ABSTRACT

HARRIS, THEODORE ROBERT. Electrothermal Analysis of Three-Dimensional Integrated Circuits. (Under the direction of Dr. Michael B. Steer.)

Transient electro-thermal simulation of a three dimensional integrated circuit (3DIC) is reported that uses a cell-based simulation to provide a selected transistor thermal profile while providing advantages of hierarchical simulation. Due to CPU and memory limitations, full transistor electro-thermal simulations on a useful scale are not possible. Standard cells are considered on a per-instance basis and modeled with electro-thermal macro-models developed in a multi-physics simulator. Simulations are compared favorably to measurements for a token-generating 3DIC clocking at a maximum of 1 GHz. The 3DIC, which is composed of 9 by 3 layers of repetitive frequency multipliers and dividers, was fabricated with the Massachusetts Institute of Technology Lincoln Laboratory (MITLL) 3DIC process. Measurements indicated a linear rise in temperature of the active areas over a range of applied background ambient temperatures. An average of 7.5 K change in temperature was measured across dense areas of circuitry. For thermal simulation, the physical characteristics of the 3DIC were extracted from flattened OpenAccess layout files. Material parameters, connections, and geometries were considered in order to create a more physically accurate resistive thermal mesh. Physical thermal networks extracted with resolutions of 10 µm and 5 µm connect thermal terminals of the electrothermal macromodel cell elements to active layers yielding temporal and spatial simulated dynamic thermal results in three dimensions. Coupled with model-order reduction techniques, hierarchical dynamic electrothermal simulation of large 3DICs is shown to be tractable, yielding spatial and temporal selected transistor-level thermal profiles.
Electrothermal Analysis of Three-Dimensional Integrated Circuits

by
Theodore Robert Harris

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2011

APPROVED BY:

Dr. David Schurig
Dr. Henry E. Schaffer

Dr. W. Rhett Davis
Co-Chair of Advisory Committee

Dr. Michael B. Steer
Chair of Advisory Committee
DEDICATION

To my parents . . .
BIOGRAPHY

Theodore Robert Harris was born in Washington, DC. He received his B.S. degree in Electrical Engineering from the University of North Carolina at Charlotte in 2005 and his M.S. degree in Electrical Engineering from North Carolina State University focusing in nanoelectronics and photonics in 2007. Since 2007, he has worked toward the Ph.D. degree as a research assistant at North Carolina State University with the Lampe Distinguished Professor of Electrical Engineering, Dr. Michael Steer.

He interned at the National Institute of Standards and Technology in the Material Science Laboratory to work on optical electronic contacts for LEDs under an NSF grant in 2004. From 2005 to 2007 he worked as a photolithography engineer in the NCSU Nanofabrication Facility.

Mr. Harris’s major areas of interest are thermal effects, solid state devices, material science, VLSI processing, MEMS, nanodevices, electromagnetics, lasers, and optics. He enjoys the outdoors, camping, hiking, snowboarding, and speleology.
ACKNOWLEDGEMENTS

I want to specifically thank the Lampe Distinguished Professor of Electrical and Computer Engineering, Michael Steer, my tireless adviser. Dr. Steer has been incredibly helpful and truly an inspiration on both professional and personal levels. He was always available to advise, including nights and weekends.

I am grateful to the rest of my advisory committee, Drs. W. Rhett Davis, David Schurig, and Henry E. Schaffer. They have all helped me in different ways over the course of this work. I want to acknowledge the staff and faculty of the Electrical and Computer Engineering Department for their support.

Thank you to my fellow grad students, with whom I have spent so much time Dr. Jonathan R. Wilkerson, Dr. Greg Mazzaro, Chris Saunders, and Glen Garner. I also want to express my sincere gratitude to Shivam Priyadarshi and Samson Melemed for all of their help with fREEDA and electrothermal modeling. Thank you to everyone else past and present in room 438 of MRC. Additionally I thank Drs. Paolo Batoni, Ed Stokes, and especially Dr. Stephen Bobbio over at UNC Charlotte, and my high school friend Dr. Mayank Varia at MIT.

Lastly and with the most gravity, I am grateful to my parents, Ted and Barbara Harris, for their love and support. They have encouraged my education, instilled honesty, and nurtured a predilection to science. I could never ask for better parents.
TABLE OF CONTENTS

List of Figures ................................................................. viii

List of Tables ................................................................. x

Chapter 1 Introduction ....................................................... 1
  1.1 Overview ................................................................. 1
  1.2 Motivations .............................................................. 2
    1.2.1 Objective: Full Fidelity Transient Electrothermal Simulation of a 3DIC ........................................... 3
  1.3 Original Contributions ............................................... 4
    1.3.1 Cell-based Approach to Electrothermal Simulation ................................................................. 4
    1.3.2 State Variable Condensation of Thermal Networks ................................................................. 4
    1.3.3 Design and Implementation of Electrothermal Elements ........................................................... 5
    1.3.4 Electrothermal Modeling of an Interconnect ................................................................. 5
    1.3.5 Design of Experiment and Validation of Simulation ................................................................. 5
    1.3.6 Analysis Flow for Full Electrothermal Simulation ................................................................. 6
    1.3.7 Time Compaction for Rapid Electrothermal Simulation ........................................................... 6
    1.3.8 Transient Electrothermal Simulation of a 3DIC ................................................................. 7
  1.4 Dissertation Outline ................................................... 7
  1.5 Published Work ........................................................ 8
    1.5.1 Conference Papers ................................................ 8
    1.5.2 Unpublished Work ................................................ 9

Chapter 2 Modeling Thermal Effects in Electronics ...................... 10
  2.1 Introduction ............................................................ 10
  2.2 Dynamic and Static Power Calculation in an IC ....................... 11
  2.3 Heat Transfer ............................................................ 14
    2.3.1 Conduction ......................................................... 14
  2.4 Interconnect Destruction by Electron Migration ......................... 16
  2.5 Long Tail Effects and Frequency Dependence .......................... 18
  2.6 Circuit-Based Simulation .............................................. 19
  2.7 Prior Work in Thermal Modeling ..................................... 20
    2.7.1 Finite Element and Difference Methods ................................................................. 20
    2.7.2 Direct Method ..................................................... 22
    2.7.3 Batty Model ....................................................... 23
  2.8 3DICs ................................................................. 24
    2.8.1 Chip-Level Stacking ............................................... 25
    2.8.2 Transistor-Level Stacking ......................................... 26
    2.8.3 Wafer-Level Stacking ............................................... 26
    2.8.4 Wafer Bonding .................................................... 27
  2.9 Summary ............................................................... 29
<table>
<thead>
<tr>
<th>Chapter 3 Experimental Setup for 3DIC and Test Board</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>30</td>
</tr>
<tr>
<td>3.2 Design and Functionality of the 3DIC</td>
<td>31</td>
</tr>
<tr>
<td>3.2.1 Block Diagram of Floorplan and Layout</td>
<td>31</td>
</tr>
<tr>
<td>3.3 Design of the Test Board</td>
<td>32</td>
</tr>
<tr>
<td>3.4 Through-PCB Heatsink Implementation</td>
<td>37</td>
</tr>
<tr>
<td>3.5 Mechanica Verification</td>
<td>43</td>
</tr>
<tr>
<td>3.6 Conclusions</td>
<td>45</td>
</tr>
<tr>
<td>Chapter 4 Electrothermal Simulation of a 3DIC</td>
<td>46</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>46</td>
</tr>
<tr>
<td>4.2 Simulation Environment</td>
<td>46</td>
</tr>
<tr>
<td>4.3 The RC cell model and expandability</td>
<td>48</td>
</tr>
<tr>
<td>4.3.1 Thermal Capacitance Calculation</td>
<td>51</td>
</tr>
<tr>
<td>4.3.2 Capacitive Scaling</td>
<td>52</td>
</tr>
<tr>
<td>4.4 Workflow</td>
<td>53</td>
</tr>
<tr>
<td>4.5 Computation of Power Numbers</td>
<td>55</td>
</tr>
<tr>
<td>4.6 Electrothermal Simulation</td>
<td>55</td>
</tr>
<tr>
<td>4.7 Extracted RC Thermal Network Approximation</td>
<td>59</td>
</tr>
<tr>
<td>4.7.1 Procedure for Thermal Netlist Extraction</td>
<td>60</td>
</tr>
<tr>
<td>4.7.2 Netlist Inclusion</td>
<td>63</td>
</tr>
<tr>
<td>4.8 Complete Electrical-only Simulation</td>
<td>65</td>
</tr>
<tr>
<td>4.9 Complete Electrical and Thermal Resistive Simulation</td>
<td>70</td>
</tr>
<tr>
<td>4.10 Hotspot Dynamic Electrothermal Simulation</td>
<td>72</td>
</tr>
<tr>
<td>4.10.1 Hotspot Dynamic Electrothermal Simulation Results</td>
<td>73</td>
</tr>
<tr>
<td>4.10.2 Comparison of Thermal Network Resolutions</td>
<td>79</td>
</tr>
<tr>
<td>4.11 Transient Computation</td>
<td>81</td>
</tr>
<tr>
<td>4.12 Summary</td>
<td>84</td>
</tr>
<tr>
<td>Chapter 5 Electrothermal Elements</td>
<td>85</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>85</td>
</tr>
<tr>
<td>5.2 Suite of Electrothermal Elements for fREEDA</td>
<td>86</td>
</tr>
<tr>
<td>5.2.1 RC-Based Electro-Thermal Elements</td>
<td>87</td>
</tr>
<tr>
<td>5.2.2 ThermalBlockRC</td>
<td>87</td>
</tr>
<tr>
<td>5.2.3 ThermalBlockRC3D</td>
<td>88</td>
</tr>
<tr>
<td>5.2.4 InterconnectRTSH</td>
<td>90</td>
</tr>
<tr>
<td>5.3 Batty Model Based Electro-Thermal Elements</td>
<td>95</td>
</tr>
<tr>
<td>5.3.1 ThermalBlock</td>
<td>95</td>
</tr>
<tr>
<td>5.3.2 ThermalHeatsink</td>
<td>96</td>
</tr>
<tr>
<td>5.3.3 ThermalShunt</td>
<td>97</td>
</tr>
<tr>
<td>5.4 Comparison of the Batty Model with an RC Thermal Model</td>
<td>98</td>
</tr>
<tr>
<td>5.5 Summary</td>
<td>104</td>
</tr>
</tbody>
</table>
# List of Figures

