**DESIGN TECHNIQUES TO IMPROVE TIME DEPENDENT
DIELECTRIC BREAKDOWN BASED FAILURE
FOR CMOS CIRCUITS**

A THESIS PRESENTED TO THE
ELECTRICAL ENGINEERING DEPARTMENT FACULTY OF
CALIFORNIA POLYTECHNIC STATE UNIVERSITY, SAN LUIS OBISPO

IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
MASTER OF SCIENCE DEGREE IN ELECTRICAL ENGINEERING

BY
EMANUEL S. TAROG

JANUARY 2010

© 2010
Emanuel S. Tarog
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Design techniques to improve time dependent dielectric breakdown based failure for CMOS circuits

AUTHOR: Emanuel S Tarog

DATE SUBMITTED: January 2010

COMMITTEE CHAIR: John Oliver, Assistant Professor

COMMITTEE MEMBER: James Harris, Professor Emeritus

COMMITTEE MEMBER: Chris Lupo, Assistant Professor

ABSTRACT

Title: Design techniques to improve time dependent dielectric breakdown based failure for CMOS circuits

Author: Emanuel S. Tarog

This project investigates the failure of various CMOS circuits as a result of Time Dependent Dielectric Breakdown (TDDB) and explores design techniques to increase the mean time to failure (MTTF) of large-scale circuits. Time Dependent Dielectric Breakdown is a phenomenon where the oxide underneath the gate degrades as a result of the electric field in the material. Currently, there are few well documented design techniques that can increase lifetime, but with a tool chain I created called the MTTF Analyzing Program, or MAP, I was able to test circuits under various conditions in order to identify weak links, discover relationships, and reiterate on my design and see improvements and effects.

The tool chain calculates power consumption, performance, temperature, and MTTF for a 'real life' circuit. Electric VLSI, an Electronic Design Automation tool, outputs a Spice file that yields parasitic quantities and spatial dimensions. LTspice, a high performance Spice simulator, was used to calculate the voltage and current data. Finally, I created MAP to monitor the voltage, current, and dimension data and process that in conjunction with HotSpot, a thermal modeling tool, to calculate a MTTF for each MOSFET.

Analysis of the data from the software infrastructure showed that transistor sizing played a role in the MTTF. To maximize the MTTF of a transistor in a CMOS inverter, the activity of the pull-up transistor should be balanced with the transistor in the pull-down chain, ensuring the electric fields are balanced across both transistors. While it is impossible to completely balance an arbitrary CMOS circuit's activity for an arbitrary set of input signals, circuits can be intelligently skewed to help maximize the MTTF without increasing power consumption and without sacrificing circuit performance. Consequently, attaining a maximum MTTF does not come at a cost as it is possible to design a circuit with a high MTTF that performs better and uses less power than a circuit with low MTTF.

Keywords: Mean Time to Failure, Time Dependent Dielectric Breakdown, CMOS, circuit design, electric field

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Oliver, for his wisdom, guidance and extreme patience in working with me over the last 2 years both during this project and in the classroom. I am also grateful to Professor Harris and Professor Lupo for generously giving their time to be on my committee and the many teachers whom I've had the honor of studying under whose passion in teaching made electrical engineering actually enjoyable. I'd also like to thank all my friends who have helped me with finishing this thesis. Lastly, I would like to thank my wonderful parents. Without their love, patience and discipline, I would not be where I am today and I would not be the person I am today.

TABLE OF CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENTS.....	III
TABLE OF CONTENTS	IV
LIST OF FIGURES & TABLES.....	VII
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 BACKGROUND	3
2.1 TDDDB.....	3
2.2 Failure Mechanisms/TDDDB Effects.....	4
2.2.1 Time Dependent Dielectric Breakdown: Quantum Tunneling	4
2.2.2 Time Dependent Dielectric Breakdown: The E, 1/E, and the $\sqrt{E}$ Models ...	6
2.2.3 Why is this an issue?.....	9
2.2.4 Creation of a Mean-Time-to-Failure Model	9
CHAPTER 3 CIRCUIT LAYOUT, CREATION & CATALOG	14
3.1 Electric VLSI	14
3.1.1 Technology Choice	15
3.1.2 Circuit Layout, Terminology, and Design Rules	16
3.2 Circuit Choice (Basic/Complex/Sub-circuits)	17
3.2.1 CMOS Kogge-Stone	18
3.2.2 8x8 Register File	20
CHAPTER 4 SYSTEM IMPLEMENTATION.....	22
4.1 General Description	22
4.2 System Tools.....	28
4.2.1 Electric VLSI	28
4.2.2 LTSpice.....	29
4.2.3 LTSpice Utility	29
4.2.4 HotSpot	29
4.2.5 Monitoring Program - MAP.....	30
4.2.6 Simple Scripts	30
4.2.7 Excel Macros	30
4.2.8 The Tool Chain	30
4.3 MTTF Monitoring Program - MAP.....	31
4.3.1 Creation of Intermediate Files	32
4.3.2 Extracting Runtime Information	34
4.3.3 Creating a Catalog.....	34
4.3.4 Assigning MOSFETs to regions	36
4.3.5 Circuit Usage Monitoring	36
4.3.6 Creation of the Power Trace File	39
4.3.7 Calling HotSpot	39

4.3.8	Changing Metal Widths	39
4.3.9	Calculating MTTF	40
4.4	Additional Tools	40
4.4.1	Setting up Multiple Tests	40
4.4.2	Changing the Temperature.....	41
4.4.3	Converting the LTSpice Utility Data File.....	41
4.4.4	Calculating Average Power for Each Region	41
4.4.5	The Speed Tester.....	42
4.4.6	Gathering the Results	44
4.5	Verifying the Models - The Basic Transistor	45
4.6	Testing the Circuits	46
4.7	Limitations of the tool chain	47
CHAPTER 5 CIRCUIT EVALUATION		49
5.1	Introduction.....	49
5.2	The Inverter.....	50
5.2.1	The Affect of Various input and circuit properties	50
5.3	Complex Circuits	54
5.3.1	Adders	54
5.3.1.1	Ripple-Carry Adders	54
5.3.1.2	Kogge-Stone Adders	56
5.3.2	8 x 8-Bit Register File.....	59
5.4	Failures and Associated Performance Flaws	61
5.4.1	E-field on MTTF.....	61
5.4.1.1	Activity	61
5.4.1.2	Ideal E-fields	62
5.4.1.3	Frequency.....	64
5.4.1.4	Speed.....	66
5.4.1.5	Gate Size.....	68
5.4.1.6	Temperature.....	68
5.4.2	Temperature on MTTF.....	70
CHAPTER 6 IMPROVING DESIGN AND RESULTS		73
6.1	Possible circuit improvements	73
6.2	Example: Improving the MTTF of an XOR-gate	75
6.2.1	Lowering Parasitic Capacitance.....	76
6.2.2	Improving Circuit Performance	76
6.2.3	Results on Small Circuits.....	77
6.2.4	Effects on Power and Delay.....	78
6.3	Summary on Small Circuits	79
6.4	Results on Larger Circuits	83
6.4.1	Kogge-Stone Adders	83
6.4.2	Register File	87
6.5	Summary on Large Circuits	91
CHAPTER 7 CONCLUSIONS		92

Appendix A – Detailed Intermediate Files description for the tools chain	93
A.1 Manual creation of floorplans and relationship files	93
A.2 Layout Preferences and Creation of Spice Netlists	95
A.3 Raw Current/voltage Data from Spice	95
A.4 The Power Trace File.....	96
Appendix B – The Full Adder	97
Appendix C – CMOS Kogge-Stone.....	98
C.1. Final Design Details.....	98
Appendix D – Gate and Semi Complex Circuit Analysis	104
D.1 Gate Level	104
D.1.1 The NAND, NOR-gates, and XOR Gates	104
D.1.2 The AND- and OR-Gates.....	106
D.1.3 Summary	107
D.2 Semi-Complex Circuits.....	107
D.2.1 Multiplexors (MUX).....	107
D.2.2 Full Adders.....	108
D.2.3 Decoder	109
Appendix E –Improving the MTTF of a MUX and Full Adder	110
E.1 Improving a MUX.....	110
E.2 Improving a Full Adder	111
E.3 Maximum Improvement	113
Appendix F – List of Acronyms	117
Appendix G – List of Equations	118
BIBLIOGRAPHY	119

LIST OF FIGURES & TABLES

Figure 2.1.1: Relationship Between TDDb and Leakage Current [10].....	4
Figure 2.2.1: Relationship Between TDDb and the Electric Field [10].....	4
Figure 2.2.2: Various Quantum Tunneling Illustrations [8].....	5
Figure 2.2.3: TDDb vs. Dielectric Constant, as well as VLSI Generations [10]	8
Figure 2.2.4: MTTF vs. E-Field with Varying Temperature [17]	10
Figure 2.2.5: MTTF vs. Temperature for varying Electric Fields [15].....	10
Figure 2.2.6: Diagram of n-MOSFET elements.....	12
Figure 2.2.7: Dependence of TDDb of Intrinsic Oxide on Oxide Field for T=246°C and 342°C (left), and the Arrhenius Plots of TDDb for $V_o = -4.5V, -4.1.0V, -3.9V$ and $3.7 V$ (right).....	13
Figure 3.1.1: Two-input XOR-gate Layout.....	16
Figure 3.2.1: 16-Bit Kogge-Stone Parallel-Prefix Adder Layout with CMOS OPL Gates	18
Figure 3.2.2: Block diagram for 16-Bit Kogge-Stone prefix adder network.....	19
Figure 3.2.3: (a) Even-layer Tiled CMOS PPA circuit for inverted input and non-inverted outputs (b) Odd-layer Tiled CMOS PPA circuit for non-inverted inputs and inverted outputs.....	19
Figure 3.2.3: Kogge-Stone Critical Path for 8 Logic Stages for a 16-Bit Adder (not all gate inputs shown).....	20
Figure 3.3.1: 8x8 2-Read / 1-Write Register File.....	20
Figure 3.3.2: The layout of a 12-transistor SRAM cell.....	21
Figure 4.1.1: System Block Diagram Description.....	23
Figure 4.1.2: Circuit Creation	23
Figure 4.1.3: Test Automation Preparation.....	24
Figure 4.1.4: Circuit Data Acquisition	25
Figure 4.1.5: The Monitoring Program	25
Figure 4.1.6: Post Processing.....	26
Figure 4.1.7: Data Consolidation	26
Figure 4.1.8: Summary of the inputs, intermediate programs and scripts and output of the tool chain.....	27
Figure 4.3.1: 10mm x 5mm Rectangle Floorplan	32
Figure 4.3.2: Floorplan File of 10mm x 5mm Rectangle	32
Figure 4.3.3: Power Consumption of 10mm x 5mm Rectangle	33
Figure 4.3.4: A code representation of sub-circuits grouped into blocks using Electric coordinates	36
Figure 4.3.5: A Visual Representation of the Data as it Appears in the LTSpice file	38
Figure 4.3.6: The Left Riemann Sum of the data.....	38
Figure 4.4.1: Manually Finding the Delay	42
Figure 4.4.2: Visual representation of a continuous waveform against the discrete representation of LTSpice	44
Figure 4.5.1: MTTF vs Electric Field Models created from Real Data.....	46
Figure 5.2.1: MTTF vs. Input Period Plot Example for Different Rail Voltages	51
Figure 5.2.2: An MTTF vs. Power Plot Example.....	52
Figure 5.2.3: Example of an MTTF vs. Input Period Plot.....	53
Figure 5.3.1: Comparison of 8-Bit, 16-Bit, and 32-Bit Adders.....	56
Figure 5.3.2: Critical Path of the KS Adder.....	57
Figure 5.3.3: MTTF data for the 16bit KS Adder run at different input frequencies, rail voltages, and temperatures.....	58
Figure 5.3.4: MTTF data for the Register File run at different input frequencies, rail voltages, and temperatures.....	60

Figure 5.4.1: Input Frequency vs. Normalized MTTF when Temperature = 373K.....	63
Figure 5.4.2: Input Frequency vs. Normalized MTTF when E_f = The Nominal Value.....	64
Figure 5.4.3: The Waveform of the Gates of Transistor P7 in Figure XOR, at 400MHz (left), and 25MHz (right).	66
Figure 5.4.4: The Waveform of the Gates of Transistor P7 in Figure 5.3.2 for Two Different Gate Sizes.	67
Figure 5.4.4: The gate voltage of the weak link of an 8-Bit RCA.....	69
Figure 5.4.5: The MTTF of the KS Adder at Ambient Temperature, with Self-Heating	71
Figure 5.4.6: The MTTF for a PMOS in a Kogge-Stone adder for a constant E-Field and a varying E-Field.	72
Figure 6.1.1 The layout of an XOR Gate and the transistors relevant to the following sections.....	76
Figure 6.5.1: The MTTF of the XOR-gate for Various Changes	77
Figure 6.5.2: The Effect of Changes with Respect to the Frequency of the Inputs	78
Figure 6.6.1: The Effects of the Changes on Delay	79
Figure 6.6.2: The Effects of the Changes on Power.....	79
Figure 6.7.1: The modifications goal	80
Figure 6.7.2: Surface plot for the MTTF of the Multiplexor when it is unmodified (a) and when it is improved (b)	83
Figure 6.8.1.: Modification results for the AND Gate of the Kogge-Stone Adder.....	84
Figure 6.8.2: Modification results for the XOR Gate of the Kogge-Stone Adder.....	85
Figure 6.8.3: Modification for the PPA Even Subcircuit of the Kogge-Stone Adder	85
Figure 6.8.4: Modification results for the PPA Odd Subcircuit of the Kogge-Stone Adder.....	86
Figure 6.8.5: The results of modifications on the various sub-circuits of the Kogge-Stone Adder.....	87
Figure A.1: Input File for Perl Script Format.....	93
Figure A.2: Example of the Floor and Relationship.....	94
Figure A.3: Sub-circuits as they are grouped into blocks using Electric coordinates.....	94
Figure C.1: Block diagram for 16-Bit Kogge-Stone prefix adder network	98
Figure C.2: Even-layer Tiled CMOS PPA circuit for inverted input (left), and non-inverted outputs (right).....	99
Figure C.3: Even-layer Tiled CMOS-OPL PPA circuit layout cell.....	100
Figure C.4: Odd-layer Tiled CMOS PPA circuit for non-inverted inputs (left), and inverted outputs (right).....	100
Figure C.5: Odd-layer Tiled CMOS-OPL PPA circuit cell.....	101
Figure C.6: CMOS OPL layout circuits for AND (left) and XOR (right) gates used for the Pi and Gi signals	102
Figure C.7: Kogge-Stone Critical Path for 8 Logic Stages for a 16-Bit Adder (not all gate inputs shown)	102
Figure C.8: 16-Bit Kogge-Stone Parallel-Prefix Adder Layout with CMOS OPL Gates.....	103
Figure D.1.1: The Circuit for a NAND-gate (left), and a NOR-gate (right).....	105
Figure D.1.2: Transistor Network that Creates an Exclusive-OR function and the layout with the relevant transistor labeled	106
Figure D.1.3: The Weak Spots of an AND-gate and OR-gate with Equal Inputs.	107
Figure D.2.1: The circuit diagram of a CMOS MUX and a layout schematic.	108
Figure D.2.2: Layout and Circuit Diagram of the Adder.	109
Figure D.2.3: Layout of the 3-to-8 Decoder.....	109
Figure E1.1 The layout of a MUX and the relevant transistors.....	110
Figure E.1.2: The MTTF of the MUX for Various Changes and Input Frequencies.....	110
Figure E.1.3: Comparison of the Power Consumption Between the Various Changes and Input Frequencies.....	111
Figure E.2.1: MTTF for different input frequencies for various modifications.....	112
Figure E.2.2: Power Consumption of the Circuit for Various Changes and Input Frequencies	113
Figure E.3.1: MTTF for the Full Adder Improvement.	114

Figure E.3.2: MTTF for the Multiplexor Improvements.....	114
Figure E.3.4: Fixed vs. Power for the Full Adder	115
Figure E.3.5: Fixed vs. Delay for the Full Adder.....	115
Figure E.3.6: Fixes vs. Power for the MUX.....	116
Figure E.3.6: Fixes vs. Delay for the MUX	116

CHAPTER 1 INTRODUCTION

With the scaling down process of microcircuits, the effects of Time Dependent Dielectric Breakdown are becoming increasingly severe. This paper will address Time Dependent Dielectric Breakdown (TDDB), the phenomena where the oxide underneath the gate material of a MOSFET degrades resulting in a short circuit. As technology scales down, the oxide becomes thinner and more fragile allowing the affects of TDDB to become more severe. In order to combat TDDB, engineers and scientists have used different dielectric materials and have also designed circuits for lower power consumption and a lower rail voltage. However, there have been few to no circuit design solutions.

