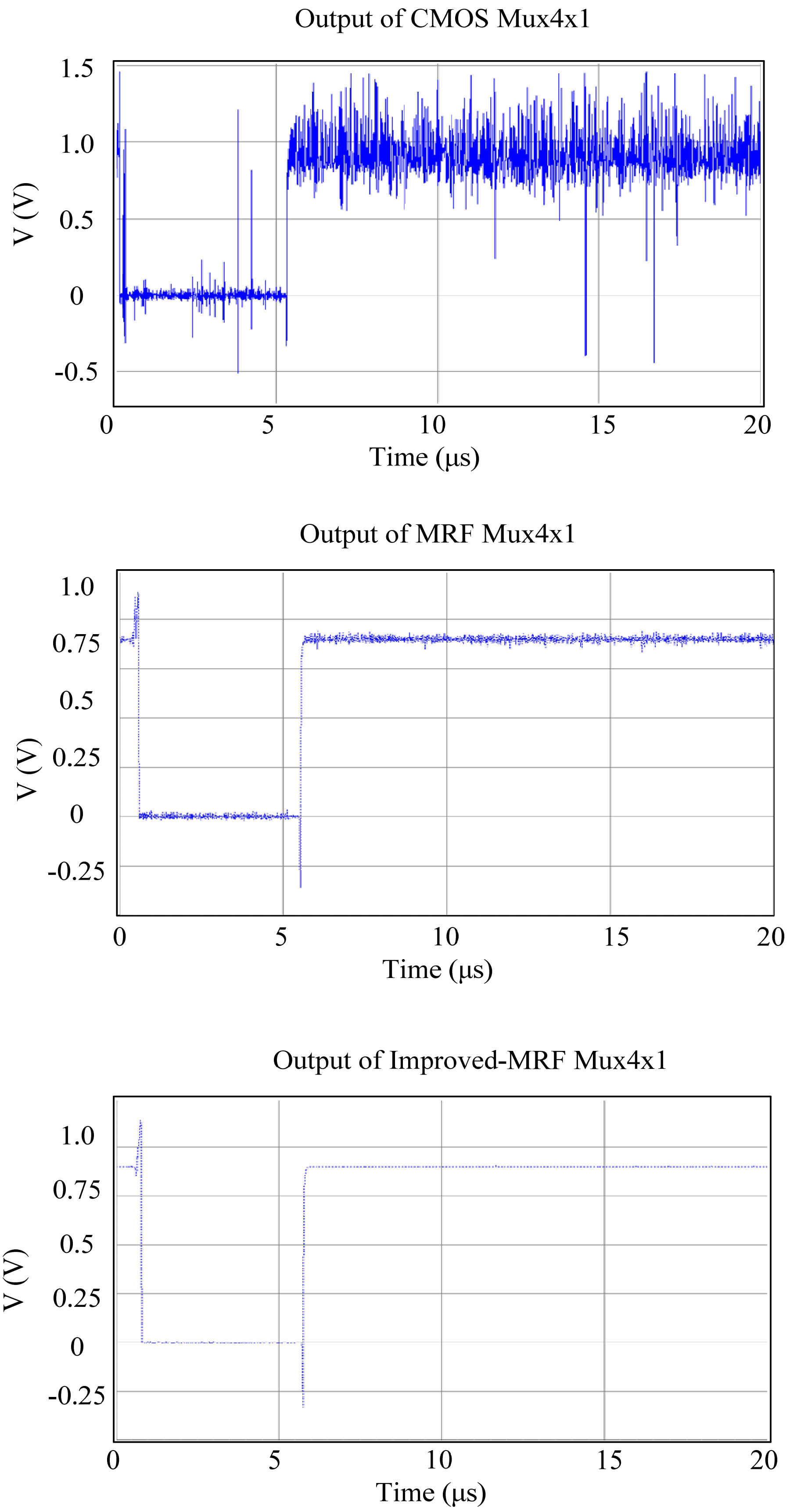
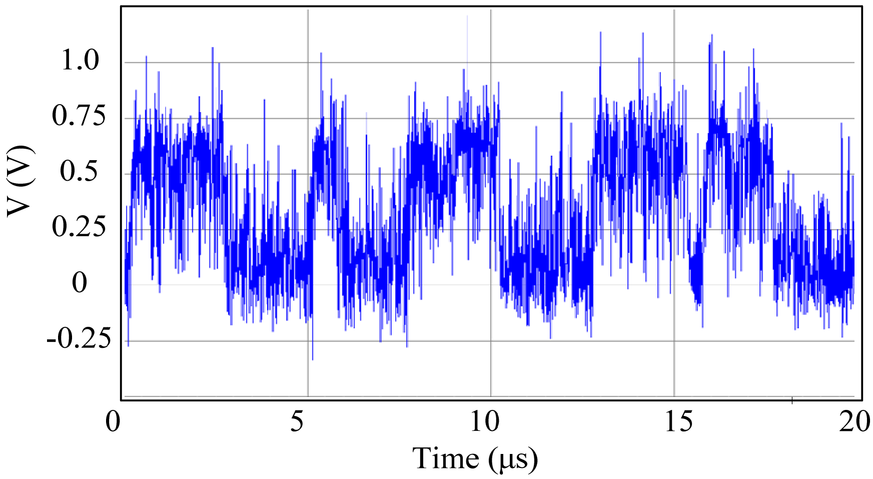
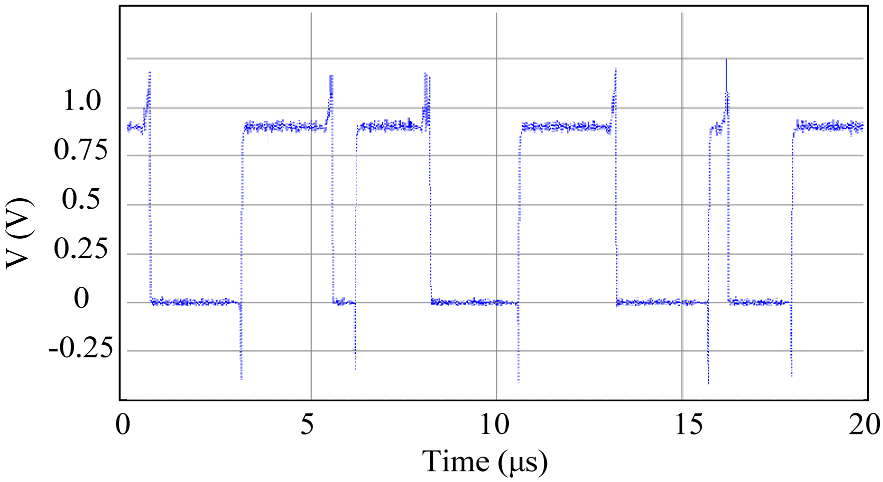