| Figure 2.1 | MITLL 3DIC Process Stackup | 28 |
| Figure 3.1 | 3DIC Floorplan with Thermal Vias | 32 |
| Figure 3.2 | Layout with Division of Cells | 33 |
| Figure 3.3 | Block Diagram of Frequency Chain | 34 |
| Figure 3.4 | PCB Test Board Layout | 35 |
| Figure 3.5 | PCB Test Board Layout | 36 |
| Figure 3.6 | Pinout Diagram of the Package | 38 |
| Figure 3.7 | 3DIC Bonding Diagram | 38 |
| Figure 3.8 | Microscopic Image of the Bonded Die | 39 |
| Figure 3.9 | Top View Technical Drawing of Testboard | 41 |
| Figure 3.10 | Front View Technical Drawing of Testboard | 42 |
| Figure 3.11 | Mechanica Simulation of 3DIC Die and Heatsink Post | 44 |
| Figure 3.12 | Mechanica Simulation of test board | 44 |
| Figure 4.1 | Transient Simulation | 50 |
| Figure 4.2 | Transient Simulation | 52 |
| Figure 4.3 | Electrothermal Simulation Workflow | 54 |
| Figure 4.4 | Frequency Multiplier ET Macromodel | 58 |
| Figure 4.5 | Frequency Divider ET Macromodel | 58 |
| Figure 4.6 | Numbering of Resistors in Three Dimensions | 59 |
| Figure 4.7 | Capacitive Thermal Grid | 61 |
| Figure 4.8 | Combined Via and Backmetal Layers | 62 |
| Figure 4.9 | Resistive 3DIC Stackup Layers | 63 |
| Figure 4.10 | Electrical output of the First Multiplier | 66 |
| Figure 4.11 | Electrical output of the Second Multiplier | 66 |
| Figure 4.12 | Electrical output of the 7th Multiplier | 67 |
| Figure 4.13 | Electrical output of the 14th Multiplier | 67 |
| Figure 4.14 | Electrical output of the First Divider | 68 |
| Figure 4.15 | Electrical output of the Second Divider | 68 |
| Figure 4.16 | Electrical output of the Eighth Divider | 69 |
| Figure 4.17 | Electrical output of the Tenth Divider | 69 |
| Figure 4.18 | Calculation of Extracted Thermal Values | 70 |
| Figure 4.19 | Full Tile Electrothermal Simulation | 71 |
| Figure 4.20 | Transient Simulation | 74 |
| Figure 4.21 | Time Evolution Temperature Profile | 75 |
| Figure 4.22 | Time Evolution Temperature Profile | 76 |
| Figure 4.23 | Three Dimensional Temperature Profile of Layers | 77 |
| Figure 4.24 | Time Evolution Temperature Profile | 78 |
| Figure 4.25 | Resolution Comparison of Thermal Results | 80 |
| Figure 4.26 | Internal RC Grid Reduction | 81 |
LIST OF TABLES

Table 2.1 Electrical to Thermal Allegory ........................................ 20
Table 4.1 Frequencies of Cell Transition Nodes, Simulated .................... 56
Table 4.2 Powers of 3DIC Cells, Simulated ...................................... 57
Table 4.3 fREEDA Transient Electrothermal Simulation Times ............... 83
Table 6.1 Measured Frequencies of tile Electrical Output ....................... 110
Chapter 1

Introduction

1.1 Overview

Over half of all integrated circuit failures of 2DICs are due to thermal issues within the die [1]. Of those failures, the metal interconnects are typically the weak point. This research focuses on a method for the analysis of three-dimensional integrated circuits (3DICs) yielding spatial and temporal dynamic thermal profiles. Several approaches to simulating thermal effects have been defined by industry, including finite element analysis. Worst-case operating temperatures have traditionally been used in electrical simulations, with the effect being that temperatures are over estimated and design performance is compromised. Combined with design back-off such that the margin between average operating temperature and critical temperatures is sufficiently wide, design of integrated circuits can proceed with only minimal attention to dynamic and spatial thermal concentrations. Excessive conservatism in chip layout has been universally observed with traditional approaches to thermal issues. With the advent of 3DICs, simple layout experience for avoiding thermal issues is no longer possible due to improvements in silicon manufacturing technology that increases complexity. Maximum temperatures in the 3DIC must be determined with precision so that thermal margins typically used can be reduced and the full potential of 3DICs may be exploited.
3DIC technology will have widespread use in combining memory and logic. However, the expansion of solid-state circuits into three dimensions overlays heat-sensitive memory circuits with heat-producing logic, creating an extreme thermal environment. Thermal effects have become one of the major inhibitors of design which inhibits the widespread adoption of 3DIC technology. It is important that the design considers thermal effects within a more severe thermal environment.

While a 3DIC may contain billions of heat-producing elements including junction and channel transportation, and resistance of interconnects, critical hot spots may affect just a few elements. Full fidelity analysis requires that dynamic or transient heat and temperature of individual interconnects and transistors be evaluated, but the vast range of scale means that it is not possible to model every element simultaneously and dynamically. The multi-scale solution involves static finite-element thermal modeling of the whole chip and package establishing thermal boundary conditions at a scale of a millimeter-cubed. Cell-based models have been implemented to simplify repeated design structures with known heat flux. A region is modeled dynamically by combining electrothermal active elements such as transistors or standard cell elements and thermal resistance and capacitance to simulate bulk materials.

The research presented in this dissertation focuses on a method for designing a 3DIC while mitigating thermal failure. The result is a realistic analysis of thermal 3DIC macromodels, which provide an important tool for efficient cutting-edge design. A greater certainty of system functionality can be achieved.

1.2 Motivations

Current efforts in electrothermal simulation have been limited to mechanical models that consider static heat flux and material properties in finite element-bases thermal analysis tools such as Mechanica™ [2]. However, analysis of systems is limited to about 10 million objects independent of the field. This limitation is a result of numerical ill-conditioning of matrix-like solution strategies. It may be possible to increase the limit of the number of elements by a
factor of ten over the next few years as computation ability increases, but this is a long was from being in a position to model the performance of ICs at the transistor level. Presently, analyses should be limited to the order of 100,000 objects so that computation times remain manageable. The very nature of the IC design process requires rapid thermal evaluation in response to incremental changes. This process may also be used to evaluate contingency scenarios. Incremental analysis plays a key role in finding solutions more efficiently, standing in contrast to the inefficiency of a full, computationally expensive reanalysis of the entire system. This is why a cell-based approach has been deemed necessary. The fundamental problem of electrothermal modeling is one of handling multiple scales, which is addressed in this research.