While functionality, power consumption, and performance are major factors in designing a circuit, lifetime is not. Currently, there is no tool that can assist designers in locating failure sites like PSpice can assist designers in verifying the functionality of a circuit. This thesis presents a tool chain with a custom program called MAP that I created for the sole purpose of calculating mean time to failure (MTTF). Ultimately, the tool was used to discover if there is indeed a circuit design solution to increase MTTF as a result of TDDB. For the purpose of this project, the tool chain specifically monitors TDDB related circuit activity, but the open source nature of the tool chain can allow for different failure types such as electromigration. In addition to calculating MTTF for each MOSFET for a given circuit input, MAP calculates power consumption, self heating temperature, and performance, which is useful in comparing different designs of the same circuit.

This paper is divided into 5 main chapters. The following chapter addresses the problem of TDDDB and the physics of why it occurs. Chapter 3 discusses the circuit selection and the importance of Electric VLSI. Chapter 4 describes the tool chain and MAP and describes the inner workings of the tool. Chapter 5 shows the results of the circuit catalog chosen for this project and addresses circuit characteristics and their relationship with MTTF. Finally, Chapter 6 explores different design techniques to increase MTTF and presents the results.

CHAPTER 2 BACKGROUND

Chapter 2 is an overview of Time Dependent Dielectric Breakdown, and the causes and issues associated with it. Section 2.1 gives a brief description of TDDB and what happens to a MOSFET when it fails. Section 2.2 details the different quantum tunneling phenomena, which is known to be the root cause of TDDB, and the mathematical models that can predict the mean time to failure. Section 2.3 gives a brief description of the importance of TDDB when dealing with digital circuits. Finally, Section 2.4 summarizes the method used to simplify those models for the scope of this paper.

2.1 TDDB

Time Dependent Dielectric Breakdown is a phenomenon where the oxide underneath the gate degrades. As the name implies, it is the breakdown of a dielectric over time. There are other ways a dielectric can breakdown but in a digital system, the only variables are operating frequency, voltage supply, MOSFET characteristics (such as gate area or dielectric material), temperature, and time.

As the gate-oxide is scaled down, breakdown of the oxide and oxide reliability becomes more of a concern. Higher fields in the oxide increase the tunneling of carriers from the channel into the oxide. These carriers slowly degrade the quality of the oxide and over time, leads to failure of the oxide [5]. Once a dielectric breaks down, current is able to flow more easily through the gate into the drain/source of a P/NMOSFET completely destroying functionality. Evidence of TDDB are changes in the threshold voltages and the drain currents as well as a great increase in current through the dielectric [6] and ultimately the gate as shown in Figure 2.1.1.

Figure 2.1.1: Relationship Between TDDB and Leakage Current [10]

2.2 FAILURE MECHANISMS/TDDB EFFECTS

2.2.1 TIME DEPENDENT DIELECTRIC BREAKDOWN: QUANTUM TUNNELING

There are many hypotheses for why TDDB occurs. Many models describe what occurs in the dielectric material over time and each model consequently has a mathematical model that can predict the expected failure of a device. There has been much speculation for the last 50 years as to which model correctly predicts the failure time. But there is general consensus that the electric field through the dielectric material is the direct cause of TDDB. This relationship is shown in Figure 2.2.1.

Figure 2.2.1: Relationship Between TDDB and the Electric Field [10]

The simple explanation is that the electric field breaks down the oxide, but electric fields could be the cause of more specific phenomena, such as band-to-band impact ionization, hole trapping near the injecting interface, and electron trapping [6], but it is accepted that it is caused by charge that remains in the oxide [7]. Ideally charge should not pass through the oxide, but thinner oxides and stronger electric fields make this possible. Strong electric fields lead to quantum tunneling effects such as Fowler-Nordeim Tunneling, Direct Tunneling, and Trap Assisted Tunneling [8].

Figure 2.2.2: Various Quantum Tunneling Illustrations [8]

Fowler-Nordeim Tunneling is a quantum mechanical process where electrons can penetrate through the oxide barrier into the conduction band of the oxide [8]. Tunneling is exponentially dependent on voltage. Direct Tunneling is another process where electrons penetrate through the gate directly to the channel [8]. In this process, tunneling increases exponentially with the decrease of oxide thickness. Trap Assisted Tunneling is again another important process where electrons go through the oxide into traps, which are empty bonding sites within the oxide, and then proceed into the silicon.

There are at least five (5) oxide breakdown models currently used by reliability engineers: the Bandgap Ionization Model, the Classic Anode Hole Injection (1/E) Model, the Hydrogen Release Model, the Thermochemical (E) Model [8], and a combination of the Charge-to-Breakdown (QBD) Hypothesis and the Poole-Frenkel (PF) Conduction ($\sqrt{E}$) Mechanism [9]. The important models to consider are the Classic Anode Hole Injection Model, the Thermochemical Model, and the Poole-Frenkel Conduction Mechanism because they have commonly used mathematical models to predict time failure and have been thoroughly tested to provide some accurate results.

2.2.2 TIME DEPENDENT DIELECTRIC BREAKDOWN: THE E, 1/E, AND THE $\sqrt{E}$ MODELS

According to the Thermochemical (E) Model, the electric field applied to the gate, creates traps within the oxide. This process is further aggravated by temperature. The mathematical equation for the mean time to failure, τ , to model this phenomenon is:

$$\tau = A \exp\left(\frac{E_a}{kT}\right) \exp(-\gamma E_f) \quad (2.4.1),$$

where A is a constant, E_a is the activation energy, k is Boltzmann's constant, T is temperature, γ is the field acceleration factor, and E_f is the electric field strength in the oxide [8] [10]. It is usually referred to as the E model because of its E dependence as can be seen in the second exponential.

Classic Anode Hole injection (1/E) Model predicts that some electrons entering the anode of a transistor will have enough energy to create a “hot” hole which can tunnel back into the oxide. This can happen at either high or low energies, but it can be observed most clearly with electrons with very high energy [8]. The mathematical equation for the mean time to failure to model this phenomenon is:

$$\tau = \frac{A \exp(\frac{E_a}{kT}) \exp(-\frac{\gamma}{E_f})}{E_f^2} \quad (2.4.2),$$

where again, A, E_a , T, and γ is the same as in equation (2.4.1) above. As with the E model, this model is called the 1/E model because of its 1/E dependence in the second exponential [11].

The charge to breakdown hypothesis and the Poole-Frenkel Leakage Mechanism ($\sqrt{E}$) are the basis for the last major model. The charge to breakdown hypothesis states that once a critical charge has been forced through the dielectric, the dielectric will break down. Also, the Poole-Frenkel Leakage Mechanism predicts that leakage current interacts with the dielectric which leads to degradation [9]. The corresponding equation is:

$$\tau = \frac{Q_{BD}}{E_f} \exp\left(\frac{q[\Phi_B - \sqrt{\frac{qE_f}{\pi\epsilon_0\epsilon_\infty}}]}{kT}\right) \quad (2.4.3)$$

where E_f is the electric field like the previous two models, and Q_{BD} is the critical charge or the breakdown charge, Φ_B is the trap depth, q is the elementary charge, ϵ_0 is permeability in a vacuum, and ϵ_∞ is permeability in the dielectric [9]. Again, the square root E dependence can be seen.

To combat the effects of TDDB, different methods were employed. Thicker oxides have been used, but as technology continues to scale down, as does the thickness of the dielectrics and with each new VSLI generation, TDDB becomes a greater concern.

Figure 2.2.3: TDDB vs. Dielectric Constant, as well as VLSI Generations [10]

There are numerous papers published about different chemical compounds that could be used as a replacement for Silicon dioxide and some show promising results [5]. Such alternate oxides are called high-k dielectrics. These oxides have a greater dielectric constant so that the same gate capacitance can be obtained with a thinner oxide. The oxides tested are Al_2O_3 , ZrO_2 , and TiO_2 [5]. A graph showing different compounds and different oxide thicknesses and the effect on MTTF is shown in Figure 2.2.3.

Another attempt is lowering the power supply voltage which serves two purposes. In addition to reducing the power consumption of a circuit, it has been proved that lowering V_{dd} of a circuit can increase its lifespan [12]. Gate area also has an effect on TDDB. It has been proven mathematically as well as through experimentation, and as expected, pulse stressed circuits lasted longer than DC stressed circuits [11].

2.2.3 WHY IS THIS AN ISSUE?

When the oxide of a transistor breaks down, as discussed in the previous section, current is able to flow from the gate to the drain or source depending on the transistor type. The effect is a short from the gate to the source or drain completely destroying transistor functionality and potentially destroying the functionality of the larger circuit in which it resides. This also adds to leakage current which will consume more power, raise the temperature, and ultimately shorten the mean time to failure.

2.2.4 CREATION OF A MEAN-TIME-TO-FAILURE MODEL

Because circuits were not fabricated for this project, the equations for the E-Model, 1/E-Model, and the $\sqrt{E}$ Model can only be used as a guideline as the constants and variables cannot be determined experimentally. Equations were derived from another papers' data [17]. In order to accomplish this, data points were taken from the mean time to failure graphs and an exponential equation was extrapolated using Microsoft Excel. This approach is applicable to today's research since there is still much debate over which model is the correct one. By forming an equation from the collective data, the debate is bypassed.

A major drawback to this approach, however, is that there is not a purely mathematical relationship between failure, temperature and electric field. In Figure 2.2.4, the data represents that relationship at electric field time to breakdown, and each line represents that relationship at different temperatures. The lines are not uniformly distributed for each temperature; each line has its own unique slope and intercept; that is, they do not have one point where they all intersect. This trend not only applies to this one graph but also applies to many other papers as well [15] [17]. Another drawback to this approach is that models and equations derived from this data does not account for the

phenomena that cannot be modeled. An example of this is shown in Figure 2.2.5. The paper explains that the right most data to the right curves as a result of Joule heating [17].

Figure 2.2.4: MTTF vs. E -Field with Varying Temperature [17]

Figure 2.2.5: MTTF vs. Temperature for varying Electric Fields [15]

The data shown in Figures 2.2.4 and 2.2.5 are the standard for presenting MTTF data. MTTF data is typically shown in the form of a graph with temperature or electric field as the input to a MOSFET versus MTTF [10] [15] [17]. Normally, the mean time to failure is calculated as a function of temperature and electric field. The Y-axis is displayed as a logarithmic scale because MTTF is exponentially proportional to temperature as well as the electric field.

As mentioned before, Figure 2.2.4 contains data for different plots under varying temperatures. In order to create a model, the data collected must obtain both an electric field component and a temperature component. The first step was to obtain the equations of all the lines. Those equations were multiplied by a function of temperature thus creating the models for a transistor at a specific temperature.

$$V_{ox} = V_g - V_p - V_s - V_{FlatBand} [19] \quad (2.6.1)$$

Another issue is translating voltage to an electric field. The equation for the electric field inside the dielectric is the oxide voltage V_g divided by the oxide thickness, which is exactly the same as the electric field equation for a capacitor – the electric field equals the voltage divided by the distance between the two conductors. The voltage inside the oxide is given by equation (2.6.1) above.

Figure 2.2.6: Diagram of n-MOSFET elements

The monitoring program calculates the average voltage at the gate of each MOSFET and designates that as V_g in equation (2.6.1). The only issue with this is that LTSpice does not provide data for V_p . In order to account for this, data from another paper was used. *Field and Temperature Dependence of TDDDB of Ultrathin Gate Oxide* contains two graphs of great importance; see Figure 2.2.7. One graph shows MTTF as a function of temperature, and the other shows MTTF as a function of electric field. The first graph contains multiple data points for different gate voltages and the second graph contains multiple data points for different temperatures [18].

Fig. 1. Dependence of TDDB of intrinsic oxide on oxide field for $T = 246$ and 342 °C.

Fig. 3. The Arrhenius plots of TDDB for $V_g = -4.5, -4.1, -3.9$, and 3.7 V.

Figure 2.2.7: Dependence of TDDB of Intrinsic Oxide on Oxide Field for $T=246^{\circ}\text{C}$ and 342°C (left), and the Arrhenius Plots of TDDB for $V_o = -4.5\text{V}$, $-4.1.0\text{V}$, -3.9V and 3.7 V (right).

By utilizing these two graphs, the electric field can be represented as a function of gate voltage. This was achieved by using the two graphs to create a data point with electric field as a function of gate voltage. Once the data points were put into Excel, a linear equation was interpolated. The y-intercept was set to $y=0$, because the original equation gave a negative electric field with no gate voltage, and the relationship was verified in the technical paper. The final step was to scale the electric field based on the oxide thickness. The oxide thickness for the paper was $39\text{\AA}$, and the present technology that will be used for this project is $14\text{\AA}$. The program developed for this project can take the oxide thickness as a parameter to calculate MTTF. The model created has the same form as in equation (2.5.2). The validity of the model created is discussed in Chapter 4.

$$\tau = a * e^{b * E_f} * e^{c * T} \quad (2.5.2)$$

CHAPTER 3 CIRCUIT LAYOUT, CREATION & CATALOG

The following sections describe the first major piece of the system. This section will discuss the choice of circuits used, the technology used to create them, layout preferences, and the generation of Spice Netlists. A variety of circuits ranging from basic individual gates to complex binary adders were tested. Section 3.1 describes the tool used to create the circuits, Electric VLSI, and the decision behind choosing the various options the tool has to offer as well as design decisions in the different circuits created for testing. In addition, this section also gives a brief overview on some basic information, such as layout terminology and concepts. Section 3.2 addresses the two most complicated circuits created and analyzed for this project: a 16-bit Kogge-Stone Adder and a 8x8 two-read one-write Register File.

3.1 ELECTRIC VLSI

ElectricTM VLSI Design System is an open-source Electronic Design Automation (EDA) System that can handle many forms of circuit designs which include: Custom IC Layout, Schematic Capture (both digital and analog), Textual Languages such as VHDL and Verilog. This program was useful for this project because of its IC layout capabilities, its cost (negligible) and the fact that it is open source. Also, it is possible to add capabilities to Electric should the need arise. This software designs MOS integrated circuits, bipolar integrated circuits, printed-circuit-boards (PCBs), or any type of circuit chosen. It has many editing styles that include: layout, schematics, artwork, and architectural specifications. Also, this software has a large set of useful tools, such as, design-rule-checkers (DRCs), simulators, routers, and layout generators. Not only does

Electric interface with the most popular CAD specifications (EDIF, LEF/DEF, VHDL, etc), but it also provides an excellent layout-constraint system. This enables a top-down design approach by enforcing consistency of connections, and it is also an efficient and effective way to create and layout a large number of different circuits without having to fabricate a single one [21]. This chapter will describe the choice of circuits and the choice of layout technology used.

After circuit creation and layout, a method is needed to generate Spice netlists for each of the circuits designed. These netlists contain valuable information that the tool chain needs in order to function properly; the netlist contains MOSFET names, connections, nodes, and parasitic capacitances and resistances. Electric can generate decks for many other simulators (Verilog, Spice, etc). Spice decks can be written for all designs, and then any Spice simulator can be used to handle the netlists.

3.1.1 TECHNOLOGY CHOICE

Electric VLSI has almost any choice of design technology available. To keep up with today's fast paced technology along with the scaling down process of digital electronics, all circuits are designed to mimic the 0.05 μ m MOSIS (Metal Oxide Semiconductor Implementation Service) CMOS process technology. Electric represents all distances in dimensionless units. The scale choice of 0.05 μ m (50nm) process technology, a transistor which size is 2x3 is actually 100nm by 150nm (or 0.1 x 0.15 microns). The MOSIS CMOS technology describes a scalable CMOS process that is fabricated by the MOSIS project of the University of Southern California [21].

3.1.2 CIRCUIT LAYOUT, TERMINOLOGY, AND DESIGN RULES

Figure 3.1.1 shows a layout of a two-input XOR-gate. The layouts can be designed in a variety of ways, but practicality is a top priority. Note the P and N-MOSFET transistors circles and the silicon that connects any of the gates between them all. The top rail is our power (V_{DD}) and the bottom is ground (GND). The blue metal represents Metal Layer 1 and the purple metal represents Metal Layer 2. The various small “squares” are contacts, used for connecting different metal types or materials together. The inputs A and B, and the output Y appear as contacts. The light green regions show what’s known as “diffusion,” or the source and drain for each of the MOSFETs.