Fig B5: Output waveforms of CMOS, MRF and Improved-MRF Multiplexer (4x1)

Output of CMOS Full Adder



Output of MRF Full Adder



Output of Improved-MRF Full Adder

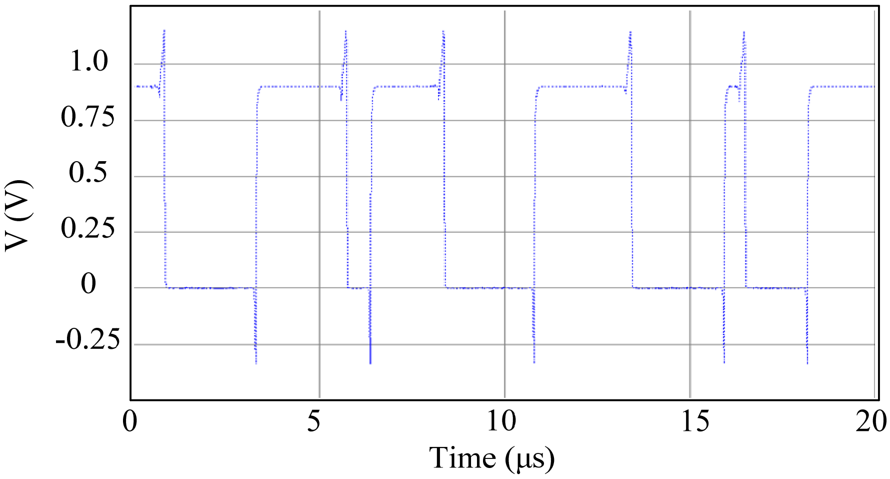


Fig B6: Output waveforms of CMOS, MRF and Improved-MRF Full Adder

APPENDIX C

PUBLICATIONS

1. Jahanzeb Anwer, Usman Khalid, Narinderjit Singh, Nor H. Hamid, Vijanth S. Asirvadam, “Joint and Marginal Probability Analyses of Markov Random Field Networks for Digital Logic Circuits,” in 3rd International Conference on Intelligent and Advanced Systems (ICIAS), Kuala Lampur, Malaysia, June 2010.

2. Jahanzeb Anwer, Usman Khalid, Narinderjit Singh, Nor H. Hamid, Vijanth S. Asirvadam, “Highly Noise-Tolerant Design of Digital Logic Gates using Markov Random Field Modelling,” in 2nd International Conference on Electronic Computer Technology (ICECT), Kuala Lampur, Malaysia, May 2010.

3. Jahanzeb Anwer, Ahmad Fayyaz, Muhammad M. Masud, Saleem F. Shaukat, Usman Khalid and Nor H. Hamid, “Fault-Tolerance and Noise Modelling in Nanoscale Circuit Design,” in 2010 International Symposium on Signals, Systems and Electronics (ISSSE), Nanjing, China, Septermber 2010.

4. Jahanzeb Anwer, Usman Khalid, Narinderjit Singh, Nor H. Hamid, Vijanth S. Asirvadam, “A Novel Error Detection Mechanism for Digital Circuits Using Markov Random Field Modelling,” in IEEE International Conference on Machine Learning and Computing (ICMLC 2011), Singapore, February 2011.