Different simulation levels can be specified for the needed fidelity in areas of interest. In order to drastically reduce computation over other methods, iREEDA TM features variable state reduction of RC networks in the .tran2 analysis by matrix condensation. A large mesh of linear elements becomes only as complicated as the external nodes of the circuit. In addition, predetermined macro models have been developed for many common VLSI standard cells, further reducing internal state variables and increasing computation speed. True flexibility is granted by concurrently running transistor level modeling with coarse block level modeling.

1.2.1 Objective: Full Fidelity Transient Electrothermal Simulation of a 3DIC

This research had four main goals:

• Determining when undesirable thermal effects will occur in an integrated circuit.

• Electrothermal modeling a volume of heat-producing elements in three dimensions.

• Developing thermal measurement methods to verify transient simulations.

• Enabling engineers to implement improved designs once thermal information is learned.
1.3 Original Contributions

The pursuit of these goals generated various research milestones along the way. These milestones, the author’s contributions to the field of electrothermal simulation, and computer aided design are summarized below.

1.3.1 Cell-based Approach to Electrothermal Simulation

Transistor-level electrothermal simulations are accurate, but exceedingly time consuming to such a degree that they are arguably not feasible for large circuits. Direct method approaches combine electrothermally-aware simulations with physically-aware thermal simulations, joined by thermal terminals. By using the cell-based approach described in Section 4.6, circuit hierarchy can be exploited to advantage in simulation, due to the fact that many subcircuits are reused in other circuits, and subcircuits are reused many times within the same circuit. These circuits maintain the same electrical functionality, and consistently cover the same physical areas, and therefore maintains similar thermal profiles. While these facts can be used to reduce simulation time, in the work presented here, these have been used to reduce model development overhead by use of macromodels.

1.3.2 State Variable Condensation of Thermal Networks

The method of simulating the thermal characteristics of materials presented in this dissertation is based on an RC electrical circuit analogy. The application of RC gridding of thermal materials results in a range of large to very high quantities of resistor and capacitor elements in simulator netlists. With regards to simulation time, the number of internal linear circuit elements has been made negligible by the use of a Kron-like matrix condensation and the reduction of unnecessary circuit state variables that was used by Christofferson [3]. This type of state variable reduction is particularly useful and important to thermal analysis. State variable reduction has successfully been applied to electrothermal circuits, and it will be shown in Section 4.3 that simulation times are nearly equivalent for circuits of vastly different sizes.
1.3.3 Design and Implementation of Electrothermal Elements

A number of electrothermal elements were designed or improved and implemented for the \textsc{fREEDA} simulator package. These elements include passive \texttt{ThermalBlockRC3D}, \texttt{InterconnectRTSH}, as well as the active devices: \texttt{ComsinvT}, and \texttt{CmosnandT}. The \texttt{ThermalBlockRC3D} is presented in Section 5.2.3, and it is compared to a Batty-based model which solves the heat equation in Fourier-space in Section 5.4.

1.3.4 Electrothermal Modeling of an Interconnect

Particular attention has been given to the metal interconnects that join active semiconductor elements in transistors due to their importance in signal integrity and likelihood of failure. A step towards merging the physical layout dimensions with macromodel simulation has been made by including the ability of the interconnect to calculate its own electrical and thermal attributes given its $xy$ coordinate information. This research is covered in Section 5.2.4.

1.3.5 Design of Experiment and Validation of Simulation

A specialized “through-PCB” heatsink was designed to interface with the back side of an IC package. A poorly designed heatsink can be detrimental to thermal imaging. A novel technique for attaching a heatsink through the PCB board is described in Section 3.4. Once developed, the heatsink was successfully demonstrated to work properly through simulation verification, which is described in Section 3.5.

A 3DIC has been thermally imaged at a high resolution and without the use of dispersive and error-inducing thermal ink, allowing comparison of extremely accurate measurements to simulations. Thermal measurement has been expanded to include transient measurements as the circuits are heated from the power-on event. These measurements were used to verify the correctly implemented transient electrothermal simulation. These contributions are presented in Sections 6.3.1 and 6.3.3.
1.3.6 Analysis Flow for Full Electrothermal Simulation

Electrical and thermal attributes function in two distinct domains, but communicate and affect each other. Many different approaches exist as to how to connect these two domains in a simulator environment. In the models presented in this dissertation, electrical and thermal networks were connected by use of the thermal terminals of electrothermal elements and macromodels.

A method of transcoding geometric data derived from the physical layout to a simulator-suitable format was designed and implemented. A complete start to finish work flow for electrothermal simulation is described in Section 4.7, which can also be applied to electrothermal simulation of standard IC designs.

1.3.7 Time Compaction for Rapid Electrothermal Simulation

A major difficulty with electrothermal simulation lies in the fact that thermal events occur on a much slower timescale than electrical events. While this research does present simulations which find that local thermal events do occur over tens of picoseconds; on an appreciable scale, a warm up transient from power-on to steady-state is on the order of milliseconds. This is due to the fact the thermal capacitance (which relates degrees of freedom of a material) must be filled by kinetic means. Heat is conducted by a diffusion of the vibrations and collisions within said degrees of freedom of motion.

To simplify this problem, thereby greatly reducing required computation time, the quantity of thermal capacitance per unit cell volume was artificially reduced to produce a time-scaled simulation. This results in much faster dynamic electrothermal simulation. This time compaction is described in Section 4.3. This made much longer simulations feasible. Furthermore the effect of thermal capacitance reduction scales the rise times of transients proportionally. Simulation results of capacitive time scaling are presented in Section 4.3.2
1.3.8 Transient Electrothermal Simulation of a 3DIC

The spatial heating of all layers in the 3DIC and the interactions between these layers are captured in transient simulations over time. For the first time, spatial and temporal results are presented in Section 4.10.1. Simulations are truly three-dimensional, with vertical as well as horizontal gridding. The volumetric simulation ability takes advantage of the fact that the technique has been applied to a 3DIC, which contains heat-producing elements on three separate layers.

1.4 Dissertation Outline

Chapter 2 of this dissertation presents a literature review of 3DIC technology and electrothermal modeling. The purpose of 3DICs and their fabrication is reviewed. The topics related to electrothermal modeling covered include thermal effects in electronics, dynamic electrothermal analysis, the long tail effect, interconnect destruction by electromigration and frequency dependence.

Chapter 3 explains the experimental setup of the 3DIC test board used in this research. The functionality and design of the integrated circuit is described, after which the technical details of the test board design and fabrication are recounted.

Chapter 4 explores the electrothermal simulation of the circuit described in Chapter 2. It details the creation of the macromodels, and netlists. A procedure for thermal netlist extraction from layout files is presented, as well as the method used for electrothermal to thermal netlists connectivity in a logical fashion.

Chapter 5 presents other electrothermal models that were simulated. These models demonstrate value to the CAD engineering community, but were not related to the 3DIC. A locationally aware self-heating thermal interconnect, a three-dimensional RC thermal material element, and other active logical elements are presented.

Chapter 6 presents and compares the electrical and thermal measurements to simulated
Chapter 7 contains a summary of the research performed and lists the significant results of this work.

1.5 Published Work

The papers listed here are based on the work presented in this dissertation.

1.5.1 Conference Papers


1.5.2 Unpublished Work


Chapter 2

Modeling Thermal Effects in Electronics

2.1 Introduction

This chapter reviews the knowledge that provides the context for the solutions presented in later chapters concerning electrothermal modeling. The topics of 3DIC technology, thermal effects in electronics, and dynamic analysis are closely related and increasingly of interest for expanding current technological boundaries. A basic introduction and points are made about each.

Modern VLSI circuits contain many millions of transistors, with each transistor consisting of its own varied assortment of materials and three-dimensional structures. The latest Intel processor released at the publication of this dissertation, known as Sandy Bridge, contains just under a billion discrete transistors [4]. The power density of the Pentium 4 is 10–15 W/cm², which nearly exceeds the limits of air cooling [5]. Power management rather than thermal management has been the area of focus on mobile devices, but is beneficial for thermal management as well. Clock scaling and code morphing were pioneered by Transmeta™, but processor load swapping utilizing different cores has not been necessary up to this point in
time [6]. There is a great deal of difficulty concerning thermal simulation circuits of this scale, nonetheless simulation is important because thermal effects constitute the majority of failure modes as well as the major contributor to circuit aging. Newer fabrication technologies such as SOI and especially 3DIC architectures greatly magnify thermal issues due to the fact that power dissipating elements are more densely packed and physically further from the heatsink. Furthermore they are separated by higher thermally resistive materials. A number of approaches and techniques have been applied to electrothermal simulation and are covered in Section 2.7.