Figure 3.1.1: Two-input XOR-gate Layout

When designing a layout, transistors, contacts, metals, and other relevant components should be kept as close together as possible to minimize space and parasitic capacitance. The design rules specified by the layout preferences created from the beginning must be closely followed to minimize the amount of parasitic capacitance. It is also best to use as few metal layers as possible because of their tendency to add parasitic capacitance and resistance. Parasitic capacitance, in electric circuits, is unavoidable and usually undesirable. Capacitance exists between the parts of an electric component or circuit simply because of their proximity to each other. Parasitic capacitance can also exist between closely spaced conductors, such as metal lines, or PCB traces. The parasitic resistance of a metal or polysilicon line can also have a profound influence on the signal propagation delay over that line. The resistance of a line depends on the type of material used (Polysilicon, Aluminum, Gold, Copper, etc), the dimensions of the line and finally, the number and locations of the contacts on that line [20].

3.2 *CIRCUIT CHOICE (BASIC/COMPLEX/SUB-CIRCUITS)*

This project explores various circuits, each of varying complexity. Simple circuits include most two-input gates. MUXs and full adder blocks were tested as well. After creating a variety of smaller sub-circuits, the circuits were used to generate larger more complex designs. This thesis explores an 8-Bit, 16-Bit, and 32-Bit Ripple Carry Adder (RCA), a 16-Bit Kogge-Stone (KS) adder, and an 8x8 two-write, one-read register file. Appendix B briefly explains the design for the full adder tested for this project. Sections 3.2.1 and 3.2.2 explore the design of the KS adder as well as the register file used for this project since it will be useful in understanding the failure behavior of these circuits.

3.2.1 CMOS KOGGE-STONE

Figure 3.2.1: 16-Bit Kogge-Stone Parallel-Prefix Adder Layout with CMOS OPL Gates

The standard Kogge-Stone adder design for 16-Bits is shown in Figure 3.2.1. The basic tiling block sub-circuit of the KS adder is a parallel-prefix adder cell (PPA). The design uses two (2) types of tiles: an even tile for even rows (with row numbering starting at zero (0)), and an odd tile for odd rows. Each tile compresses two (2) prior propagate and generate signal sets into a single propagate and generate signal set for the column. The main sub-circuits that make up the KS adder are XOR- and XNOR- gates, AND-gates, and propagate and generate circuits and are shown in Figure 3.2.3 for even-layered and odd-layered tiles. The final architecture selected for the final design is OPL Kogge-Stone Radix-2 Prefix Adder. The complete 16-Bit Kogge-Stone Parallel-Prefix Adder layout is shown in Figure 3.2.2.

Figure 3.2.2: Block diagram for 16-Bit Kogge-Stone prefix adder network

Figure 3.2.3: (a) Even-layer Tiled CMOS PPA circuit for inverted input and non-inverted outputs (b) Odd-layer Tiled CMOS PPA circuit for non-inverted inputs and inverted outputs

The architecture selected for the final design is Output Prediction Logic (OPL) Kogge-Stone Radix-2 Prefix Adder. This adder was chosen because the PPA sub-circuits that were already developed and tested would properly evaluate the carry-in values for

each bit in a Kogge-Stone prefix tree. The critical path for the Kogge-Stone Adder is shown in Figure 3.2.4, which consists of eight (8) logic stages for the 16-Bit adder. This bit of information will become important to consider in Chapter 5. For more information about the creation and design of this KS adder, refer to Appendix C.

Figure 3.2.4: Kogge-Stone Critical Path for 8 Logic Stages for a 16-Bit Adder (not all gate inputs shown)

3.2.2 8x8 REGISTER FILE

Figure 3.3.1: 8x8 2-Read / 1-Write Register File

The register file shown in Figure 3.3.1 is comprised of eight (8) 8-bit words with two read lines and one write line. There was nothing ground breaking about the design of the register file. Two (2) pairs of eight (8) MUXs are used to separate the various outputs, as

it is a two-read register file, and three decoders are used to control the data flow. Finally, SRAM cells are used to hold the individual bits. The circuit is not complex, it is just incredibly large; see Figure 3.3.1 for complete circuit.

Each subcircuit has a relatively basic design. The 8-to-1 Decoders are created using inverters and AND-gates. The MUXs are designed using twelve (12) transistors. Each 8-bit word is formed from eight (8) single 12-transistor SRAM cells. Figure 3.3.2 shows the layout of one of these bit cells.

Figure 3.3.2: The layout of a 12-transistor SRAM cell

CHAPTER 4 SYSTEM IMPLEMENTATION

Chapter 4 is an overview of the tools and their role in the overall testing procedure. Section 4.1 gives a breakdown of the major sections of the tools chain. Section 4.2 briefly describes the tools necessary to perform the failure analysis. Section 4.3 presents the custom program, which is the main component of the tool chain. All the major functions are described in the order it is run. Section 4.4 describes additional tools and scripts used for the tool chain. Section 4.5 verifies the models and the entire tool chain. Section 4.6 discusses the test parameters used for every circuit analyzed in this report. Finally, Section 4.7 describes the limitations of the tools chain.

4.1 GENERAL DESCRIPTION

This project mixes various elements of theory, hardware and software. The block diagram in Figure 4.1.1 shows the approach taken to calculate the MTTF of a single circuit under various conditions.

Figure 4.1.1: System Block Diagram Description

There were six major stages involved for each circuit:

Circuit Creation - Creation of the layout of a digital circuit in Electric VLSI and the exporting of the spice deck file for the circuit; see Figure 4.1.2.

Figure 4.1.2: Circuit Creation

Test Automation Preparation– Preparation of the input files for the program used to calculate MTTF. Also, a testing script is created to automate all of the desired tests; see Figure 4.1.3.

Figure 4.1.3: Test Automation Preparation

Circuit Data Acquisition – LTSpice is used to simulate each circuit for a given input and starting temperature to get the raw current and voltage data needed to calculate power and MTTF; see Figure 4.1.4

Figure 4.1.4: Circuit Data Acquisition

The Monitoring Program (MTTF Analyzing Program – MAP) - The calculation of the MTTF; see Figure 4.1.5.

Figure 4.1.5: The Monitoring Program

Post Processing - The calculation of other desired circuit characteristics such as average power consumption and performance; see Figure 4.1.6.

Figure 4.1.6: Post Processing

Data Consolidation– Consolidation of all the test results into an organized readable file; see Figure 4.1.7.

Figure 4.1.7: Data Consolidation

The entire tool chain requires many files and tools to produce a single file. A block diagram of the inputs and outputs are shown in Figure 4.1.8. Only four (4) files, three (3) executable programs, and a few scripts are needed to create a single spreadsheet document with the data of multiple tests. A fifth file, the .met file, is unnecessary for TDDB but can be included in the tool chain to calculate MTTF for Electromigration.

Figure 4.1.8: Summary of the inputs, intermediate programs and scripts and output of the tool chain

The main goal of the entire tool chain is to extract the appropriate voltage and current data in order to find temperature and calculate a mean time to failure. Additionally, other information is found such as performance and power consumption of the circuit. With that information, relationships between failure and many different important circuit

characteristics can be found. Ultimately, when the circuit is improved, this tool chain is useful in comparing the different characteristics and seeing what trade-offs there are.

4.2 *SYSTEM TOOLS*

$$\tau = A \exp\left(\frac{E_a}{kT}\right) \exp(-\gamma E_f) \quad (2.4.1)$$

Referring to equation (2.4.1), one can see that the general form for mean time to failure is shown in equation (2.5.2), which was discussed in Section 2.2.4. There is a constant multiplied by two exponentials, one exponential has electric field as a parameter and the other has temperature as a parameter, and each parameter is multiplied by a scaling factor.

$$\tau = a * e^{b * E_f} * e^{c * T} \quad (2.5.2)$$

With this equation in mind, the monitoring program will need to extract both electric field and temperature values from any given circuit fed into the system. Unfortunately, those two values cannot readily be determined from simply looking at circuit schematics. Many tools are needed to discover those quantities, some of which are readily available and others of which have to be created. The tools will be described in the following subsections.

4.2.1 *ELECTRIC VLSI*

A VLSI program that creates circuits that has the information of both the placement of specific components such as wires and MOSFETs as well as the realistic capacitance of a circuit. This program is important because the placement of a component can determine the amount of heat transferred to other components. Also, capacitance increases as a result of the wires connecting those components. This is especially

important when improvements have to be made because there will be trade-offs that cannot be simulated by simply modifying a Spice file. All relevant technology information was discussed in Section 3.1.1.

4.2.2 LTSPICE

LTSpice is a high performance Spice III simulator used to simulate the circuit and provide the appropriate voltage and current data for a given input. LTSpice is an important tool because it is able to calculate the node voltages and line currents accurately for a given technology setting and temperature. Ultimately, this information is necessary to calculate a proportional electric field, as well as the power necessary to calculate temperature.

4.2.3 LTSPICE UTILITY

The LTSpice Utility is a small program independently developed from LTSpice. It has many functions, but for the purposes of this project, it is used to convert the raw binary data files that LTSpice outputs to an ASCII format that can be used by the monitoring program. When manually outputting data from LTSpice, the data was arranged in a certain format readable by the monitoring program. Manually retrieving the data was a repetitive process, so a shell environment was used to run LTSpice and the utility was run at the completion of the simulation to convert the raw LTSpice file to the usable format.

4.2.4 HOTSPOT

HotSpot is an accurate and fast thermal modeler suitable for use in architectural studies. Given power consumption, displacement and dimension information about a circuit, HotSpot can calculate the steady state temperature of a region within a circuit and calculate the affect of a particular region's temperature on other regions.

4.2.5 MONITORING PROGRAM - MAP

A program is necessary to monitor all the data output from LTSpice. The monitoring program also needs to know the various regions within the circuit as well as where each individual MOSFET is located. The monitoring program calls HotSpot and once all the data is consolidated, it determines a relative mean time to failure for TDDB. In the following sections, the monitoring program will be referred to as “MAP,” which is short for “MTTF Analyzing Program.”

4.2.6 SIMPLE SCRIPTS

Simple programs were written in Perl that run the necessary tools more efficiently through automation. These scripts were created to format information, modify files, and test circuit performance. These tasks could be done by hand, but doing so would be too tedious and would take much time.

4.2.7 EXCEL MACROS

The data output from MAP was organized in a very specific way for all circuits tested. Often times, the same sets of data contained relationships that were interesting to see for different types of circuit. For this reason, macros were created to automatically separate, process, and graph the data.

4.2.8 THE TOOL CHAIN

The entire tool chain runs in the Ubuntu Linux/Unix environment. The main reason for this is because the Unix environment is ideal for creating, compiling, and running C programs and Perl scripts. Other underlying reasons are unfamiliarity with programming in a Windows environment, and improving MAP’s ease of use through utilizing various Unix programming capabilities, such as the ability to call other programs and redirect its output back to the calling program that were used.

4.3 *MTTF MONITORING PROGRAM - MAP*

MAP is written in C in the Unix environment because of my familiarity with the programming language, and also because it can execute other Unix-based programs, which makes running MAP for the user much simpler. MAP is divided into eight (8) major functions:

- processConfig() – This function processes the configuration file for each simulation of the circuit. This file is crucial to MAP as it provides the details of all the important paths and files needed for MAP to run and other necessary information.
- create() – This function reads the PSpice file and catalogs all MOSFET and connections.
- assignRegions() – Once each MOSFET is cataloged, the relationship file is read and assigns the regions to each individual MOSFET.
- monitor() – This function monitors the voltage at the gate of each MOSFET and the current in each connection.
- printPower() – This function creates the power trace file.
- getTemperature() – This function calls HotSpot and reads its output (temperature) and assigns the appropriate temperature to the appropriate MOSFET.
- changeMetal() – This function reads a file and changes the line widths between transistors to the widths specified in the file. The metal heights are assumed to be the same throughout the circuit.
- dMTTF() – Given an average gate voltage and a steady state temperature, a mean time to failure is calculated.

After MAP runs its course, it will output an excel file with the name of each MOSFET as named by LTSpice, a mean time to failure, a temperature in Kelvin (K), an

average electric field, a gate width, and the total power dissipated by the MOSFET and by the entire circuit.

4.3.1 CREATION OF INTERMEDIATE FILES

MAP needs additional files in order to perform an analysis of a circuit. As stated in Section 4.2, many tools are used to calculate MTTF, but in order to translate information between the tools, four (4) major files are needed. One of these files is the floorplan file, and is used to translate information for HotSpot. An example floorplan for a 10mm x 5mm rectangle will look like Figure 4.3.1, where b0 and b1 are functional blocks.

Figure 4.3.1: 10mm x 5mm Rectangle Floorplan

The floorplan file corresponding to this example would be similar to Figure 4.3.2.

<unit-name>	<width>	<height>	<left-x>	<bottom-y>
b0	0.010	0.0025	0	0
b1	0.010	0.0025	0	0.0025

Figure 4.3.2: Floorplan File of 10mm x 5mm Rectangle

The power consumption data corresponding to this floor plan, which HotSpot refers to as a power trace, would be similar to Figure 4.3.3 where the numbers under b0 and b1 are Watts per calling interval [13].

b0	b1
7	2
5	1
7	2
5	1
...	

Figure 4.3.3: Power Consumption of 10mm x 5mm Rectangle

For the purposes of the project, a region of MOSFETs or sub-circuits is grouped together into functional blocks so that MAP is able to keep track of the power consumption of each individual block. However, that information is not readily available. A file must be made that relates each MOSFET, subcircuit, group of MOSFETs or group of subcircuits to a predetermined region. For this, a relationship file must be created that is specific to MAP and is independent to HotSpot. The next file that needs to be created is a power trace file. This file is created in MAP using the LTSpice data as described in Section 4.3.6. Another important file is the Spice deck file, which MAP uses to keep track of activity. The final file needed for MAP to run is the raw current and voltage data that the LTSpice utility converts from LTSpice. Appendix A goes into further detail about the creation of these files.

It is assumed that most of the power dissipated in a circuit is from the MOSFETs and not the metal lines. So, when choosing a region of the floor plan, originally, the selection of the region was based solely on the gate of the MOSFETs, but because of HotSpot's limits for the size of a region, regions as small as MOSFETs were not possible. Instead circuits like NAND-gates and NOR-gates became the smallest region of a floor plan.

4.3.2 *EXTRACTING RUNTIME INFORMATION*

MAP needs multiple pieces of information to run and the configuration file that it reads provides that information. The information includes the location of the HotSpot program and the name of the circuit, relationship, floor plan, raw data, and the metal width files. The other information it extracts is used to modify the MTTF equation. Four constants can be set that represent the constant in front of the equation, the multiplier for each of the terms inside the exponential terms and an offset; see equation (4.3.1). The rest of the data is information used to produce the excel spreadsheet that has the MTTF data.

$$\text{MTTF} = A * \exp(B * E_f + C * T) + D \quad (4.3.1)$$

4.3.3 *CREATING A CATALOG*

In order to catalog each MOSFET, MAP needs to read the PSpice file. It goes through the file line by line looking at the first word of each line. Resistors and capacitors are ignored but subcircuit definition, MOSFET and subcircuit declarations, and the .TRAN statement are each treated in a special way.

When MAP reads in a MOSFET, it creates a MOSFET object recording its name and whether it is a PMOSFET or NMOSFET. Three connection objects are created representing the metal line between the gate, drain, and source to a node. The objects are “attached” to the MOSFET terminals. Next, the nodes connected to that gate, drain, and source are checked to see whether or not those nodes are already created by the MOSFET. If the node, or nodes, exist then the connection objects are connected to the appropriate nodes. If the node does not exist, the node is created and named as declared

in the PSpice file. In all cases, any MOSFETs that are created are connected to three nodes.

When MAP reads a subcircuit definition, MAP behaves as if it is creating a separate circuit. It first creates the appropriate number of interface nodes or outer nodes and then, as it reads a MOSFET line it creates the MOSFETs and the nodes, if the node does not already exist. The names of each MOSFET and node reflect that it is part of a subcircuit. For example, Electric will name the 5th PMOSFET in the 29th “xor” subcircuit, “xor@29:Mpmos@5.” The resulting circuit is saved into a catalog.