The basics of the source of heat within an IC are presented in Section 2.2. Once heat is produced within an IC, it will flow to another region. The mechanics of this heat transfer are covered in Section 2.3. In particular, the mode of surface contact conduction is explored in Section 2.3.1 by noting the partial differential heat equation. Section 2.9 summarizes the lessons learned from this chapter.

### 2.2 Dynamic and Static Power Calculation in an IC

At the most basic level, power in an IC is the sum of both dynamic and static powers. Dynamic power includes glitches, logic activity, and short circuit current, while static power includes leakage. The power consumption and dissipation of an IC is closely related to its switching rate, but leakage current is playing an increasing role in recent technology trends. It has been shown that 40% or higher of a chip’s power consumption in the 90 nm process node is due to leakage current [7]. The components of power in an IC are,

\[ P_{\text{total}} = P_{\text{dyn}} + P_{\text{sc}} + P_{\text{stat}}, \] (2.1)

where the total power is composed of the dynamic, short circuit, and static powers. The rest of this section shows how \( P_{\text{total}} \) is derived and estimated. Many of the smaller components of \( P_{\text{total}} \) are estimations; every circuit is different and some small leakage currents can only be approximated. Power is used and dissipated whenever current flows. CMOS devices consist
generally of charging and discharging capacitors while in operation, in addition to leakages and short circuit current. It is the transfer of charge to capacitors which dissipates power, the frequency of which is related to the speed of switching. For example, where calculating the power of an inverter, the most basic building block of CMOS logic, the charging of a capacitor is considered. The amount of current required to charge a capacitor is equivalent to,

\[ i(t) = \frac{dq}{dt} = C \frac{dv}{dt} = \frac{V - v(t)}{R}, \]  

(2.2)

where \( V \) is the voltage applied and eventually the final voltage, if held constant. This current rearranged in terms of voltage and integrated to remove differentials is,

\[ \int \frac{dv}{V - v(t)} dt = \int \frac{1}{RC} dt. \]  

(2.3)

Integrated and solved in terms of current, the function becomes,

\[ i(t) = CVe^{-\frac{t}{RC}}. \]  

(2.4)

To find the energy of the transition and power dissipated by the energy, a double integral over time of the above equation in terms of \( R \) is performed to yield,

\[ E = \frac{1}{2} CV^2, \]  

(2.5)

which is also equivalent to the energy in the stored MOS capacitor. To move from energy to the power dissipated during transition on a clocked circuit, the switching frequency must be incorporated with the energy of the transition by,

\[ P_{dyn} = E_{trans} \alpha f_{ck}. \]  

(2.6)

Similarly, short circuit power assuming equal rise and fall times is found by,
\[ P_{sc} = \alpha f_{ck} E_{sc}, \]  

(2.7)

where \( \alpha \) is activity, a term based on the switching statistic of the particular circuit [8]. One of the results of this is that the short circuit power increases with longer transition rise and fall times, and due to the fanout of digital circuits, it is beneficial if output rise and fall times are less than input rise and fall times.

The leakage power is derived from the small amount of current able to leak through the gate oxide, \( I_G \), drain to source punch through, \( I_{PT} \), reverse bias p-n junction conduction, \( I_D \), subthreshold current, \( I_{sub} \), and gate induced drain leakage, \( I_{GIDL} \) [9]. As process dimensions decrease with technology iterations, \( I_G \) becomes worse with Fowler–Nordheim tunneling through the thin oxide becomes possible. The major components are \( I_{sub} \) and \( I_D \) only, which are found by,

\[ I_{leakage} = I_{sub} + I_{DS} = \mu_0 C_{ox} (W/L) V_t^2. \]  

(2.8)

Equation 2.6 and Equation 2.8 are combined for the most complete dynamic power equation [10]:

\[ P_{dyn} = (V_{sd} I_{sd}) + (2f_{ck} C V_{dd}) + I_{leakage}, \]  

(2.9)

where \( V_{sd} \) and \( I_{sd} \) are the voltage applied across the source and drain and the current which flows from source to drain, respectively. \( V_{dd} \) is the process voltage supply rail. Another useful relation is found in the equation for the temperature of the junction in a semiconductor,

\[ T_j = T_{ambient} + (P_D R_{\theta JA}), \]  

(2.10)

where \( T_A \) is the ambient temperature, \( R_{\theta JA} \) is the junction to ambient thermal resistance which is dependent on area (°C/W), and \( P_D \) is power dissipation in the package [11].
2.3 Heat Transfer

Heat is a measure of the kinetic energy of particles. In metals, heat is mainly attributed to the movement of free electrons, while in dielectrics, heat is composed of phonons. Phonons are quasi-particles which represent quantized lattice vibration. Heat may be transferred by one or a combination of three modes; convection, radiation, and conduction. This research focuses on conduction due to the small scale of integrated circuits, proximity to the heatsink, and lack of surface area of 3DICs. Other modes of transfer make up a negligible amount of temperature difference.

Convection is the bulk fluid movement of particles from the body to the universe or to another body. In truth convection is a complicated form of conduction that also contains or contributes to a macroscopic movement of the transfer medium. Convection is bound by the second law of thermodynamics, which states, “No process is possible whose sole result is the transfer of heat from a body of lower temperature to a body of higher temperature,” [12], and Newton’s law of cooling which states that, “the rate of heat transfer of a body is proportional to the difference between the body and the surroundings.” Radiation is the final form of heat transfer wherein the kinetic movement of electrons is transmogrified into electromagnetic waves through emission. All objects above absolute zero radiate, and radiation is possible through any medium, including vacuum.

2.3.1 Conduction

All materials have thermal properties of specific heat capacitance and thermal conductivity, $k$, measured in watts per kelvin-meter $W\cdot (K\cdot m)^{-1}$. In thermal modeling, the inverse of $k$, thermal resistance, measured in kelvin-meters per watt $(K\cdot m \cdot W^{-1})$ is generally taken to maintain a better allegory to the electrical circuit concept of resistance. When thermal resistances occur in series they are additive. This is the case in the following structure: a thermal source in an active layer of semiconductor joined to a silicon handle and finally coupled to a heatsink by way of a a pressure contact. Each stage described in this series can be considered as an
independent additive thermal resistance. When thermal resistances occur in parallel, as is the case of multiple vertical metal thermal vias in an IC surrounded by silicon dioxide, the conductances add.

Heat capacity, $C$, is the amount of heat $Q$, in Joules (J), needed to change the temperature of a body by a given amount. $C$ is related mathematically by

$$C = \frac{\Delta Q}{\Delta T}.$$  \hspace{1cm} (2.11)

Heat capacity is extrinsic, depending on the mass of material present, whereas specific heat capacity is intrinsic with the units,

$$[C] = \frac{J}{kg \cdot K}.$$ \hspace{1cm} (2.12)

The specific heat capacity is the focus of most simulation schemes due to the fact that mass of material is related to the model by some other means, such as by increasing the grid size. Heat capacity is not linear over a wide range of temperatures for a given material. It should be noted that heat capacity differs in solids from heat capacity in gases, which adhere to the ideal gas law. Since temperature is the average kinetic energy of particles in a system, heat capacity depends on the degrees of freedom that are available to the particles in the system. The degree of freedom between particles is determined by the number and type of bonds, as well as by the bond lengths and strengths. In solid state electronics, the general materials of concern are metals and ceramics such as silicates. Properties of the materials such as phase, crystal structure, crystal orientation, and to what degree materials are polycrystalline all affect thermal resistance and capacitance. A noticeable example of this phenomenon lies in silicon’s diamond crystal lattice which has varying thermal resistances in different directions due to differences in packing densities. So from these relations, it can be seen that strained semiconductors will have specific thermal properties. In addition, as silicon is strained, thermal conductivity can increase or decrease, potentially worsening aging effects.
The differential equation for heat diffusion in a solid, such as an integrated circuit is,

$$\rho c_p \frac{\partial T(x, y, z, t)}{\partial t} = \nabla [\kappa(x, y, z, t) + g(x, y, z, t)],$$  

(2.13)

which becomes

$$\kappa(x, y, z, T) \frac{\partial T(x, y, z, t)}{\partial n_i} + h_i T(x, y, z, t) = f_i(x, y, z)$$  

(2.14)

when the general boundary equation is applied [13]. In Equation 2.14, $g$ is power density, $\kappa$ is thermal conductivity, $\rho$ represents the density of the material, $c_p$ represents specific heat, and $h_i$ is the heat transfer coefficient at the boundary surface. The boundary conditions assume the body is free on five sides, while at the same times remains grounded to a heatsink.

### 2.4 Interconnect Destruction by Electron Migration

The electrical resistance of an interconnect is given by

$$R = \frac{\rho l}{A},$$  

(2.15)

where $A$ is the area of the interconnect, $l$ is the length, and $\rho$ is the resistivity of the material. As shown in Section 2.3.1 resistance is thermally dependent, so in therm of temperature, the above equation becomes

$$R = \frac{\rho l}{A} [1 + \beta (t - t_0)].$$  

(2.16)

Modeling work has been done by [14] where complex equations concerning the thermally simulated interconnect are presented. These equations were developed in order to aide in simulating of non-uniform substrate temperature effects on clock signal integrity. A distributed $RC$ delay model was employed in [14], which mainly focused on interconnects.

Destruction of interconnects and chip failure occurs in part by electromigration. Over time,
or with sharp transients of high intensity, material is physically moved from the fabricated constructs. Atoms are tightly bound within crystal lattices, but not at grain boundaries or surfaces between materials. For this reason, in these locations, momentum can be transferred easily intensifying vibration and therefore increasing material diffusion. Defects in the materials composing interconnects cause vibration, when current flows through them, causing scattering, the source of electrical resistance. The heat of the electron kinetic energy in metals scatter and their energy is stored in the lattice as phonons. If there is intense current density, the momentum transferred to the defect by the electrons can cause displacement. These are the modes of electromigration in ICs. In the past, due to the resilience of interconnects and short product cycle of consumer goods, electromigration in solidstate devices rarely emerges, however in newer, smaller technology nodes with smaller features, or products intended to have higher tolerances, such as critical safety components, or military specification devices, electromigration becomes a problem. Black’s equation is a means to calculate the lifetime of an interconnect under stress [15], where mean time to failure is found by,

\[
\text{MTTF} = A J^{-n} e^{Q/(kT)},
\]

(2.17)

where \(A\) and \(n\) are constants, \(J\) is current density, \(Q\) is the activation energy in eV, and \(T\) is the operating temperature. Damage from electromigration is visible under an electron microscope, leaving evidence unique from other kinds of destruction, such as electrostatic discharge. Copper has replaced aluminum in more recent processing technologies [16] because of its greater conductivity, and therefore is less at threat from electromigration. Design rules are built into layout CAD and EDA tools to enable testing prior to manufacture, the the accuracy is questionable. Some interconnects are either scaled back too much, decreasing performance, or not enough, decreasing lifetime. More accurate design rules for interconnects could increase circuit performance. Temperature budgets are built into processing steps of VLSI fabrication so that dopants do not diffuse beyond their designed depths, and interconnects remain intact.
The heating which occurs in operation can continue to contribute to, and exceed this thermal budget.

2.5 Long Tail Effects and Frequency Dependence

Material heating over time does not have a purely exponential response. The thermal response of a material such as a silicon substrate exhibits what is known as the long tail effect. Long tail memory is inherent in electrothermal systems and manifests itself as a deviation from exponential behavior [17]. This memory is extremely important to circuit performance as temperature can be significantly different from what conventional compact models would suggest transiently. Current through transistors will be altered accordingly, significantly altering transient switching performance of the transistors in the circuit. This cycle will adversely affect the quality of design simulation, leading to inconsistency and even failure in design. Electrothermal modeling in fREEDA accounts for these inconsistencies in conventional compact models, leading to more consistency between design and the finished die. Due to thermal long tail memory, the transient rise in temperatures will be different from the exponential case used in conventional thinking. The effect of the thermal memory can range from altered switching transients to device failure due to thermal run-away, as the actual circuit may be far from design specification. Compact electrothermal models use a resistor and capacitor internally to model heat capacity and conductivity to compute temperature, the result of which is an RC circuit possessing an exponential characteristic equation. In the fREEDA electrothermal simulation, the long tail effect is captured in both the initial transient difference and then continues to increase beyond the exponential. The higher temperature that results from long tail memory alters transistor speeds through modification of the electrical conductivity, which alters the available current, in turn.

In dynamic cases, the long tail memory will affect several cycles of data throughput. The switching speed of the transistor becomes a dynamic function of the input signals, which in practice will likely be irregular. The regions on a die where thermal hot spots exist will also
slow transistor switching speeds down in a gradient over the surface depending on proximity.

2.6 Circuit-Based Simulation

In fREEDA simulation `.model` statements reduce overhead and add convenience. `.model` can be used for defining the detail level of a transistor model or other element. Another use which enables netlists to be future proof is to use `.model` to easily indicate different process technologies or material properties such as silicon, SiGe or GaN. This enables quick re-simulation of different options without the pain of material parameter look ups or netlist re-writing.

Accumulated error is a possibility while simulating with any method which can cause corrupted output or a simulator crash due to non convergence [18]. This becomes more likely to occur during long simulations with many time steps. This is something that must be dealt with by the user by making adjustments to the transient run commands in the netlist. The most basic analysis, the Forward Euler algorithm is represented by

\[ x_1 = x_0 + hx'_0. \]  

(2.18)

The Backward Euler method is given as

\[ x_1 = x_0 + hx'_1. \]  

(2.19)

A better combination of the later two techniques is the Trapezoidal iteration. The derivative of the transient is calculated at \( x_0 \) as well as \( x_1 \) using the Backward Euler technique. Thus the step becomes,

\[ x_1 = x_0 + \frac{h}{2}hx'_0 + h'_1. \]  

(2.20)
2.7 Prior Work in Thermal Modeling

Heat sources and their environments can be modeled with the same methods as electrical circuits, because they are both means to represent systems of differential equations. The following table summarizes these relations.

Table 2.1: Relation between electrical and thermal quantities and symbols.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Electrical Units</th>
<th>Thermal Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Capacitance (F)</td>
<td>Capacitance J/°C</td>
</tr>
<tr>
<td>R</td>
<td>Resistance (Ω)</td>
<td>Resistance m·°C/W</td>
</tr>
<tr>
<td>V</td>
<td>Voltage (v)</td>
<td>Temperature °C</td>
</tr>
<tr>
<td>I</td>
<td>Current (i)</td>
<td>Heat Flux W or Q</td>
</tr>
</tbody>
</table>

A voltage supply can therefore be used to keep a temperature constant, such as the boundaries of a body in a universe that is room temperature (300K), and a current source can be used to inject heat into a thermal RC network. AC or DC analysis has been used with SPICE like simulators to calculate thermal profiles [19][20].

2.7.1 Finite Element and Difference Methods

The two main methods of determination of a thermal profile of an IC have been the finite element or finite difference methods (FEM and FDM), and the boundary element method (BEM)[21], which applies a Green’s function. The FEM/FDM methods discretize volumes, and time, if it is a transient analysis, and create a system of partial differential equations. FDM approximates the differential equations [22, 23, 24], while FEM approximates the solution [25, 26]. Both methods are problematic because of the large size of the networks created by the meshing of their volumes. FEM can be applied to differing meshes, rather than square grids. Typically triangular shapes are used, which can change in size and shape to cover important areas, where gradients are
higher triangles are more densely packed. The accuracy of both methods is dependent on the
gridding or meshing size, because if the gradient is changing more abruptly than the distance
of the span between calculated points, the fluctuation will be missed by the solver. Error can
potentially accumulate over a large area of an improperly meshed area to yield more inaccurate
results. While increasing mesh resolution increases accuracy of results, computation time also
increases, therefore the mesher is an important component of the simulation, designed to allow
for maximum efficiency.

Commercial thermal simulators such as Comsol™ and Mechanica™ use the FEM method
to solve for temperature and heat flow and most are not capable of transient thermal modeling.
Most FEM analysis are static only, neglecting a time evolution perspective. All advanced
FEM solvers have a fundamental limit of one million to twenty million unknowns (or degrees
of freedom) set by numerical conditioning and memory limits. The higher limit is usually
for very regular structures solved using computers with 64-bit architectures. This problem-size
limitation prevents the analysis of ICs if submicron feature sizes are to be captured in structures
that are a millimeter or more across.

3-D Thermal ADI is reported in [27] and the Arnoldi model order reduction used in [28]
are further examples of applications of the FEM/FDM method. The Arnoldi iteration is a
sparse matrix algorithm which makes matrix map vectors and makes conclusions from their
images [29]. A method of performing rapid thermal analysis using a compact resistive network
was proposed by Cong et al. [30]. This is an interesting approach but cannot incorporate all
thermal issues nor can it be used to determine thermal transients. The tile approach presented
by Wilkerson et al. [31] uses a resistive grid and has been used as part of the floor-planning
step. Detailed FEM simulations are required for each tile to determine the value of that tile’s
thermal resistance in the final resistive grid.