If a subcircuit declaration is encountered, all the MOSFETs and nodes are copied, and if a declaration was encountered inside a subcircuit, then the MOSFETs and nodes are renamed to reflect that they reside within a subcircuit and connected with the rest of the main circuit. Luckily, Electric creates a Spice file with the inner most sub-circuits appearing closer to the top of the file before sub-circuits that contain other sub-circuits appear.

When MAP encounters the `.TRAN` statement, the first argument is saved and is interpreted as half the period of the fastest input. This is because the `monitor()` function records several power readings; each individual power reading will be the power over that interval. The second argument is saved as the entire length of the transient analysis.

The main job of the `create()` function is to record the names of each node and MOSFET as LTSpice knows it as, because the LTSpice output file contains the specific names and if the names are not recorded exactly, then the various currents and voltages

cannot be monitored. Ultimately, all the nodes and MOSFETs are saved into a circuit object which makes accessing each individual circuit element easier.

4.3.4 ASSIGNING MOSFETs TO REGIONS

Once the MOSFETs are named appropriately, they must be assigned to a region. In order to do this, the relationship file is read in. The format for this file is shown in Figure 4.3.4.

```
REGION MOSFETName MOSFETName ... subcircuitName subcircuitName ...
```

Figure 4.3.4: A code representation of sub-circuits grouped into blocks using Electric coordinates

In order to assign the regions, the words after the ‘REGION’ are compared with the name of each MOSFET in the circuit object. If a match is found, the ‘REGION’ is saved into the MOSFET. If there is an error in the relationship file, for example if a MOSFET is not assigned a region, MAP stops execution. A list of the regions is also saved into the circuit object in order to quickly access the information for creating the power trace file.

Because of the complexities of the metal lines that connect each MOSFET, the metal lines that connect the MOSFET to each node are assumed to be inside the region of the MOSFET even though they may not be in the Electric VLSI view.

4.3.5 CIRCUIT USAGE MONITORING

After everything is cataloged, monitoring can be done via processing the LTSpice data file. The top line of the data file contains the traces, so the first line is processed element by element. For example, an element may be Id (*MOSFET name*), or V (*node name*), and the information in the same column represents the value at a certain time.

The point in time is the first element displayed on each line. If the name inside the parenthesis matches the name of one of the MOSFETs or nodes, then its position or index is saved in the node or connection object. This is done to make information retrieval quicker within a file. The indices that are saved are the nodes connected to the gate, drain, and source, and the I_d , I_g , and I_s parameters, which represent the drain, gate, and source current from the MOSFET. If a node or connection is not assigned an index number, the value is assumed to be zero for all time.

After the indices are saved, the file is processed line by line. Power, voltage, and current averages are calculated using a left Riemann Sum. This is done because the information provided by the data file is expressed discretely as shown in Figure 4.3.5. At a certain point in time the voltage or current is a certain value. It would be too difficult to get the intermediate points or interpolate the data into an equation, so the left Riemann Sum is used because it is a lot simpler to program; see Figure 4.3.6. A large error is avoided because each time interval is not predetermined; LTSpice makes the time interval as small or as big as necessary with large steps having data that is mostly constant. To verify the accuracy of MAP, the average voltage or current calculated is compared to the average calculated by LTSpice and the value is equivalent.

Figure 4.3.5: A Visual Representation of the Data as it Appears in the LTSpice file

Figure 4.3.6: The Left Riemann Sum of the data

In order to calculate the average power in an interval, MAP first gets the voltage at the drain, source, and gate of the MOSFET as well as the currents in and out of the MOSFET. MAP then uses those values to calculate the instantaneous power and multiplies the value that by the difference between the current time and the time at the next iteration. Once the time goes past an iteration of half of the period as defined in he parameters file, the average power is saved into the MOSFET object. So if the fastest

signal switches every 1ns and the transient analysis is 10ns long, each MOSFET object will have ten (10) power values, each representing the average power in each interval.

The average gate voltage and the average current of each connection are calculated the same way except that the value is calculated as an average for the entire transient analysis instead of for each half period. Once the end of the data file is reached, monitoring is complete and the function returns to main.

4.3.6 CREATION OF THE POWER TRACE FILE

When the power has been calculated, the power trace file can be created. MAP goes through the list of regions saved in the circuit object and uses a linear search to find out which MOSFETs are in region currently being processed. The average power for the interval for every MOSFET in the region are added together and ultimately output to the power trace file. This procedure is done for every region and every interval one hundred times in order to give HotSpot the data necessary to determine a steady state temperature.

4.3.7 CALLING HOTSPOT

Once the power trace file is created, MAP executes HotSpot and provides the appropriate arguments that would allow the temperature in each region to be calculated. The output of HotSpot is redirected to MAP where the data is parsed, and then the final temperature is saved to each individual MOSFET for a particular region

4.3.8 CHANGING METAL WIDTHS

For cases when metal lines need to be adjusted from their default widths, MAP reads in a file and then adjusts the appropriate lines. MAP is very limited in the amount of information that can be input. This in turn leads to a lack of geographical and dimensional information from the layout tool. The connections between the various

terminals of the MOSFET are assumed to be straight lines that do not turn and are assumed to have the same dimensions even though they may not.

4.3.9 CALCULATING MTTF

After all the important information has been calculated and saved, a mean time to failure for each MOSFET can be calculated. Equation (2.5.2), created from the work mentioned in Section 2.2.4, is used with temperature and voltage values as inputs. MAP outputs several spreadsheet files with additional information, which is done in the `dMTTF()` function. Multiple constants can be input into the configuration file. The function calculates the MTTF for every combination of constants and outputs the appropriate spreadsheet. The total power for each MOSFET for the entire transient analysis is also calculated.

4.4 ADDITIONAL TOOLS

Because C's primary function is not to be a text processor, Perl scripts were written to set up the test environment. Perl scripts were also written also to calculate extra pieces of information that could be important when trying to find a correlation between anything else like power or performance and mean time to failure.

4.4.1 SETTING UP MULTIPLE TESTS

One script called `spiceMultiply.pl` was written to create multiple Spice files, configuration files, and shell scripts that would test a circuit under various conditions. More specifically, the `spiceMultiply` script takes in a single PSpice file, a configuration file for MAP, a parameters list and then creates multiple copies of the Spice file under different conditions as well as the shell script to simulate the circuit and calculate its MTTF.

The parameters list allows the user to vary the conditions under which the circuit is simulated: half the input period, temperature at which the circuit is run and rail voltage. Also, multiple constants can be entered to modify the MTTF equations. The script will produce a circuit file for every combination of frequency, temperature, and V_{DD} . It will also create a shell script to run the appropriate pre and post scripts, simulate each version of the circuit, and calculate its MTTF. For of the tests, a larger shell script file is created that will invoke each of the individual shell scripts, effectively running the entire simulation, take the lowest MTTF for both TDDb and electromigration, and write its information to an excel file. Electromigration is included because of a concurrent project that uses the same tool.

4.4.2 CHANGING THE TEMPERATURE

HotSpot can calculate temperature at a different started temperature as the temperature at which the PSpice file is simulated under. In order to avoid this and to ultimately avoid producing erroneous results, a script was created to consolidate the two files. The script reads the argument of the `.TEMP` parameter in the PSpice file and alters the initial temperature in the configuration file of HotSpot.

4.4.3 CONVERTING THE LTSPICE UTILITY DATA FILE

The format of the raw data file that MAP can process is different than that which LTSpice and the LTSpice Utility produces. In order to make the file compatible, a script was written to convert those files to a compatible format.

4.4.4 CALCULATING AVERAGE POWER FOR EACH REGION

Another important piece of information that could be used is the average power of each region. This is done by simply adding up all the values for each region in the power

trace file and dividing that by the number of iterations. The average power is appended to the end of the spreadsheet that was created by MAP.

4.4.5 THE SPEED TESTER

The speedTester.pl script was written to test the delay for a circuit given the data file and the rail voltage. This can be done manually by looking at when a signal reaches 10% or 90% of its maximum depending on if the signal should be high or low; see Figure 4.4.1 This is tedious and inefficient especially if a circuit has multiple outputs, and the transient analysis is very long compared to the period of the fastest input. For example, trying to find the delay for every possible output of a 4-Bit ripple carry adder would require looking at five (5) different signals 512 times.

Figure 4.4.1: Manually Finding the Delay

The alternative would be to use the data file again to determine delay. A script was written which requires three (3) pieces of information: the nodes to monitor, the rail

voltage, and a value equal to half the period of the fastest input. As was done with the monitoring section of MAP, given the nodes to watch, an index can be assigned to the location of the particular node. With the rail voltage, the script can calculate the 10%/90% values. Finally, the half period value is needed to tell the script when to start timing the delay.

The script assumes the circuits tested work correctly, that is, it assumes the output of the circuit is steady before the next clock cycle. The script can determine if that particular iteration of the output should be a high or a low. This is done by checking the value of the node right before the start of the next iteration. That is, if an inverter has an input signal that switches every 1ns, in order to tell if the output should be high or low, the script checks the value of the output node at 0.999ns since the output will not switch before the input is changed.

Once that information is acquired, the delay must be measured. Because LTSpice does not usually have a value at exactly the 10%/90% point, the script gets the two (2) points around the 10%/90% point and calculates the equation for the line connecting those two points as a function of voltage. Once that is done, the script plugs in either the 10% or 90% value to get a delay estimate. As seen in Figure 4.4.2, this can create some error because the signal will not be linear, but this is as accurate as measuring the delay by hand because as seen in Figure 4.4.1, LTSpice has line segments connecting the points.

Figure 4.4.2: Visual representation of a continuous waveform against the discrete representation of LTSpice

Another potential problem is glitches. This is accounted for by keeping track of which nodes cross the 10%/90% value twice in one interval. With that information, the first crossing can be ignored and the data from the second crossing can be used to calculate delay.

The script calculates the delay for each iteration of each node and records the largest delay. Once the script has calculated the largest delay, it appends that value to the end of the spreadsheet that was created by MAP.

4.4.6 GATHERING THE RESULTS

If a Spice file is created for every combination of temperature, input frequencies, and rail voltages. If three different temperatures, input frequencies and rail voltages were chose, twenty-seven (27) different analyses would be done by the entire tool chain. In order to combine all of the results, a script was written. The script goes through every excel spreadsheet that was created, locates the one with the lowest MTTF and copies its information to a larger excel spreadsheet. The final result is a spreadsheet of the data of

all analyses organized by the different constants, temperature, rail voltage, and input frequencies.

A sensitivity analysis was done as well. But this was done through an excel spreadsheet. The equations from the models were simply put into the spreadsheet and the effects of the electric field or temperature were amplified. That data will prove useful when analyzing a circuit where the effects of temperature are too great to counteract any useful electric field improvement. The MTTF data that was added to the spreadsheet was when the circuit did not add any heat to the ambient temperature, when the electric field was at a value that maximized the life of the circuit, and when the affects of the temperature were increased.

4.5 VERIFYING THE MODELS - THE BASIC TRANSISTOR

A single inverter was tested by doing a voltage ramp test in several papers [3] [4] [11]. In order to test the inverter, the input of the inverter was changed from 1.0V to 5.0V in increments of 1.0V at three (3) different temperatures: 100 Degrees Celsius (373 Kelvin), 150 degrees Celsius (423 Kelvin), and 200 degrees Celsius (473 Kelvin). MAP was run and the MTTF data points were compared to the original paper's data [17] and the results were accurate. Because of the need to generalize the equation for all electric field intensities and temperatures, differences between the model and the data arose, but the amount of error is negligible because units for the final result for MTTF is years instead of seconds. When comparing the model created for this project to the original data, there was only a difference of hundreds of thousands of seconds which translates to a few days. All graphs pertaining to collected data will have a normalized MTTF to simply show the relative difference in the MTTF for different variables. A graph showing

the relationship between temperature, electric field intensity and MTTF is shown in Figure 4.5.1.

Figure 4.5.1: MTTF vs Electric Field Models created from Real Data

4.6 TESTING THE CIRCUITS

The circuits were tested under various conditions. Three different variables were changed, the rail voltage and consequently the input voltage, the starting ambient temperature of the circuit, and the frequency of the inputs. The rail voltages chosen were 1.00V, 1.25V and 1.50V. The starting ambient temperatures chosen were 100 degrees Celsius (373 Kelvin), 150 degrees Celsius (423 Kelvin), and 200 degrees Celsius (473 Kelvin). For the rest of the report, Kelvin will be used as the standard unit of temperature. Three (3) different input frequencies were chosen to simulate a circuit: 400MHz, 100MHz, and 25MHz. These may seem like arbitrary input frequencies, but when choosing appropriate inputs, it was easiest to determine the “speed” of the inputs by

the period, as the period of a signal is one of the parameters needed when trying to generate a square wave in PSpice. Also, an input frequency that would stress the circuit was desired; an input having a period of 2.5ns allowed the output of a minimum-sized inverter to stabilize for a fair amount of time before the next transition of the input. Originally, the periods chosen were 2.5ns, 5ns, 10ns, 20ns, and 40ns (400MHz 200MHz, 100MHz, 50MHz, and 25MHz respectively), but having five (5) different values for input frequency would have increased simulation time by over half. For most cases, such as with the gate and the smaller circuits, the inputs were selected so as to get every possible input output pair effectively tested for an entire circuit, but for larger circuits, a certain input was chosen to test a certain part of a circuit.

All the data for each of the cases were consolidated and the data for the “weakest link” or the MOSFET that had the lowest MTTF was collected and placed into a single spreadsheet. A graph was created that showed a summary of the data for the twenty-seven different data points. The analysis of this data is discussed in Chapter 5.

Regardless of the complexity in a design, failure times and locations for any size circuit can be found with MAP. Although the smaller circuits are insignificant on their own, it is still beneficial to know the circuit’s lifetime; it will be easy to understand why a larger circuit fails if the smaller less complex circuit is understood.

4.7 LIMITATIONS OF THE TOOL CHAIN

There are many limits to the entire tool chain, most resulting from the imperfections of MAP. One important limitation is the time used for the analysis. Most of the important work is done with tools that already do something for a specific purpose. MAP attempts to interface the various tools together and, in doing so, a lot of efficiency

is lost. For example, MAP has to call LTSpice to get the voltage, current, and time data. Before that data can come back to MAP to be processed, it must be uncompressed in order for MAP to be able to read it. At the same time, this process uses large amounts of memory, especially for circuits created in Electric; there are many extra nodes and connections that Electric produces to account for parasitic capacitances and resistances. LTSpice calculates the voltage and current data for those nodes even though, in the overall picture, that data does not matter. LTSpice still has to compute and output that data which MAP ignores.

Another important limitation is the accuracy of the analysis. Section 4.4.4 referred to performing a Left Riemann Sum to approximate the discrete data output from LTSpice, and this addition causes some inconsistency with the actual value. This causes the temperature and the electric field to be inconsistent with what the true value should be as well. In addition to the data being slightly inaccurate, the equation created to model failure for a given temperature and electric field is not completely accurate, and as mentioned before, the model cannot account for phenomena like Joule heating as shown in Figure 2.5.5. The mathematical model summarizes the predictable mathematical relationship between failure and the two given parameters that MAP can calculate from the circuit and dimensional data. Finally, the analysis is only as thorough as the testing done with it. For large circuits with many inputs, the inputs must be varied a certain amount to fully test a section of the circuit or the entire circuit.

CHAPTER 5 CIRCUIT EVALUATION

Chapter 5 presents the results of testing various circuits. Section 5.1 describes the test environment and the conditions of the test. Section 5.2 presents the MTTF data for a thoroughly tested inverter in order to try to find relationships between MTTF and different tests. Section 5.3 presents failure analysis for large circuits like a Ripple Carry Adders, a Kogge-Stone adder, and a register file. Section 5.4 explains the relationships between electric field, temperature, and MTTF. Appendix D contains a thorough analysis of gate level and semi complex circuits.

5.1 INTRODUCTION

The next chapter discusses the results from analyzing circuit using MAP and the various other tools. After MAP was completed, the main task was to verify that the model created to calculate failure given a temperature and electric field was accurate. Modifications could be made to the equation to model different data sets, but for testing purposes, the data from the original paper was used to verify validity [17].