The use of Green’s function formulation, which is two-dimensional algorithm, has been ap-
piled to resolve thermal problems by [32, 33, 34, 35], which are fast, but less accurate than
their FEM counterparts. Any of the FEM, FDM, BEM, or Green’s function solvers can be
applied to either a cosimulation between an electrical solver and thermal simulation, or independent electrical and thermal simulators, which iterate, known as the relaxation method [22, 26, 36, 37, 38]. With relaxation method projects, existing solvers can be used together to provide results. The relaxation method can be relatively simple to apply if preexisting tools are used, but fast changes cannot be captured, and convergence cannot be attained in highly coupled problems.

ILLIADS-T [22] is a steady state chip temperature solver which enjoys widespread use. ILLIADS-T generates a thermal profile and identifies problematic areas, reliability, and performance from the input of the layout, packaging information, and periodic input signal pattern. Power and temperature calculations are decoupled to increase efficiency when the 3D heat diffusion equation for the chip substrate is solved, while a one-dimensional effective heat transfer macromodel is applied to the package and heatsink boundaries, again for computational efficiency. This approach utilizes a similar approach to that described by Equation 2.10. The relaxation method of [38] relies on using HSPICE as a circuit simulator, then the power information is feed into a thermal simulator to arrive at the thermal profile, which then updates the thermal terminals back in the electrical circuit. This is a similar method to that which has been implemented in fREEDA. fREEDA uses local reference terminals, which is an advantage due to the fact that in other direct approaches, the thermal circuit must in effect be a part of the electrical circuit.

2.7.2 Direct Method

Simultaneous electrical and thermal solutions are solved in the direct method [25]. Direct method simulators consider a thermal network which has been modeled by electrical components that has common thermal nodes with the electrical network [39, 40, 41]. This has been the multiphysics approach used by fREEDA, where terminals can be electrical, thermal, optical, electromagnetic, mechanical, etc [3]. This is an advantageous approach due to the ability to capture fast changes and their effects, though it is a more complex implementation.
Direct method implementations require an electrical-type model to capture the thermal behavior of the circuit and materials composing the body. Thermal nodes have been used to connect thermally equivalent circuits and their electrical counterparts [40]. At the thermal nodes, voltages represent temperatures and currents heat flux. A large network of the combined thermal and electrical networks is then solved using any of the above methods. In this method, device models must be electrothermal, i.e. thermally aware with a separate thermal node. Element equations not only output temperature, they are also thermally dependent upon the environment. Temperature is both input and output by thermal terminals. The time required for the thermal power-on transient to reach steady state is many orders of magnitude higher than electrical events, milliseconds and up to seconds compared to a picosecond or less clock cycle, some moving window power averages have been considered [5].

2.7.3 Batty Model

Quasi-analytic approaches to compact thermal modeling were introduced by Hefner and Blackburn [42, 43], and by Batty et al. [44, 45]. In the Hefner and Blackburn approach, symmetry is used to reduce the 3D heat diffusion equation to a 1-D form. Ad-hoc modifications are required to accommodate both lateral heat spreading and 3D effects at edges and corners. The resultant model can be evaluated rapidly. The method of Batty et al. also results in a partially pre-computed quasi-analytic model and does not have ad-hoc features and is an exact 3D thermal solution but only for limited geometries. It correctly handles thermal nonlinearities and evaluation is sufficiently fast that it can be used in transient electrothermal simulation. The drawbacks of these models are the extensive development time required to derive the models for different types of problems.

Luniya applied the Batty Model to estimate the power dissipated in an X-band MMIC [46]. The Batty model takes Laplace transforms of the heat equation in each spatial direction in order to calculate diffusion.

\[
\nabla[k(T)\nabla T] + g = \frac{\rho C \partial T}{\partial t}
\]  

(2.21)
As described in the above section on thermal conductivity, \( k \) is dependent upon temperature itself by the following,

\[
k = k_s \left( \frac{T}{T_s} \right)^{-b} \tag{2.22}
\]

\[
\theta = T_s + \frac{1}{k_s} \int_{T_s}^{T} k(T) \, dT \tag{2.23}
\]

\[
\nabla^2 \theta - \frac{1}{k(\theta)} \frac{\partial \theta}{\partial t} = -\frac{g}{k_s} \tag{2.24}
\]

The Batty model is efficient given its level of accuracy and fidelity due to its use of Fourier transforms, however large scale problems become intractable, therefore its use is limited. A parochial application of the Batty model to small scale interconnects is appropriate and useful, but RC models are suited to be more generally encompassing.

### 2.8 3DICs

As Moore’s law progresses, operating frequencies increase, transistor dimensions decrease, and circuit densities increase. An effect of decreasing dimensions has been to shrink interconnects drastically in all dimensions. Shortening interconnects is beneficial with respect to electrical gate delay, however at the same time thinning and narrowing interconnects is deleterious. The aspect ratio of interconnect geometry does not necessarily remain constant as dimensions are reduced. Electrical gate delay decreases as lithography dimensions are reduced with diminishing returns, at which point interconnect delay dominates the curve and results in increasing the overall gate delay.

The performance decline due to the \( RC \) time constants of the interconnects occurs around 130 nm, but low-K dielectrics have pushed this minimum feature size down between 90 to 32 nm. There are significant economical burdens to pushing beyond 65nm currently [47] such
as developing ultralow-k dielectrics. Beyond this point 3DICs become a favorable alternative to reducing feature size for some time [48]. 3DICs can greatly reduce the length of interconnects, and in turn reduce delay.

At a 90 nm process, half of the power consumed is due to charging and discharging of parasitic interconnect capacitances. RC delay has become the factor limiting the absolute speed of circuits, due to the fact that propagation delay dictates the minimum clock period [49]. Not only will capacitance increase, but the ITRS includes increasing resistivities in future technology nodes [16]. Due to these factors, wire delay can be expressed by,

\[ t_d = 0.35 r c l^2. \]  

(2.25)

While the decrease of feature size worsens the aspect ratio of interconnects, 3DICs are able to reduce the length, \( l \). The average length of an interconnect in a 3DIC is the square root of the number of layers [50], so according to the above equation, a two-layer 3DIC will half \( t_d \), doubling the performance. This scales favorably by increasing the number of layers of 3DIC by further increasing the performance gained.

### 2.8.1 Chip-Level Stacking

There are three classifications of 3DICs, chip stacking, transistor stacking, and die-on-wafer, also known as wafer-level stacking. Chip-level stacking is the level of integration most commonly used as of publication. Typically wire bonding, along the edges, or flip chip mountings, utilizing the faces, are used with as many as eight chips [51]. Different substrates can easily be stacked with the chip stacking method due to the simplicity of mounting. This is desirable for System on a Chip (SoC) or mixed analogue, digital, or power applications. Placement is as accurate as 10 \( \mu \)m, which limits interconnect density [52]. The main benefit here is the reduction of physical space needed on a PCB. These are typical on cell phones, mobile computing devices, or other applications requiring lightweight computation, such as military appliances. With stacked chips, regular input and output buffers must be used for intercommunication between
the chips, so delay does not benefit.

2.8.2 Transistor-Level Stacking

Transistor stacking is the ultimate goal in 3DICs at some point in the future, where more layers of active semiconductor are continually fabricated during processing. This is extraordinarily difficult for a number of reasons, the relevant one being that the thermal budgets discussed in Section 2.4 are easily exceeded. Normal fabrication methods would cause destruction to copper or aluminum layers in finished layers. Dopant depth and density in active semiconductor regions would be impossible to control due to migration. Some success has been found through laser annealing of smaller areas and nickel nucleation [53, 54].