The circuits were tested in an environment where the ambient temperature was 373, 423, and 473 Kelvin. This was done because the MTTF data from the papers were collected for high electric field and high temperature [15] [17]. The temperature was set high because microprocessors operate under very high-temperature environments, and because of the exponential relationship between temperature and MTTF, testing the circuits at room temperature would lead to an MTTF of trillions of years. If HotSpot indicated a temperature that was significantly above the settings in the Spice file, the circuit was re-simulated with the temperature parameter updated. By default, the MOSFETs are assumed to have a thickness of 14 Å.

5.2 *THE INVERTER*

The inverter is the most basic gate in a digital system and represents the topology of all CMOS circuits. With this in mind, it makes sense that if one studies the MTTF of an inverter for various conditions, then one can predict the MTTF for larger circuits based on the findings of that inverter.

In order to study the relationship of various factors against mean time to failure, six (6) different inverters were tested, each with a varying gate width. The results presented in this section are for the PMOSFET only because the NMOSFET had an almost identical MTTF; the only reason for the difference is because of the parasitic resistances which changes the voltage at the gate of the NMOSFET, changing the electric field, and ultimately the MTTF. The input to the inverter was a square wave of varying frequency with a 50% duty cycle.

5.2.1 *THE AFFECT OF VARIOUS INPUT AND CIRCUIT PROPERTIES*

The circuits were tested in an environment where connections between terminals of a MOSFET were wires, which have capacitance and resistance. This adds to the delay of a circuit and acts as a load that each gate and subcircuit must drive, which adds to the power consumption and temperature, which ultimately lowers the life time. The main reason for simulating a circuit with parasitic wire capacitance and resistance is because when changes are made to increase the lifetime of a circuit, there will be certain drawbacks which could come in the form of increased power consumption or lower performance.

The rail and input voltage also have an influence on MTTF. For this test, the gates of the inverters were connected directly to the input. The input and rail voltage is

proportional to the MTTF. The data is shown in Figure 5.2.1 shows the effect of increasing the rail voltage. The higher the rail and input voltage is, the lower the MTTF.

Figure 5.2.1: MTTF vs. Input Period Plot Example for Different Rail Voltages

The effects of power on the inverter are similar to the effects of input and rail voltage on an inverter. One major difference is that MTTF is proportional to the square of the power. As shown in Figure 5.2.2, the power usage is directly proportional to the MTTF of a circuit regardless of the size and configuration of the inverter. The data shown is the MTTF plots for the PMOSFETs for six (6) different inverters. Each point represents the power usage of the inverter under different stress.

Figure 5.2.2: An MTTF vs. Power Plot Example

Figure 5.2.3 shows the relationship between the gate size of the PMOSFET, the input frequency, and MTTF. As expected, the inverter with the largest gate area has the lowest MTTF. At very low input frequencies, the MTTF is less affected than if the input frequency is very high.

Figure 5.2.3: Example of an MTTF vs. Input Frequency Plot

For a constant voltage, the electric field is the main factor for a low MTTF, whereas the temperature is not because there is very little static power consumption. For a switching input, the MTTF gets much smaller the higher the input frequency. Although, it will be much lower than a static input because of the balanced electric field throughout the circuit.

The complementary nature of CMOS circuitry ensures that half of the transistors will be on while the other half will be off even if a particular circuit is not being used. This can most easily be seen in an inverter where the duty cycle is not 50%. If the input is high 60% of the time and low 40% of the time, the NMOSFET transistor will have a much lower MTTF than the PMOSFET transistor. The opposite is true when the input is

high 40% of the time and low 60% of the time. This concept will be very important when analyzing circuits with many different input-output pairs.

5.3 *COMPLEX CIRCUITS*

5.3.1 *ADDERS*

The adders were tested in a different way than the inverter. Since there were many possible input/output pairs of the adders, the input was carefully chosen in attempts to stress the circuits equally. For the ripple carry adder, the inputs chosen ensured all gate voltages had close to the nominal gate voltage. The Kogge-Stone adder is much more complex than the ripple carry adders so there was a lot of inactivity which led to high electric fields and consequently low relative MTTF compared to other MOSFETS in the circuit.

5.3.1.1 *Ripple-Carry Adders*

Three RCA circuits were analyzed: an 8-Bit, 16-Bit, and 32-Bit RCAs. The full adder block analyzed in the previous section was used as the full adder components. In order to test the RCAs, the inputs were chosen so that the A inputs and the Carry-in inputs were switching twice as fast as the B inputs. This input scheme was able to equally stress all of the MOSFETs so that there was no inactivity throughout the circuit. The circuit matches the general trend discovered when testing simple circuit; the faster the input frequency, the lower the MTTF.

There are two noteworthy differences between the RCAs and the previous circuits. One difference is the inclusion of self heating. All the MOSFETs have close to the optimal average gate voltage, so temperature becomes a large factor in the MTTF of the circuits. As expected, the temperature gets higher the faster the RCA is run as well as the

more complex (i.e. the more bits it can add) the RCA is. The other difference is the MTTF of a circuit when it is too slow (i.e. the delay is longer than the clock period). Many circuit anomalies can happen when the input frequency is faster than a circuit can handle. The circuit acts like a low pass filter and so voltage remain at a certain level making certain circuit area have no activity and effectively maximizing the electric field lowering MTTF.

A comparison of the different RCAs will be limited to an input frequency of 25MHz, which is the fastest input frequency at which the output stabilizes before the next clock cycle. Figure 5.3.1 shows the MTTF against the different sized RCAs. For the 8-Bit and 16-Bit adders, the MTTF is virtually the same but offset, but the 32-Bit adder has a significantly lower MTTF because of the self heating affects. The 8-Bit and 16-Bit RCAs' temperature only increases a small amount ($< 1K$), but the temperature of the 32-Bit RCA raised 4K. The heating further increased with the increasing of the rail voltage. Again, the increase in temperature 32-Bit RCA was more dramatic than the two lower bit RCAs.

Figure 5.3.1: Comparison of 8-Bit, 16-Bit, and 32-Bit Adders

5.3.1.2 Kogge-Stone Adders

Only one Kogge-Stone adder was analyzed – a 16-Bit KS adder. The design of this adder was discussed in Chapter 3 and Appendix B. The KS adder was tested in the same way as the ripple carry adder. Unlike the RCAs, the input did not ensure 50% activity; the complexity of the circuit made it so that many circuit elements were dormant or active all the time, as a result, the MTTF is a lot lower than it should be. A special approach was used to analyze the KS adder. First, when analyzing the circuit, it was assumed that for the lifetime of the circuit all of the elements would have 50% activity and that the same trends for other circuits would apply to this KS adder; the added delay of each stage would ultimately make the last stage of a circuit the quickest to fail because of the asymmetric rise and fall times. Referring to the critical path of the adder shown in Figure 5.3.2, it can be seen that the last XOR-gate will be the most likely to fail. Code

was written specifically to focus on the last stage as well as some of the sub-circuits that were about 50% active.

Figure 5.3.2: Critical Path of the KS Adder

One major difference between the KS adder and all other circuits analyzed is that with increased input frequency, the average electric field of the weak links increased dramatically. The electric field contribution added to the increased power consumption and to the higher self-heating temperatures, and dramatically decreased the MTTF the higher the input frequency became. This particular circuit was optimized so that the gate would be pre-charged slowly, where time is not so much of a factor, and then discharged quickly. Just like in the smaller gates, the inputs to the particular weak link located at the XOR-gate that outputs the seventh Sum bit, suffer from the asymmetric rise/fall times which results in a very low average voltage and in a high electric field for the PMOSFET.

The KS consumed a large amount of power. The power it consumed increased the internal temperature by 5 Kelvin to 11 Kelvin depending on the rail voltage. Figure 5.3.3 shows the effects of starting ambient temperature, rail voltage, and input frequency. The Y-axis is the normalized MTTF while the X-axis is the average power consumed by circuit. As stated earlier, twenty-seven (27) different tests were run. In the graph, nine (9) lines are plotted and each line contains three (3) points. Each point is the MTTF and power value for a specific input frequency. In Figure 5.3.3, the rightmost point of each line is for the circuit when the input frequency is 400MHz while the leftmost point the

data when input is 25MHz. Each group of three lines represent the lifetime of the circuit at three different rail voltages for three different starting temperatures as discussed in Section 4.6. For the starting temperature, it appears to offset the starting MTTF of the circuit. As the rail voltage is increased, the MTTF becomes offset a little more due to the increased maximum electric field. The higher the rail voltage, the more power the circuit is able to consume, which is shown by the last point in each line being plotted further to the right. More power relates to higher temperatures, which is proportional with power consumption. The higher the input frequency is, the lower the MTTF. This is due to the increased average electric field at the gates of the weak link. Also, running the circuit faster increases the temperature which contributes to the failure. In Section 5.4, the electric field and temperature effects will be analyzed separately using this data.

Figure 5.3.3: MTTF data for the 16bit KS Adder run at different input frequencies, rail voltages, and temperatures

Certain circuit elements were given special attention, such as the PPA layers (odd and even) and the standard gates. While it is impossible to determine where the weakest link would be without testing the circuit with equal inputs, it is possible to analyze the MTTF of each subcircuit and attempt to improve on its design for the given test vector. In Chapter 6, each subcircuit, modified or unmodified, will be analyzed individually since it will be assumed that that subcircuit is the weak link for the circuit.

5.3.2 8 X 8-BIT REGISTER FILE

The register file was not tested like the other circuits in this paper; instead of setting an input that would equally stress the circuit, the inputs chosen for this circuit was chosen to demonstrate its functionality and test that against MTTF. As expected, this led to many MOSFETs with electric fields that were extremely skewed due to the inability to test every possible input of the circuit. The circuit elements that have low MTTF due to inactivity will be ignored for the analysis and instead, parts of the circuit will be focused on to reinforce data and trends discovered by analyzing the smaller circuits.

For this particular test, there was temperature variation throughout the circuit which caused certain areas to be hotter and therefore most likely to fail first. Because of the way the register file was tested, this section will focus on the 8-to-1 MUXs, 8-to-3 decoders, and the inverters of the register file. The results of the register file for the given input show that the region with the highest temperature was the SRAM cells followed by the inverters surrounding those cells, then the decoders, and finally the MUXs.

Similar to previous tests, the MUX weak link is towards the end of the circuit where the output is. The MUX used for the register file was created using two (2) 4-to-1 MUXs

and a single 2-to-1 MUX. For the purposes of this paper, the output PMOSFETs for the MUX subcircuits will be analyzed. The decoders used for the register file were designed using AND-gates and inverters. The weak link for the decoders is the inverter at the output of the AND-gates. Given an equal input, the output of an AND-gate is normally low, so the PMOSFET of the output inverter is the weak link. As seen in Figure 5.3.4, the same type of graph as Figure 5.3.3 is created but there is much less uniformity, however the trend still holds. An increase in temperature shifts the MTTF of the circuit down and an increase in the rail voltage uses more power, both which lower the MTTF of the circuit.

Figure 5.3.4: MTTF data for the Register File run at different input frequencies, rail voltages, and temperatures

The register file is a simple circuit made of up much simpler sub-circuits, but the complexity is in how the overall circuit behaves when the subcircuits are connected together and given a test input. By analyzing the smaller circuits, the reasons for the weak links in the larger circuits become clearer. For this there were many different variables affecting the MTTF. For this particular circuit, there is a temperature gradient across the circuit. This temperature gradient effect made it so that hot regions increased the temperature of other regions, which lowered of the MTTF of the adjacent regions. However the regions that overheated transferred their heat to adjacent regions that produced little to no heat which helped to increase the MTTF of the hotter region

5.4 *FAILURES AND ASSOCIATED PERFORMANCE FLAWS*

There are two (2) main reasons for TDDDB: electric field, or the average electric field, and temperature. This section will explain various circuit parameters and conditions that can affect average electric field and temperature.

5.4.1 *E-FIELD ON MTTF*

5.4.1.1 *ACTIVITY*

The amount of time which a transistor is active is proportional to the average electric field in the dielectric. If a transistor is off, then the dielectric will not breakdown. The only problem with that statement is that in a CMOS circuit, when a transistor is off, another transistor is on. It is shown in Appendix D that for the AND-gate and the OR-gate, the NAND/NOR portion of the circuit was active 50% of the time making the NMOSFET in the inverter portion of the circuit active 25% of the time. Because of the complementary nature of CMOS circuits, the PMOSFET is active for 75% of the time, which causes it to be the weak link of the circuit.

Ideally, for a circuit to have a maximum MTTF, the entire circuit has to be active 50% of the time. This, however, cannot be controlled by designers, because inputs are dependent upon other circuit elements, which are dependent on external stimuli that are unknown. For a given circuit given enough time, the activity of all the transistors can approach 50%, so the tests done for the larger circuits in this project do not accurately simulate a real world test, however, useful data can still be obtained from the findings. Some circuits like a two-input AND-gate built using a NAND-gate and an inverter cannot have a global activity of 50%. With equal inputs, the NMOSFET of the inverter will be the weak link. The only input that will allow the circuit to have an activity of 50% is when both inputs are high and low half the time. That kind of input effectively makes the circuit a buffer, which defeats the purpose of using an AND-gate at all.

An alternative to designing an AND-gate with a NAND-gate and an inverter is by connecting two inverters to the inputs of a NOR-gate. With equal inputs, the activity of the circuit will be 50% but that comes at a cost of increased area, as well as increased power consumption, which could increase the temperature. Some circuits can be transformed in this way but others cannot.

5.4.1.2 IDEAL E-FIELDS

For the KS adder, trying to balance the circuit for an optimal electric field can greatly improve the MTTF of that circuit especially at lower temperatures. At higher temperatures, the circuit's improvement is not as drastic, which is a result of the exponential relationship between temperature, electric field and MTTF. Figure 5.4.1 and Figure 5.4.2 shows the MTTF versus the input frequency of the KS adder with a starting ambient temperature of 373K and 423K respectively. The solid line is the MTTF for the

ideal case where the activity is 50%, and the dotted line shows the MTTF for the activity simulated. The ideal line shows the circuit with optimal electric fields throughout the circuit, so in this case the MTTF is only affected by temperature. The dotted curves show that increase in activity severely affects the MTTF. The increasing electric field in addition to the increasing temperature reduces MTTF of the overall circuit.

Figure 5.4.1: Input Frequency vs. Normalized MTTF when Temperature = 373K

Figure 5.4.2: Input Frequency vs. Normalized MTF when $E_f = \text{The Nominal Value}$

5.4.1.3 FREQUENCY

The input switching frequency affects the electric field contribution of MTF depending on how fast the voltage rises and falls. If the rise and fall times are equal, the frequency does not affect the MTF. That is because an equal rise and fall time means the average voltage at the gate of a MOSFET will be at the nominal level. If there is a faster rise time than fall time, the MTF of the PMOSFET will increase while the MTF of the NMOSFET of the circuit will decrease due to increased NMOSFET activity. If the fall time is faster than the rise time then the opposite scenario is true.

The frequency at which the circuit is run multiplies the effect of the uneven rise and fall times. If the gate voltage of a transistor switches very fast, the voltage will spend less time at a steady high or low voltage and more time switching from high to low or low to

high. Increasing frequency will offset the average gate voltage higher if the rise time is faster than the fall time and vice versa.

Figure 5.4.3 shows two waveforms for the XOR-gate. Each waveform is the input to transistor P7 when the input frequency is 400MHz and 25MHz. When the input frequency is 25MHz the input has time to stabilize making the average electric field its nominal value. At 400MHz, the input is very erratic and has very little time to settle. For this particular input, the average voltage at the gate of the PMOSFET is lower than when the input is 25MHz so the average electric field is higher causing the reduction in the MTTF. For the case when the input frequency was 25MHz, the average gate voltage was able to reach its ideal level, because the input had time to stabilize before the input switched.

High frequency is also detrimental to MTTF because of its relationship with power consumption. The amount of power a CMOS circuit consumes is proportional to the frequency times the load it is driving times the square of the rail voltage as seen in equation (5.4.1). The higher the frequency the more power consumed, the lower the MTTF.

$$P = \frac{1}{2} f_C C_L V_{DD}^2 \quad (5.4.1)$$

Figure 5.4.3: The Waveform of the Gates of Transistor P7 in Figure XOR, at 400MHz, and 25MHz.

5.4.1.4 SPEED

Speed in this context is the time it takes a circuit to drive a load from low to high or high to low. The same effect of frequency can be seen with the speed of the gates. Figure 5.4.4 shows the comparison between an XOR-gate with small gate sizes and an XOR-gate with gate sizes four (4) times that.