2.8.3 Wafer-Level Stacking

Wafer-level 3DICs are fabricated, at the start, using fairly standard VLSI processing techniques. Finished wafers are then stacked, facing one another, and bonded together. The remaining silicon wafer, known as the handle is removed from one of the wafers by chemical mechanical planerization (CMP) or some other means, such as etching. At this point, the resultant configuration is a standard IC, on top of a wafer, then a bond, then another inverted, face down circuit, which was originally part of another wafer. This second layer of circuitry, known as a tier, is of course interconnected with the first tier with vertical interconnects, known as vias. There must be an alignment step during the tier stackup, which is critical to functional chips in the end. Special alignment marks may be present on the back, or etched through the entire wafer to ensure that electrical connections are in fact made. It is a particularly difficult alignment, because it is blind, in the sense that all of the structures and features are face-to-face. Another possibility is the use of IR cameras to see alignment features through the wafer during this step.
2.8.4 Wafer Bonding

The method of wafer bonding is important to thermal considerations and thermal modeling due to the facts that the materials used have different thermal properties, and the bond layer effectively caps and isolate the active regions. The bonding material may either be an epoxy or organic glue, such as that used in the IBM, and RPI processes [55], or the spontaneous growth of a native SiO$_2$ oxide in the Massachusetts Institute of Technology Lincoln Laboratory (MITLL) and ZiptronixTM processes [56]. Metal bonding is also used by Tezzaron [48]. While SiO$_2$ is a bad thermal conductor, epoxy can be worse, and particular thermal parameters of IBM’s process are proprietary, and it does not provide a metal interconnect because a SiO$_2$ bonding layer is used between tiers which must be drilled and filled to allow for interconnects, rather than the metal bonding layers used in Tezzaron’s process. Epoxy is however a good management tactic for contaminants. Oxide bonding allows room temperature bonding, which does not further add to thermal budgets, but again does not provide a metal interconnect layer. Metal bonding does accommodate an interconnect layer with no extra steps, but requires heating of the entire structure to 400°C. Alignment must take place at this temperature, and if different substrates are used, their thermal coefficients of expansion further complicate the process.

The tiers are innervated by vertical vias that can either be created after the wafers are bonded, or before. The via-first method consists of two pre-fabricated via halves on each wafer which are then joined. This is more accessible and the most cost effective [57]. The density of vias is dictated by alignment tolerances, which are $< 1 \mu$m for Tezzaron [48], better than a tenth of chip to chip alignment.

3DICs can probably benefit the most for some years to come with only two layers before adding more, on average. Indeed, much research has been in perfecting processes with two tiers before moving on [58, 59]. The 3DIC used in this work used the MITLL process, covalent oxide bonding, and three tiers [60].
Figure 2.1: MITLL 3DIC process stackup. After [60].
2.9 Summary

There are a number of problems derived from thermal effects, such as interconnect destruction due to electro migration, dopant diffusion, and increase in electrical resistivity due to temperature. All of these problems are presently bad, and will worsen in future technology generations due to increasing interconnect impedance.

There are a multitude of simulator approach classifications for the electrothermal problem. Each can be appropriate for specific scales, effects, and circuit types. The decision of which type to employ, each with their own sets of detractors and strengths rests with the engineer. 3DIC’s alleviate many of the technological boundaries, but require thermal study by simulation in order to achieve payoff.

Having reviewed simulator methodologies, 3DIC technology, and past research, the remainder of this dissertation is devoted to work the author contributed to in collaboration with others toward forwarding the current abilities in electrothermal simulation.
Chapter 3

Experimental Setup for 3DIC and Test Board

3.1 Introduction

Experimental validation is the crux of any simulation model, however accurate thermal measurements to the scale needed for this body of research are difficult. A 3DIC was chosen for measurement that could also be fairly easy to implement in simulation. A token generating 3DIC designed by the Asynchronous VLSI and Architecture group at Cornell University was chosen, because it generated a significant amount of heat, was limited in the number of circuit cell types, and possessed a repetitive architecture on each level. The die contains three working tiers, each with nine separate, independently working tiles. Each tile is composed of a token generator, followed by a chain of frequency multipliers that become increasingly fast, then a chain of frequency dividers, and finally an output. This chip can be electrically measured to verify output and thermally imaged to verify simulation. The 3DIC is also a good candidate because it contains heat producing elements at different levels, which is an excellent proof of concept for the problems attempting to be solved.

Section 3.3 describes the procedure for operating the circuit board, as well as its assembly.
Due to the method of thermal measurement, a requirement was that the die must be held at a constant temperature well above room temperature. This was problematic because the package was mounted on a printed circuit board (PBC), and also the top had to be unobstructed for imaging. Section 3.4 describes the design of the heatsink. Finally, precise technical drawings of the board, package, and heatsink had to be made in addition to collection of all thermal properties for simulation.

3.2 Design and Functionality of the 3DIC

The physical dimensions of the die have been important to consider because heating is particularly sensitive to the mass of material that power is dissipated into. The size of the die is $3.76 \text{ mm} \times 3.76 \text{ mm}$, with the individual $1 \text{ mm}^2$ tiles repeated nine times on each of the three tiers. The total thickness can be computed by summing the layers described in the MIT Design Guide [60], for a thickness of 22,450 nm, the 675 $\mu$m handle, and the 1 $\mu$m over glass passivation layer. The total die thickness is found to be 698 $\mu$m. The pad size is 70 $\mu$m and pad pitch is 100 $\mu$m.

3.2.1 Block Diagram of Floorplan and Layout

The automated design described in Section A.1 provides the framework for a chain of operations to be performed on an initial generated signal. A block diagram describing the function of every cell onto the tokens as they pass though the circuit is presented in Figure 3.3. Tokens are generated at the controller circuit in the top left of the diagram, and follow the daisy chain of frequency multipliers and dividers to the final bucket circuit stage. The highest frequency of switching is between the last frequency multiplier and the first frequency divider at 1 GHz. The fastest switching junction is also the highest contributor of heat in the circuit and has been focused on in simulation. The hotspot is localize within the bottom center of the block diagram, which corresponds to the circuit layout.

The 3DIC layout is based on a custom cell design. The layout of the four fastest switching...
cells comprising the hotspot focused on in this research is depicted in Figure 3.2. Yellow rulers delineate the borders of the cells. A representation of the three-dimensional floor plan is depicted in Figure 3.1 by [61]. On the right side of the die there is an increased number of vertical metal thermal vias; these could be used to explore the impact of internal cooling structures on the die.

![Figure 3.1: The floorplan [61] of the three tiers of the 3DIC (a) with thermal vias and (b) the functionality of the cells within each tile.](image)

**3.3 Design of the Test Board**

The four-layer test board was designed by the Asynchronous VLSI and Architecture group at Cornell University [61]. The layout of the test board is depicted in Figure 3.4. The threedimensional model depicted in Figure 3.5 was useful along side the layout to determine which areas of PCB could safely be removed for heatsink attachment.
Figure 3.2: The four hotspot cells, delineated by yellow rulers in the layout.
Figure 3.3: The block diagram of the connectivity of each IC tile including the frequency multiplier and divider chains.
Figure 3.4: The four layers of the PCB test board with IC socket, output region, and reset circuitry positions. Designed by [61]
Figure 3.5: The 3D model of the test board generated by PCB123 was useful in determining areas safe to modify for heatsink attachment. Designed by [61]
The pinout diagram shown in Figure 3.6 was used in addition to the bonding diagram depicted in Figure 3.7 to interface the IC with the package. The Cadence layout in visible in the bonding diagram to represent the die. The output buffers are attached to the pads on the inside flange of the IC package in accordance to the diagram. Gold wires were sonically welded into place. Wire bonds are potentially an important route of heat flux escape from the IC to the package pins. They are long and thin, but good thermal conductors. This conduction is undesirable if it can be avoided by redirection though the heatsink because of potential thermal distortion of high frequency I/O signals. The signals are then internally routed within the black alumina package by gold traces to the output pins. The package used was a black alumina SSM P/N CPG12031 by Spectrum Semiconductor Materials, Inc™.

A 325 mil PGA121M square cavity pin grid array socket was used to attach the package to the PCB. The socket was mounted on the back of the PCB, keeping the pin connections consistent. Every large component was mounted to the back of the PCB except for small surface mount passive components and the NAND gate IC. As many components as possible including the output header pins, voltage jacks, and operation switches, so that there was a low clearance from the PCB to the heatsink. It was beneficial to measurements to maintain the 3DIC die as close as possible to the heatsink.

3.4 Through-PCB Heatsink Implementation

The ability to thermally image integrated circuits without thermal ink diminishing the quality of the accuracy and resolution is accompanied by the need to securely attach to a heated thermal chuck. The method used to map the thermal emission of the surface of the 3DIC required the 3DIC to be held at temperatures well above ambient. There were a few different possibilities, all of which were complemented by the PCB. In order to bring the core of the IC to as close as possible to the temperature of the heated chuck, a heatsink was designed which possessed a
Figure 3.6: The pinout diagram of the package with signal names mates to the IC socket on the PCB.