Figure 5.4.4: The Waveform of the Gates of Transistor P7 in Figure 5.3.2 for Two Different Gate Sizes.

The waveforms show the voltage at the gate of P7. The average voltage that is calculated by LTSpice is 494.87mV. For a rail voltage of 1.0V, this average is slightly below half of the rail voltage. This is due to the strength of the NMOSFET transistors versus the PMOSFET transistors. The rise and fall times are not equal; the voltage falls faster than it rises. This causes the offset average. All the circuits simulated are adjusted so that the ratio of the width of the gates of the PMOSFET transistors and the NMOSFET transistors are 2-to-1, so as to equalize rise and fall time. The difference in rise/fall time determines which transistor is most likely to fail first.

A fast circuit that is run at a faster clock frequency could have the same MTTF as a slow circuit that is run at a slower clock frequency. The gate voltage will either be steady

at a high voltage or a low voltage which means the uneven rise and fall times will not have as large of an effect. One drawback to this is that a faster circuit will use more power which could increase the temperature of the overall circuit. The drawback will be compared in a later section.

5.4.1.5 GATE SIZE

The size or the width of a MOSFET indirectly affects the speed of a circuit as seen in the XOR-gate example. The more intelligently sized gate will be able to keep a nominal voltage at the gate of the load (i.e. another CMOS circuit) and a high MTTF. An excessively large or small gate will lead to lower MTTF.

5.4.1.6 TEMPERATURE

Temperature contributes to the MTTF of a circuit in more ways than one. It plays a small part in the speed of a circuit. When the temperature of the circuit increases, the resistivity of the metal increases [22]. An increase in resistivity means increased resistance and that leads to an increased RC time constant for a given circuit path. When the temperature increases, the circuit slows down. As discussed in the Section 5.4.1.4, or the “Speed” section, a slower circuit with asymmetric rise and fall times will have a lower MTTF than a faster circuit.

For relatively slow input frequencies, the gate voltage across any given gate for different temperatures will be about the same with very small differences. But for faster input frequencies, the temperature affects the gate voltage greatly. For the XOR-gate and the full adder, the gate electric field for the weak link for starting temperatures of 423K and 473K were higher, than at 373K. This is due to the increased RC time constant discussed earlier. But for certain circuits, the individually lower speed of each gate can

change the gate voltage due to the lag of the intermediate signals. For example, the full adder for the most significant bit in a ripple carry adder relies on the full adders for the least significant bits for its input. This lag can cause the gate voltage of certain MOSFETs to increase or decrease depending on various conditions dependent on the circuit. For an 8-Bit RCA, increasing the temperature increased the average gate voltage for the weak link of the circuit. Figure 5.4.5 shows the gate voltage of the weak link of the 8-Bit RCA. The dark waveform is the gate voltage at 373K and the light waveform is the gate voltage at 473K. The lag caused by higher temperatures caused the gate voltage to stay higher longer than the 373K waveform.

Figure 5.4.5: The gate voltage of the weak link of an 8-Bit RCA when run at 373K (dark waveform) 473 K (light waveform)

5.4.2 *TEMPERATURE ON MTTF*

The same factors that can affect electric field can also affect the temperature of the circuit. This project is not about researching the effects of how circuit activity or elements can affect temperature, but the effects will be summarized because designing to minimize the effects of the electric field may affect the temperature. Trade-offs are necessary.

The activity of a circuit could greatly or insignificantly influence the temperature of a circuit depending on which part of a circuit is active. Decreasing the clock frequency of a circuit leads to less power consumption which leads to lower temperature. It has been shown that the performance is exponentially proportional to power consumption, i.e., the smaller the output delay of a circuit, the more power it has to use. The size of the MOSFET affects the performance; the wider the gate, the faster the circuit, the greater the power consumption, and the hotter it becomes. One interesting side effect of starting at a higher temperature is that the self heating will be greater at higher temperatures because a circuit will use more power. This will further decrease the MTTF of the circuit.

In the case of the Kogge-Stone adder, the temperature could have been regulated in a way where the effects of self heating become negligible like in the cases of the smaller circuits (i.e. the gate level circuits). As was done with the electric fields earlier, Figure 5.4.6 shows the MTTF versus the input frequency of the circuit if the circuit stayed at the ambient temperature and shows what was simulated. The temperature of the circuit did not increase drastically and as a result, the MTTF is not significantly improved by an ideal temperature. At lower rail voltages, the improvement is a good amount, but at

higher rail voltages, the improvement is much greater. This is due to the increased power consumption at higher rail voltages.

Figure 5.4.6: The MTTF of the KS Adder at Ambient Temperature, with Self-Heating

Increasing temperature does not significantly affect electric field and consequently does not significantly affect the MTTF due to the electric field effects. Figure 5.4.7 shows the MTTF for the weak link of the XOR-gates of a 16-Bit Kogge-Stone adder. The two lines shown is the MTTF of the MOSFET when the electric field is changed as a result of a change in temperature and when the electric field is held constant (the electric field for the MOSFET at 423K and 473K is the same as at 373K). There is no major difference in the MTTF of the MOSFET; the MTTF of the MOSFET for various temperatures is about the same. For this reason, improvements discussed in Chapter 6

will not include data from simulating circuits starting at higher ambient temperatures as the data will be redundant. For the most part, temperature just offsets the MTTF up or down for a given electric field as shown in Figure 5.4.6.

Figure 5.4.7: The MTTF for a PMOSFET in a Kogge-Stone adder for a constant E-Field and a varying E-Field.

CHAPTER 6 IMPROVING DESIGN AND RESULTS

Chapter 6 describes possible implementations of circuit modifications that can improve MTTF. Section 6.1 describes a list of possible fixes and the motivations behind them. For each possible modification, there are possible drawbacks which are explained as well. Section 6.2 describes the process of improving an XOR-gate and the effect on MTTF, power, and performance. Section 6.3 discusses the results of the various circuit design techniques on small circuits. Section 6.4 and 6.5 show the results and effects of improving the Kogge-Stone adder and the register file. Finally, Section 6.6 explores the possible increase in sensitivity of a circuit to temperature and compares that with results from the previous sections.

6.1 POSSIBLE CIRCUIT IMPROVEMENTS

Based on the findings from the previous sections, there are three obvious ways to improve the weak link of the circuit. One way is to have the gate voltage switch less often. By switching less often the asymmetric rise/fall time skew will be less of an impact on the weak link. This solution, however, is out of the control of the designer. Be that as it may, an extremely intelligent control unit can distribute the “frequency load” of a circuit. For example, for a two-input XOR-gate, if the A input switches twice as often as the B input, then the MOSFETS that are affected by the A input will contain the weak link. However, if the control unit can vary the frequency load by making the A input switch twice as fast half the time and then make the B input switch the other half of the time, then that will maximize the life time of the circuit.

Another solution is to decrease the parasitic and gate capacitance between the output of one circuit and the input of another circuit. This involves decreasing the width of the

metal lines connecting the gate to the inputs and making the path from the input to the gate as short as possible. Another modification to consider is decreasing the gate size of the weak link. The size of the gate affects the capacitance of the circuit. Therefore, the smaller the gate, the less capacitance the circuit has. Reducing the capacitance increases the performance of the circuit and like frequency, will make the impact of the skewed rise/fall times less. Decreasing the fan-out of a particular gate will also decrease capacitance. One drawback to this approach is that decreasing the metal width increases its resistance, which increases the RC time constant making the circuit slower. Also, it would reduce the amount of current that can flow through the metal reducing the drive capacity of the circuit. Reducing gate width would do the same as well. Sometimes this solution is more effective as it can lower power consumption, which can lower the self heating temperatures while increasing the switching speed of the circuit. This will be discussed in later sections.

The last solution involves designing the stage before the weak-link to drive the next stage faster. This can be done by increasing the gate width of the previous stage as well as by optimizing the circuit as a whole to be faster. If the circuit is faster, the switching time decreases, which allows the average electric field to settle close to the ideal electric field. One drawback to this approach is the increased gate width would use more current, which will increase power consumption and ultimately increase the self heating temperature of the circuit.

Each of these techniques will be applied to different circuits in attempts to see which one yields the best results. The changes mentioned mainly change the electric field component of the MTTF model. Although, any change in the circuit can affect the power

consumption and ultimately the temperature of the circuit. When trying to change the average electric field for the weak link, capacitance for the input should be lowered and the stage before that should be made faster. But to lower the temperature, the capacitance of the entire circuit would have to be reduced. Another option would be to layout the circuit over a wider area so self heating affects are lessened. However, this would reduce performance because of the increased metal capacitance. Other factors that can reduce power consumption is lowering the rail voltage and decreasing the clock frequency, but both of those changes will drastically affect performance.

6.2 *EXAMPLE: IMPROVING THE MTTF OF AN XOR-GATE*

The XOR-gate is the simplest “complete” circuit in the catalog of circuits tested. By complete, it is meant that the circuit has at least two stages of logic with the first stage driving the second stage. The modifications suggested will first be tested on the XOR-gate for effectiveness. This will also be a good opportunity to see how the changes affect power consumption and performance. For this following section, it will be necessary to know that P3 was the weak link of the XOR-gate. The layout of the gate is shown in Figure 6.2.1.

Figure 6.2.1 The layout of an XOR Gate and the transistors relevant to the following sections

6.2.1 LOWERING PARASITIC CAPACITANCE

One attempt was to lower the capacitance by decreasing the width of the metal line connecting the gate to the input. This resulted in a circuit change so small that there was no discernable change to the MTTF. The metal was reduced to three-fourths its size and then one half of its size and still no change.

Another attempt to lower MTTF was by decreasing the gate size of the weak link to two-thirds of its size. By decreasing the size of the weak link, the transistor's strength decreased. The output of P3 was connected to the input of P7 which became the new weak link. The changes increase the MTTF of P3 but lower the MTTF of P7.

6.2.2 IMPROVING CIRCUIT PERFORMANCE

The last change was to increase the speed of the stage before the weak link. This was done by increasing the gate width of P5 which is the MOSFET in the inverter portion of

the XOR-gate. Doing this increased the MTTF of the circuit because of the lower average electric field of the weak link. By trying to design the circuit around the weak link, another MOSFET becomes the weak link, and again it is P7.

6.2.3 RESULTS ON SMALL CIRCUITS

Figure 6.2.2 shows the MTTF of the XOR for various changes. For this particular circuit, increasing the size of the stage before the weak link had the greatest effect, and decreasing size of the weak link had little effect. When the circuit is run faster the changes have less of an effect than when the circuit is run very slowly.

Figure 6.2.2: The MTTF of the XOR-gate for Various Changes

Figure 6.2.3 shows a bar graph of the effects of the changes. The graph on the right with dark bars corresponds to the circuit running with an input frequency of 400MHz. The graph on the left with light bars represents the MTTF of the circuit when run at 25MHz. The effects of changing the previous stage can be seen more easily.

Figure 6.2.3: The Effect of Changes with Respect to the Frequency of the Inputs

6.2.4 EFFECTS ON POWER AND DELAY

There are trade-offs for each change, but it appears that for changing the characteristics of the XOR-gate, there are not that many trade-offs. As can be seen from Figures 6.2.4 and 6.2.5, the changes made to the circuit do not have a detrimental effect on the circuit. Three sets of data are shown: the original unmodified circuit, the circuit where the gate size of the weak link is reduced, and the circuit where the stage before the weak link is strengthened. The original unmodified circuit is included for comparison purposes. For each of the changes, the delay decreased. For decreasing the size of the weak link, the power consumption went down along with the delay. However, for increases in the size of the previous stage, the power consumption went up. So for making changes to the XOR-gate, increasing the previous stage greatly improved the MTTF of the circuit but at a cost of greater power consumption.

Figure 6.2.4: The Effects of the Changes on Delay

Figure 6.2.5: The Effects of the Changes on Power

6.3 SUMMARY ON SMALL CIRCUITS

Three small changes were proposed: decreasing the gate size of the weak link, increasing the gate sizes of the MOSFETs from stage before the weak link, and finally, decreasing the metal width connecting the output of one stage to the gate of the weak link. Because of the redundant results for the MUX and the full adder, the analysis is shown in Appendix E, but the results discovered from analyzing changes for those

circuits are included in this section as well. Figure 6.3.1 shows the goal of the modifications. In order to improve MTTF, the goal is to make the rise and fall times equal.

Figure 6.3.1: The modifications goal

The most effective single modification was increasing the gate size of the previous stage. This modification makes it so that the voltage across the gate of the weak link is at its optimal value by decreasing the rise time, which makes it about equal to the fall time. By increasing the size of only the PMOSFETs for the previous stage, the PMOSFETs are strong enough to drive a load as quickly as the NMOSFETs. Also, increasing only the PMOSFETs makes that stage less prone to becoming the weak link, because the load that stage produces does not become too great. One drawback to this modification is the power consumption. The circuit becomes faster because of the increased gate sizes so it uses more power, which could increase the temperature. As seen with the MUX, this made a large difference even when the improvement to the electric field was minimal..

Another effective modification was decreasing the gate size of the weak link. This modification made both the rise and fall times faster. By doing this, the average voltage

would be less skewed by the relatively faster fall times. Decreasing the gate area of one stage lessens the load for the stage driving it. A positive side effect is that decreasing the gate size decreases the power consumption of the circuit. This was beneficial for the MUX when the lower power consumption saved the circuit from increasing the temperature due to self heating. One drawback to this method is the potential of creating another weak link. The goal is to increase switching speed, but by decreasing the size of the weak link, the strength of that gate is lessened. This will cause the rise and fall times to increase for that stage, potentially making that stage contain a new weak link.

Decreasing the size of the metal width was done to achieve the same goal as decreasing the gate size. This method did not produce any favorable results possibly because of the size of the circuits. For potentially large circuits, this modification may make a difference but for the smaller circuits, there was no significant positive change.

The last improvement, explored in Appendix E3 for more complex circuits, involved strategically increasing the PMOSFETs of the entire circuit and slightly decreasing the size of the weak link. The data showed that the fix greatly improved the MTTF of the circuits tested, especially for when the circuits were run at high speeds. While the results of the improved fix are impressive, the correlation between MTTF, power and performance are not very clear. With the full adder, the MTTF was improved, but the power consumption was the highest of all the fixes, but the circuit's performance was the worst of all fixes. The MUX, on the other hand, had the highest MTTF and the best performance, and it also consumed less power than most of the other fixes.

It appears that small localized improvements prove to be much more beneficial than designing a circuit for maximum MTTF; the MUX had ten (10) MOSFETs, compared to

the full adder's twenty-four (24) MOSFETS. Changing four (4) of the PMOSFET in the MUX produced much better results than changing the thirteen (13) PMOSFETs in the full adder. There is indeed an efficient way to improve a circuit; brute-force sizing, however, is not the answer because it can increase power consumption, and temperature could be an large influence in failure.

Figure 6.3.2 shows a surface plot representing the MTTF of the each transistor for the MUX discussed in this project. The higher the peaks are, the higher the MTTF of the transistor. Figure 6.3.2 (a) is the unmodified circuit and Figure 6.3.2 (b) is the circuit modified as discussed in this section. It appears that when trying to increase the MTTF of a transistor, other transistors are affected. By increasing the MTTF of P5 the MTTF for N6 decreased, but the MTTF of the overall circuit increased. So it appears that increasing MTTF for CMOS circuits is complimentary and that increasing the MTTF of one MOSFET means decreasing the MTTF for another.

Figure 6.3.2: Surface plot for the MTTF of the Multiplexor when it is unmodified (a) and when it is improved (b)

6.4 RESULTS ON LARGER CIRCUITS

6.4.1 Kogge-Stone Adders

Changes were made throughout the entire circuit in attempts to increase the lifetime of the overall circuit, not just the weak link. This was done by increasing the size of the AND-gate and XOR-gates on the front end and decreasing the size of the PPA sub-circuits. In other words, the load for the XOR and AND-gates was reduced in order to increase the overall switching speed in the connections within the circuit. This fix was a general fix that was aimed to improve the circuit as a whole and not fix just the weak link. Figures 6.4.1 through 6.4.4 show the results of the modification against the original circuit for the four main gates that make up the KS adder. The bar graphs on the left show the normalized MTTF of the subcircuit with a rail voltage of 1v run at two different

input frequencies. The scatter graph on the right shows the relative MTTF when the KS adder consumes different amounts of power. Each data point is the average for each rail voltage, because plotting all nine points would make the graph unreadable.

Figure 6.4.1.: Modification results for the AND Gate of the Kogge-Stone Adder

For the AND-Gate shown in figure 6.4.1, the fix was generally positive. The relative MTTF for the fix is higher when the input frequency is high. The MTTF is also lower for the amount of power the overall circuit consumes.

Figure 6.4.2: Modification results for the XOR Gate of the Kogge-Stone Adder

For the XOR Gate the results are mixed. Improvement can be seen for high input frequencies, but as can be seen from the graph on the right, it comes at the cost of power.

Figure 6.4.3: Modification for the PPA Even Subcircuit of the Kogge-Stone Adder

For the PPA Even block the results are also mixed. The relative MTTF is about the same (the unmodified relative MTTF is only higher by a very small amount), but as shown in the right type of graph in Figure 6.4.4, the relative MTTF is about the same for less power.

Figure 6.4.4: Modification results for the PPA Odd Subcircuit of the Kogge-Stone Adder

For the PPA Odd block the results are positive. The relative MTTF is higher at both low and high input frequencies. As shown in Figure 6.4.4, the relative MTTF is about the same for less power.

As shown in Figures 6.4.1 through 6.4.4, the modifications proved to be effective most of the sub-circuits of the adder, with the PPA Even subcircuit being the only component that dropped slightly in relative MTTF. Although the modifications lowered the electric field of the components in the circuit, they effectively reduced the temperature of the overall circuit. There is also a significant reduction in the overall power consumption of the circuit.

Figure 6.4.5: The results of modifications on the various sub-circuits of the Kogge-Stone Adder

Figure 6.4.5 shows the output delay as well as the power consumption for the circuit. The modification proved positive for all aspects of the circuit. For the Power Data graph, the left Y-axis corresponds to the power consumption when the circuit is run at 400MHz and the right Y-axis corresponds to the power consumption when the circuit is run at 25MHz. When run at 25MHz, there is very little difference in the power consumption. There is only a 6 μW difference compared to the 0.2mW difference when the circuit is run at 400MHz. The lower power usage lowers the temperature of the circuit by up to 1.5K. The circuit also performs better from the performance graph.

6.4.2 Register File

The Register file was comprised of three different sub-circuits: the MUXs, the decoders, and the dynamic memory cells. Improvements were tested on three areas of the register file: the MUXs, the decoders, and the inverter at the output of the decoders. The memory cells were unmodified because the circuit is passive and does not help to

propagate a signal but instead holds a signal until it is ready to be read. Two improvements were tested to improve the circuit. The first attempt involved increasing the size of the PMOSFETs, and the second attempt involved increasing both the N and PMOSFETs before the weak links and slightly decreasing the size of the weak link itself. Overall, both approaches increased the lifetime, but the second approach was more effective in some cases and less effective in other cases.

Figure 6.4.6: MTTF data for the MUX sub-circuits in the register file

For the MUX in Figure 6.4.6, the second fix was the most effective, increasing the MTTF of the MUX by a drastic amount when running the circuit with an input frequency of 400MHz. One thing to note is that as the circuit runs slower, the fixes become less effective. This is due to the electric field increases in the PMOSFET portion of the circuit and decreases in the NMOSFET. What makes the second fix more effective for fast input frequencies is not only the decrease in electric field for the weak link, but also, the decrease in power consumption, which lowered the self heating temperature.

Figure 6.4.7 MTTF data for the Decoder sub-circuits in the register file

For the decoder in Figure 6.4.7, the second fix was more effective for both the slow and fast input frequencies. The temperature increased slightly, but the electric field improved enough to where temperature was not a main contributing factor.

Figure 6.8.8 MTTF data for the Inverter sub-circuits in the register file

The data shown in Figure 6.4.8 is the same simple inverter analyzed in previous sections. Slow input frequencies led to little improvement but fast input frequencies produced better results.

Power consumption was different between the unmodified circuit and the two fixes. The first fix used much more power at fast input frequencies than the unmodified register file. The second fix used less power than the unmodified circuit and the first fix. At slower input frequencies, the power decreased with each new iteration of the circuit. There were certain sections of the circuit that were less affected by the self heating of the circuit, in this case, it was the decoder. Its temperature stayed close to the ambient temperature while the other elements, like the MUX, were greatly affected by the self heating effects. In the case where temperature increased, the second fix was more effective since it decreased the electric field of the weak link and also decreased the temperature. In cases where self heating is not a big factor, the first fix is more effective since it is more effective for reducing the electric field of the weak link.

Figure 6.4.9 Power and Performance data for the register file

The last piece to consider is the performance of the register file; see Figure 6.4.9. The delay was measured by measuring the time it takes for the read instruction to be inputted and the data to be outputted. The fixes made the circuit perform slightly better.

6.5 SUMMARY ON LARGE CIRCUITS

The larger circuits that were modified and analyzed were a 16-bit Kogge-Stone adder and an 8x8 Register File. For the KS adder, the circuit was viewed as a simple circuit only consisting of MOSFETs instead of sub-circuits. Just like the front end MOSFETs before the weak link were optimized to drive the weak links faster to improve MTTF, the sub-circuits before the weak link subcircuit was optimized to drive a given load faster. Also, in a small circuit, the gate of the weak link was made smaller so that it would be seen as a smaller load to the MOSFETs driving it. For the KS adder, the size of the entire subcircuit was reduced in order to reduce the load of the driving subcircuits. For the register file, each subcircuit was treated as its own independent component. Just like in the previous sections, each subcircuit was modified to maximize the MTTF of the weak link. This also gave positive results.

CHAPTER 7 CONCLUSIONS

Time Dependent Dielectric Breakdown has the potential to be a large problem if not addressed, especially with the scaling down of technology. There has to be circuit design solutions that exist to combat this increasing problem. This paper presented a tool chain with MAP at its core that was used to calculate mean time to failure for a given circuit. When completed, the tool chain was used to analyze a variety of circuits to simulate the wear and tear of real life circuit under stress. Using that information, I found that intelligent gate sizing of MOSFETs could increase intra-circuit switching, which improves the MTTF of the weak links of circuits making the average electric field at the gate of each MOSFET as close to ideal as possible.

The tool chain I created can be an extremely useful asset in circuit design. The main part of this project was the circuit creation process. A circuit was first designed in Electric VLSI. Then, the functionality was tested in LTSpice. Finally, the MTTF was tested with the tool chain and MAP. If there were any problems with a step, the whole process would start over. Before I started this project, I used to think circuit design was a two step process, but I realize now that in the future there will be the necessity for the third step. Presently, even with MAP, the circuit creation cycle is still incomplete. MAP can only calculate MTTF for Time Dependent Dielectric Breakdown and electromigration, but in the future, hopefully all failure types can be analyzed and modeled into MAP to make the tool chain whole and the circuit creation process complete.

APPENDIX

APPENDIX A – DETAILED INTERMEDIATE FILES DESCRIPTION FOR THE TOOLS CHAIN

A.1 MANUAL CREATION OF FLOORPLANS AND RELATIONSHIP FILES

The circuits that are created for testing purposes are created in Electric, but it does not have an option to create a floor plan that is compatible with HotSpot. Floorplans were created visually in Electric; a simple Perl script was written to take the two opposite end points of a rectangular region as input and translate that to a compatible format. The input file for the Perl script also contained information on which sub-circuits or MOSFETs are contained in each region.

```
Scale
name x1, y1 x1, y2 subckt1 subckt2... MOSFET1 MOSFET2... n 0 1 2... p 0 1 2...
name x1, y1 x1, y2 subckt1 subckt2... MOSFET1 MOSFET2... n 0 1 2... p 0 1 2...
...
```

Figure A.1: Input File for Perl Script Format

The ‘scale’ represents the actual measurable length of each unit. For example, the distance from $(x_1 = 0, y_1 = 0)$ to $(x_2 = 1, y_2 = 0)$ would be equal to value represented by ‘scale.’ The coordinate can be entered without regard to which two corners each corner since a simple if statement can determine the two different cases, the case where the first coordinate represents the upper left corner and the second pair represents the lower right or if the first pair represents the lower left corner and the second pair represents the upper right corner.

Figure A.2: Example of the Floor and Relationship

The freedom to choose coordinates is ideal because Electric has a toggle measurement tool that can show multiple continuous measurements as shown in Figure 4.3.6. As seen in a ring oscillator, several inverters are grouped together. The colored rectangular boxes represent shows which sub-circuits are grouped together.

Figure A.3: Sub-circuits as they are grouped into blocks using Electric coordinates

The parameters after the coordinates are for the relationship file. The name that electric gives each subcircuit and MOSFET instantiation, follows the coordinates. The ‘n’ and ‘p’ statements are used as a shortcut since Electric provides a default MOSFET name of “mn-MOS@0” or mp-MOS@0.” The numbers following the ‘n’ and the ‘p’ will generate the appropriate name.

The floor planning script will read each line and create two files: the floorplan which HotSpot needs, and the relationship, which MAP needs.

A.2 LAYOUT PREFERENCES AND CREATION OF SPICE NETLISTS

Circuits were created incorporating parasitic resistance and capacitance. This becomes an issue when using metal lines, and especially apparent when changing the interconnect sizing. Up to six (6) metal layers were used in a given design, yet keeping them down to a minimum was always a design concern. As far as metal sizing, there is really no concrete way of keeping this consistent. Almost all metal lines have a width of three, other than the power and ground rails, which whose widths are slightly wider. Silicon is set to have a width of two. MAP assumes all metal sizes to be of width 3 unless otherwise stated in the “met” file. In Electric widths are unitless and the final measurement is determined from the technology process discussed in Section 3.1.1.

Spice decks were automatically generated for each of the laid out designs, incorporating all layout preferences earlier mentioned. The spiceMultiply script was then used to create various versions of a given circuit (combination of input voltage, input frequency, and temperature adjustments). The netlists can then be easily simulated using LTSpice.

A.3 RAW CURRENT/VOLTAGE DATA FROM SPICE

The next major file MAP needs is the current and voltage data from LTSpice. This is done by running a simulation in LTSpice given a Spice file with inputs and a transient analysis statement and exporting the data. The exporting is done without opening LTSpice since that would break the flow of the tool chain. This done by running LTSpice from a command line and enabling the batch option, which simply runs a

simulation of the Spice file given in the arguments and outputs the binary .RAW file. Once the simulation is complete, the LTSpice Utility is run and it converts the .RAW file into another intermediate file that has ASCII characters that represent the data. This file is then converted using a Perl script to a format compatible with MAP.

This will produce a file with the currents output from every voltage or current source and the currents that flow in and out of the various terminal of each MOSFET and the voltages of every node. Ideally the inputs are chosen to stress the circuit as it would be stressed in real life as MAP assumes that the inputs are repeated for all time. For example, if an inverter was tested with a square wave input that switches every 1ns, MAP will determine a mean time to failure assuming that the input stays the same for all time. So the inputs have to be chosen intelligently in order to get an accurate MTTF calculation. The only problem with this approach is the amount of processing time and memory needed to produce the files. For slow computers, simulations of large circuits could take a long time and the files can be well over one gigabyte.

A.4 THE POWER TRACE FILE

The last intermediate file that needs to be created is created by MAP. HotSpot needs the power trace file to run a temperature analysis for each functional block defined in the floorplans. In order to create the power trace file, MAP needs both the relationship file and the raw current and voltage data from LTSpice. With the data file, MAP calculates the power dissipated by each MOSFET for every half period as defined in the transient response. Once each power value is calculated and saved, the power dissipated from MOSFET in each region is added together and output to a power trace file that is compatible with HotSpot.

APPENDIX B – THE FULL ADDER

The full adder was implemented using the equations below:

$$\begin{aligned}C_{\text{out}} &= C(A + B) + AB, \\S &= !A!BC + !AB!C + A!B!C + ABC\end{aligned}\tag{B.1}$$

The alternative to the sum would be:

$$S = A \text{ xor } B \text{ xor } C\tag{B.2}$$

This implementation is different then the intuitive way, which is to simply use gates. The gate was implemented to reduce the amount of transistors used in the design. This is done by using some of the logic to calculate the carry out bit to also calculate the sum bit.

APPENDIX C – CMOS KOGGE-STONE

The Kogge-Stone (KS) adder is one of the fastest single architectures for binary addition in parallel stages. It does however use a large amount of area and power. The standard Kogge-Stone adder design for 16-Bits is shown below. The basic tiling block sub-circuit of the KS adder is a parallel-prefix adder cell (PPA). Our design uses two types of tiles: an even tile for even rows (with row numbering starting at zero), and an odd tile for odd rows. Each tile compresses two (2) prior propagate and generate signal sets into a single propagate and generate signal set for the column.

Figure C.1: Block diagram for 16-Bit Kogge-Stone prefix adder network

C.1. FINAL DESIGN DETAILS

The propagate and generate gate circuits are shown below for even layer and odd layer tiles. Each layer produces inverted outputs from negating logic gates. The circuits

shown below are shown is CMOS, whereas our signals were created using Output Prediction Logic (OPL). This is a simple tweak and simply requires the addition of a pre-charge transistor between V_{DD} and V_{OUT} and a footer transistor at the bottom of the NMOSFET stack to ground.

One other issue with the Kogge-Stone is the unbalanced loading of individual tiles. Tiles in the top right corner of the tree are required to drive more gates than tiles in the left of the tree. This should require sizing some of the gates with higher output capacitance up to keep delay through the tree fairly even, however with the current OPL design this sizing adjustment was not made and it was assumed that the gates will drive their load adequately.

Figure C.2: Even-layer Tiled CMOS PPA circuit for inverted input (left), and non-inverted outputs (right)

Figure C.3: Even-layer Tiled CMOS-OPL PPA circuit layout cell

Figure C.4: Odd-layer Tiled CMOS PPA circuit for non-inverted inputs (left), and inverted outputs (right)

Figure C.5: Odd-layer Tiled CMOS-OPL PPA circuit cell

The architecture selected for the final design is OPL Kogge-Stone Radix-2 Prefix Adder. This adder was chosen because the PPA sub-circuits already developed and tested would properly evaluate the carry-in values for each bit in a Kogge-Stone prefix tree.

The PPAs of the KS tree generate the same carry-in for each bit that the tiles of the CLA would generate, and therefore both adder architectures require similar front and back logic blocks. The front logic block takes X_i and Y_i (inputs) and in two logic stages outputs the propagate (p_i) and generate (g_i) signals for each bit. Those signals are inputs to the KS tree, along with the carry-in for bit 0, and the outputs of the KS tree are the

carry-in (c_i) for each bit. The KS tree has five logic stages. The final two stages calculate the sum (S_i) for each bit and the carry-out (C_{16}) for the adder. The basic design requires nine logic stages from X_i, Y_i to S_i and C_{16} .

Figure C.6: CMOS OPL layout circuits for AND (left) and XOR (right) gates used for the P_i and G_i signals

The critical path for the Kogge-Stone Adder is shown in Figure 3.5.7, which consists of eight logic stages for the 16-Bit adder. Since the CMOS logic blocks are all built with OPL pre-charging transistors, the timing of the entire adder is constrained by the clock.

Figure C.7: Kogge-Stone Critical Path for 8 Logic Stages for a 16-Bit Adder (not all gate inputs shown)

The complete 16-Bit Kogge-Stone Parallel-Prefix Adder layout is shown below. The center of the layout area has three north-south parallel signal lines for V_{DD} , GND, and the clock (CLK). These lines branch at each layer of the tree to provide connection to the individual cells. The last PPA for evaluating carry-out from the G_i/P_i bit 15 column has been pushed up into the fourth row to save total area in the cell; that fourth row is shifted to the right by one PPA cell to accommodate that last row collapse.

Figure C.8: 16-Bit Kogge-Stone Parallel-Prefix Adder Layout with CMOS OPL Gates

APPENDIX D – GATE AND SEMI COMPLEX CIRCUIT ANALYSIS

D.1 GATE LEVEL

The following section is the MTTF analysis of standard gates. The findings from testing an inverter will help us predict the failure of a particular gate. Each gate tested was a two-input gate. The inputs were chosen to get every possible input-output pair; this type of input will be referred to very frequently, so the concept will be referred to as the “equal input.” Also having an average gate voltage exactly half of the rail voltage will be referred to as the “optimal voltage” which also corresponds to the “optimal electric field.” The self heating thermal effects will be ignored in this section because the amount of power produced for the smallest allowable size in HotSpot did not produce significant heat. In larger circuits, self heating will play a large part in lowering the MTTF of the circuits.

D.1.1 THE NAND, NOR-GATES, AND XOR GATES

The NAND- and NOR-gates are designed almost the same way. If given the equal input, the MTTF will be basically the same for all MOSFETs in the circuit. Figure D.1.1 shows the circuit diagram of the two gates. As can be seen, the two circuits are just flipped versions of each other. The two gates have a lower MTTF than the inverter.

Figure D.1.1: The Circuit for a NAND-gate (left), and a NOR-gate (right).

The XOR-gate is made with two inverters for the inputs and eight (8) MOSFETS arranged in Figure D.1.2. With equal inputs, all the transistors are turned on 50% of the time. As the input frequency increases, the MTTF of the circuit decreases. One of the main differences between the XOR and the AND and OR Gates is the need for an inverter to provide the inverted inputs to the main part of the gate. This differences leads to a transistor with a lower MTTF than the others rather than all transistors having an equal MTTF. For this particular circuit and for the given test inputs, MOSFET P3, as seen in Figure D.1.2, is the weakest link.

Figure D.1.2: Transistor Network that Creates an Exclusive-OR function and the layout with the relevant transistor labeled

D.1.2 THE AND- AND OR-GATES

The AND-gate was designed by putting an inverter at the output of a NAND-gate. The OR-gate was designed the same way but with a NOR-gate. The results are interesting. If the inputs are still the same as in previous cases, the inverter at the output of the NAND- and NOR-gates will be the weak link of the circuit, but the PMOSFET will be the most likely failing transistor for the OR-gate whereas the NMOSFET will be the failing transistor for the AND-gate. This is because of the activity of the circuit as will be discussed later. This effect causes the MTTF of the AND- and OR-gates to be much lower than the gates mentioned previously. For high input frequencies the MTTF of the AND-gate increase while the opposite is true for the OR-gate.

Figure D.1.3: The Weak Spots of an AND-gate and OR-gate with Equal Inputs.

D.1.3 SUMMARY

In this section, it was mentioned that increasing the input frequency can lower the MTTF of an OR and an XOR-gate, but MTTF increases for an AND-gate. With equal inputs most circuits should behave like an XOR-gate with each transistor being active 50% of the time, but for some circuit may behave like an OR- and AND-gate. In the next section, the same behavior will be seen in larger circuit and the self heating effects will be seen as well.

D.2 SEMI-COMPLEX CIRCUITS

D.2.1 MULTIPLEXORS (MUX)

Figure D.2.1 shows the circuit diagram of a two-input MUX and the layout used for simulation. The MUX utilizes inverters but none of the standard gates and instead uses the transistors to effectively connect the output inverter to the rail or ground based on what the select bit is. Based on what was learned with the two-input gates, it is expected that a PMOSFET transistor will be the weak link. Also looking at the layout it appears like the inverter before the output of the MUX will be the weak link. MAP confirms this.

P5 is the weak link of the MUX. Temperature was not a large factor in the MTTF of the circuit.

Figure D.2.1: The circuit diagram of a CMOS MUX and a layout schematic.

D.2.2 FULL ADDERS

The circuit is designed in a way very similar to the MUX, so the analysis is very similar since the circuit is one large circuit as opposed to many disjoint gate connected together. The weak link for this design of the full adder is P0 or the PMOSFET that inverts the NOTed carry out bit. Again temperature was not a large factor in the MTTF of the circuit.

Figure D.2.2: Layout and Circuit Diagram of the Adder.

D.2.3 DECODER

The decoder simulated is a 3-to-8 decoder which is made up of NOR-gates and inverters. Three inverters made up the inverter chain which inverted the three inputs and those inputs were connected to the appropriate NOR-gate inputs. The furthest PMOSFET from the inverted A signal was the weakest link for the circuit.

Figure D.2.3: Layout of the 3-to-8 Decoder.

Temperature contributed to the failure of the circuit. The circuit is designed in such a way that the failure due to the electric field was minimized making the temperature a large part of the failure. The rise in activity and the circuit size increased the temperature a few degrees Kelvin.

APPENDIX E –IMPROVING THE MTTF OF A MUX AND FULL ADDER

E.1 IMPROVING A MUX

The changes on a two-input MUX produced interesting results. The weak link was located in the middle of the circuit, so making changes to the weak link as well as the previous stage greatly affected the rest of the circuit. At low input frequencies, decreasing the width of the weak link and increasing the width of the previous stage increased the MTTF, and decreasing the width of the metal actually lowered the MTTF.

Figure E1.1 The layout of a MUX and the relevant transistors

Figure E.1.2: The MTTF of the MUX for Various Changes and Input Frequencies

For weak link and previous stage changes, the weak link went from P5 to P1. Figure E.1.2 shows the MTTF data for the various changes. As stated earlier, for lower input frequencies, changing the metal size has a negative affect on the MTTF and increased power. Decreasing the size of the weak link as well as increasing the size of the previous stage had a positive effect on the MTTF, but decreasing the width of the weak link decreased power consumption while increased the width of the previous stage greatly increased power consumption. At higher input frequencies, the only change that improves upon the MTTF is decreasing the width of the weak link. Decreasing the size of the metal widths is ineffective.

Figure E.1.3: Comparison of the Power Consumption Between the Various Changes and Input Frequencies

E.2 IMPROVING A FULL ADDER

A similar procedure was performed on the Full Adder, but the results were drastically different. For the full adder, increasing the size of the previous stage worked the best for increasing the MTTF and also the metal actually had a factor in the MTTF. For all three changes the weak link remained fixed on a single transistor. Decreasing the width of the weak link as well as decreasing the width of the metal lines proved to be detrimental to the circuit at when run at higher input frequencies. At lower input frequencies, it is

shown that the MTTF has been improved for each of the methods mentioned with increasing the size of the previous stage being the most effective. At lower input frequencies, the electric field becomes the only influence on the MTTF since there are no self heating effects. As the input frequency increases, the improvements use more power than the unmodified circuit and increase the temperature by 0.1K.

Figure E.2.1: MTTF for different input frequencies for various modifications

This causes the electric field improvements via decreasing metal size and decreasing the weak link width to lose its effectiveness. At higher input frequencies, only by increasing the size of the previous stage does the MTTF of the circuit improve. Figure E.2.2 shows the power consumption of the circuit for various states. When run with the fastest input frequency, the unmodified circuit uses the least power. Increases the size of

the previous stage uses the most power but the electric field improvement still increases the MTTF of the full adder.

Figure E.2.2: Power Consumption of the Circuit for Various Changes and Input Frequencies

E.3 MAXIMUM IMPROVEMENT

The MTTF of the two circuits were maximized by strategically increasing the PMOSFETs of the entire gate and slightly decreasing the size of the weak link. Figures E.3.1 and E.3.2 show the results for the two circuits for the changes discussed. The data shows that the fix greatly improves the MTTF of the circuits, especially for when the circuits were run at a fast speed. Please note the differences in the axis for 25MHz (left graph) and for 400MHz (right graph). The normalized difference for the 25MHz case changes by 0.001 while the difference for the 400MHz case is 0.01. So there is a drastic improvement when the circuit is run faster over the unmodified circuit.

Figure E.3.1: MTF for the Full Adder Improvement.

Figure E.3.2: MTF for the Multiplexor Improvements

While the results of the improved fix are impressive, the correlation between MTF and power and performance are not so clear. With the full adder, the power consumption for the improved fix is higher than the other smaller fixes, but for performance aspect, the improved fix caused the full adder to perform worse than the other fixes. This is caused because of the extra load of the NMOSFETs. By increasing the sizes of all the PMOSFETs, all the NMOSFETs must drive a larger load causing all the rise times within the circuit to be longer which ultimately affects the output of the full adder. Again when measuring delay, the slowest output for all given inputs is chosen as the delay in this case. The size increase of the PMOSFETs also caused the increase in power. There are twenty-eight (28) transistors in the circuit, so increasing the size of half the total transistors will increase power. Using this kind of fix for a larger circuit could be prone to higher temperatures.

Figure E.3.4: Fixed vs. Power for the Full Adder

Figure E.3.5: Fixed vs. Delay for the Full Adder

The MUX is a smaller circuit, but the effects of increasing all the PMOSFETs had completely different effects. The fix described greatly increased the MTTF of the circuit with very little power and performance tradeoffs. Figures E.3.5 and E.3.6 show that MUX1 and MUX2 consumed slightly more power than the unmodified circuit and MUX2 also performed much better. The propagation delays within the circuit were not affected because the MUX is smaller than the full adder.

Figure E.3.6: Fixes vs. Power for the MUX

Figure E.3.6: Fixes vs. Delay for the MUX

APPENDIX F – LIST OF ACRONYMS

Al ₂ O ₃	Aluminum Oxide
DC	Direct Current
DRC	Design Rule Checker
EDA	Electronic Design Automation
EDIF	Electronic Design Interchange Format
FET, MOS, MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
IC	Integrated Circuit
KS	Koome-Stone
LEF/DEF	Library Exchange Format/Design Exchange
MOSIS	Metal Oxide Semiconductor Implementation
MTTF	Mean Time to Failure
MUX	Multiplexor
NMOS, NMOS	n-Channel MOSFET
OPL	Output Prediction Logic
PCB	Printed Circuit Board
PF	Poole-Frenkel
PMOS, PMOS	p-Channel MOSFET
PPA	Parallel-Prefix Adder
RC	Resistor-Capacitor Circuit
RCA	Ripple Carrier Adder
SRAM	Static RAM (Random Access Memory)
TDDDB	Time-Dependant Dielectric Breakdown
TiO	Titanium Oxide (?)
VHDL	VHSIC (Very High Speed Integrated Circuits)
VLSI	Very Large Scale Integration
ZrO	Zirconium Oxide

Table 7.1: List of Acronyms

APPENDIX G – LIST OF EQUATIONS

$$\tau = A \exp\left(\frac{E_a}{kT}\right) \exp(-\gamma E_f) \quad (2.4.1)$$

$$\tau = \frac{A \exp\left(\frac{E_a}{kT}\right) \exp\left(-\frac{\gamma}{E_f}\right)}{E_f^2} \quad (2.4.2)$$

$$\tau = \frac{Q_{BD}}{E_f} \exp\left(\frac{q[\Phi_B - \sqrt{\frac{qE_f}{\pi\epsilon_0\epsilon_\infty}}]}{kT}\right) \quad (2.4.3)$$

$$V_{ox} = V_g - V_p - V_s - V_{FlatBand} \quad [19] \quad (2.5.1)$$

$$a * e^{b * Ef} * e^{c * T} \quad (2.5.2)$$

$$MTTF = A * \exp(B * E_f + C * T) + D \quad (4.4.1)$$

$$P = \frac{1}{2} f_C C_L V_{DD}^2 \quad (5.4.1)$$

$$\begin{aligned} C_{out} &= C(A + B) + AB, \\ S &= !A!BC + !AB!C + A!B!C + ABC \end{aligned} \quad (B.1)$$

$$S = A \text{ xor } B \text{ xor } C \quad (B.2)$$

BIBLIOGRAPHY

- [1] Hauschildt, M., M. Gall, and S. Thrasher. "Analysis of Electromigration Statistics for Cu Interconnects." *Applied Physics Letters* 88 (2006).
- [2] Hiroyuki, Abe, Sasagawa Kazuhiko, and Saka Masumi. "Electromigration Failure of Metal Lines." *International Journal of Fracture* (2006): 219-240.
- [3] Dumin, David J, et al. "High Field Related Thin Oxide Wearout and Breakdown." *IEEE Transactions on Electron Devices* Vol. 42 (1995).
- [4] Wu, Jie, Rosenbaum, Elyse. "Gate Oxide Reliability Under ESD-Like Pulse Stress." *IEEE Transactions on Electron Devices* Vol. 51 (2004).
- [5] Zeghbroeck, B. Van. "Chapter 7: MOS Field-Effect-Transistors." *Principles of Semiconductor Devices*. 2004. <http://ece-www.colorado.edu/~bart/book/book/Chapter7/ch7_7.htm>.
- [6] Subramoniam, R. A Statistical Model of Oxide Breakdown Based on a Physical Description of Wearout. *Proceedings of IEEE International Electron Devices Meeting*, 13 Dec. 1992, Electron Devices Soc. IEEE.
- [7] Lee, J.c., Chen Ih-Chin, and Hu Chenming. "Modeling and Characterization of Gate Oxide Reliability." *IEEE Transactions on Electron Devices* 35 (1998).
- [8] Henderson, Christopher L. "Time Dependent Dielectric Breakdown." *Semiconductor Reliability*. 2002. <<http://www.semitracks.com/manuals/12.pdf>>.
- [9] Allers, K.-H. "Prediction of Dielectric Reliability From I–V Characteristics: Poole-Frenkel Conduction Mechanism Leading to Root(E) Model for Silicon Nitride MIM Capacitor." *Microelectronics Reliability* 44 (2003): 411-423.
- [10] Noguchi, Junji. "Dominant Factors in TDDB Degradation of Cu Interconnects." *IEEE Transactions on Electron Devices*, 52 (2005): 1743-1750.

- [11] Shiono, N, and M Itsumi. "A Lifetime Projection Method Using Series Model and Acceleration Factors for TDDB Failures of Thin Gate Oxides." 31st Annual Proceedings. Reliability Physics 1993 (1993): 1-6.
- [12] Rodder, M. "Oxide Thickness Dependence of Inverter Delay and Device Reliability for 0.25um CMOS Technology." International Electron Devices Meeting 1993. Technical Digest (1993).
- [13] Sankaranarayanan, Karthik. "[23] Multiple Core Chip Thermal Model." The HotSpot Archives. 4 Feb. 2008. <<http://www.cs.virginia.edu/pipermail/HotSpot/>>.
- [14] Rodder, M., S. Iyer, S. Aur, A. Chatterjee, J. McKee, R. Chapman, and I. C. Chen. "Oxide Thickness Dependence of Inverter Delay and Device Reliability for 0.25um CMOS Technology." Electron Devices Meeting, 1993. IEDM '93. Technical Digest., International (1993): 879-82.
- [15] Suehle, John S., and Prasad Chaparala. "Low Electric Field Breakdown of Thin SiO Films Under Static and Dynamic Stress." Electron Devices, IEEE Transactions on Electron Devices 44 (2007): 801-08.
- [16] "Temperature Coefficient of Resistance." Georgia State University. 05 Apr. 2009 <<http://hyperphysics.phy-astr.gsu.edu/hbase/electric/restmp.html>>.
- [17] Vincent, E., N. Revil, C. Papadas, and G. Ghibaudo. "Electric Field Dependence of TDDB Activation Energy in Ultrathin Oxides." Reliability of Electron Devices, Failure Physics and Analysis, 1996. Proceedings of the 7th European Symposium (1996): 1643-646.
- [18] Yassine, Abdullah, H. E. Nariman, and Kola Olasupo. "Field and Temperature Dependence of TDDB of Ultrathin Gate Oxide." IEEE Electron Device Letters 20 (1999): 390-92.
- [19] Yassine, Abdullah M., H. E. Nariman, Michael McBride, Mirac Uzer, and Kola R. Olasupo. "Time Dependent Breakdown of Ultrathin Gate Oxide." IEEE Transactions on Electron Devices 47 (2000): 1416-420.

- [20] "Design of VLSI Systems - Chapter 4." Welcome to LSI. 18 Mar. 2009
<<http://lsiwww.epfl.ch/LSI2001/teaching/webcourse/ch04/ch04.html>>.
- [21] Rubin, Steven M. "Electric™ User's Manual, version 8.08." Electric User's Manual, version 8.08. 9 Dec. 2008. <<http://www.staticfreesoft.com/>>.
- [22] Nave, Carl R. "Resistance: Temperature Coefficient." HyperPhysics. 2001. Georgia State University. <<http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html>>.
- [23] "HotSpot 5.0 Temperature Modeling Tool." *LAVA Lab forward page*. Ed. Kevin Skadron et al. National Science Foundation, 24 July 2009.
<<http://lava.cs.virginia.edu/HotSpot/>>.
- [24] "Linear Technology - Design Simulation and Device Models." *Linear Technology - Linear Home Page*. Linear Technology, 2007.
<<http://www.linear.com/designtools/software/>>.
- [25] "E158 Spring 2007 MIPS Project." The Harvey Mudd College. Web.
<<http://www4.hmc.edu:8001/Engineering/158/07/project/index.html/>>.