Figure 3.7: The bonding diagram of the IC die with labeled pin outs as it mates to the package.
Figure 3.8: The microscopic image shows the gold wire bonds connecting the 3DIC die to the package pads. The gold layer with traces and black alumina are visible. The three independent circuit tiles are visible on the silicon die [Image collaboration with S. Melamed and S. Lipa].
post that mated with the IC package through the PCB.

In order to determine if it was possible to cut a hole in the PCB in the area inside the package socket, the PCB layout had to be examined. A four-layer board was used with ground and $V_{DD}$ planes, in addition to a thermal $TV_{DD}$ plane, which provides power to heating transistors in the chip. The $TV_{DD}$ plane is a separate power plane used to deliver power to heating transistors within the 3DIC, which were not used in this work. There is a wire trace as well as a thin section of ground plane which transects the area inside the socket, visible in Figure 3.4. The size of the copper post was dictated by the largest size possible to remove from this area of the PCB. A large copper block of copper with a post projecting from the center was machined to fit the requirements of the test board. Screw holes were threaded at each corner allowing the test board to be tightly fastened down to the post. Flexibility of the test board allows for a high amount of pressure to be applied between the IC package and copper post, eliminating air gaps. A high quality silver thermal paste was also used to assist in thermal consistency across the junction. In measurement, electrical shorting of the internal PCB layers by the copper heatsink post prevented functionality. To alleviate the electrical shorting, the post was insulated.
Figure 3.9: The top view mechanical drawing used for board-level thermal simulation.
Figure 3.10: The front view mechanical drawing used for board-level thermal simulation.
3.5 Mechanica Verification

PTC®’s Creo Elements/Pro Mechanica™, formerly known as Pro/ENGINEER Mechanica™, is a mechanical design simulation CAD environment which incorporates the ability to apply thermal loads, prescribed temperatures, and convection coefficients to an FED/TEM analysis of the body.

A concern was that the heatsink would be ineffective as the post was so long. A good thermal ground would not be provided and then calibration for thermal imaging would be incorrect. PCB level simulations demonstrated that the heatsink accomplished its designed function and relatively little heat flux entered the PCB via the package pins and wirebonds. Figure 3.11 presents the Mechanica simulation of the heatsink post. Vector arrows indicate the direction and intensity of the heat flux. Large white arrows in the heatsink post point down away from the die. Smaller blue arrows are present in the package pins as some heat is conducted to the PCB. All of the thermal properties of all materials used in the IC package and PCB were considered. Figure 3.12 depicts the entire board level thermal simulation. Some minimal heating of the PCB occurs.
Figure 3.11: Mechanica simulation of 3DIC die showing that the majority of heat flux goes through the Heatsink Post.

Figure 3.12: Mechanica simulation of 3DIC package and heating of the PCB.
3.6 Conclusions

Establishment of novel modeling techniques necessitates result verification. Complicated modeling, which in particular examines two or more related phenomena, e.g., heat and voltage, require experimental setup. This chapter has recounted how the physical experiments may be repeated. All of these steps have been important in demonstrating the application of the research to a real world problem.

The bottom up design of the 3DIC has been described, including the asynchronous language implemented by [61]. The functionality of the series of frequency multipliers and dividers, and the inclusion of extra vertical metal structures to act as thermal vias has been presented. The design of the test board PCB, and design of the through-PCB heatsink suitable for thermal imaging, have been described. All of these innovations have been necessary for verification of simulation.
Chapter 4

Electrothermal Simulation of a 3DIC

4.1 Introduction

Time evolving transient simulation techniques may be employed in simulation to preemptively determine problematic design regions. In previous technology generations, design engineer experience was relied upon for dependable functionality of design. In current and future generations, especially those employing 3DIC technology, thermal issues can no longer be predicted by intuition, e.g. not placing heat sensitive memory architecture close to heat intensive clock trees. The methods described in this chapter are the first steps to discovering the severity of interconnect destruction during operation. The extremes of operation are only realizable through transient simulation because the intensities of maximum heating spikes are responsible for electromigration, rather than time-averaged steady state temperatures.

4.2 Simulation Environment

Large-scale computations of physical systems involve the simultaneous solution of a number of differential or partial differential equations that approximate the operation of the components of the physical system. The numerical algorithms that generate solutions to these systems of equations involve the evaluation of a Jacobian matrix: a matrix consisting of partial derivative
entries of the outputs of the system with respect to the inputs. A typical approach for finding the values of these derivatives has been to use finite differences, but they are known to have several drawbacks. In particular, choosing a step size that will provide minimal truncation error and still allow for a fast and stable simulation is far from being trivial, especially when the problem under consideration is strongly nonlinear. Having strong nonlinearities in a system is often the case when modeling real physical systems. Another approach is to compute these derivatives exactly using the technique of Automatic Differentiation (AD) [62]. With this technique it is possible to compute first and higher-order derivatives with arbitrary accuracy, thus eliminating a process in simulation which is known to be time-consuming and error-prone. AD has been used in circuit simulators in the past [63] and is currently used extensively in the fREEDA simulator, the implementation details of which are described in [3]. AD can be used in two modes: a so-called Forward Mode and Reverse Mode. In the Forward Mode, the partials are evaluated while carrying out each operation on the expression of interest, while in the Backward Mode, all the expressions are evaluated first and re-visited in reverse order to compute the partials. Details of these modes can be found in [62].

As described in [3], prior versions of fREEDA used a C++ package ADOL-C [64] for obtaining derivatives automatically. ADOL-C uses the C++ feature of operator-overloading which requires minimal modifications to existing source code. It also uses a temporary buffer known as a tape to record the arithmetic operations while evaluating an expression in the model code. Subsequently, when derivatives are required, they are “played back” from this tape rather than revisiting the relevant source code. The advantage of this is that reading off the tape can be made very efficient and can reduce some of the overhead associated with AD. The disadvantage of this approach is that the length of this tape must be set prior to simulation. If one wants to run a simulation involving a large number of elements, then one has to know beforehand the length of the tape required. This involves recompiling ADOL-C source code. Furthermore, as the length of this tape increases, the simulation can slow down considerably. Also, ADOL-C

\footnote{The implementation of the Trilinos package and development of work described in this section was by Nikhil Kriplani at North Carolina State University.}
requires conditional statements in the source code to be modified to a format that makes it suitable to place on the tape. However, in many cases this is very hard to do, if not impossible, and whenever a non-conforming conditional statement switches to an alternate branch of execution, the entire tape must be recreated which further slows down the simulation. Obtaining automatic derivatives without having to consider tape lengths, source recompilation, and conditional statements, and still be able to perform large-scale simulations in a reasonable amount of time is an ideal scenario.

An alternative is to use an AD package called Sacado\textsuperscript{TM} [65] which was developed at Sandia National Labs as part of the Trilinos package. Sacado still creates a temporary memory buffer but this is executed iteratively on the fly, as much is necessary without the need of converting conditional statements in code to a tape-compatible version. The management of the size of this temporary buffer is automatic and does not require any intervention from the model developer. The most recent version of fREEDA uses Sacado for all its AD purposes, and this has enabled performing large simulations both in terms of number of model elements and simulation length. The electrothermal simulations in this work were not possible otherwise with the large number of thermal resistive elements present in the layers of the 3DIC. When compared with the simulation time incurred using ADOL-C (within the limitations of ADOL-C), Sacado’s performance was found to be about 1.4 times slower.

### 4.3 The RC cell model and expandability

Two thermal bulk models created for fREEDA are the ThermalBlockRC and ThermalHeatsink elements. The ThermalHeatsink element is based on the Batty thermal model, while ThermalBlockRC is a two-dimensional multiple resistor and capacitor element. The Batty model performs a Fourier transform in three dimensions so that elaborate heat models are not needed. The RC method reduces a high order differential heat equation to a lower order, where a fractional model can be used. Long tail effects can be realized through this solution. Thermal-BlockRC3D has also been created as a three-dimensional improvement on the latter with a 6
resistor structure. Under some cases, these models are equivalent in results. Since Thermal-
Heatsink has a higher CPU time cost, ThermalBlockRC3D is preferred in these cases.

The $RC$ approach in thermal analysis works off the analogy to electrical circuit elements as
representations of, in this case, first-order differential equations. The current into or out of a
capacitor is,

$$i = \frac{dq}{dt} = C \frac{dv}{dt}. \quad (4.1)$$

So the voltage across a capacitor is,

$$v(t) = v(t_0) + \frac{1}{C} \int_{t_0}^{t} idt. \quad (4.2)$$

Working from the above capacitor equation and rearranged so that a capacitor discharges now
through a resistor,

$$-\left( C \frac{dv}{dt} \right) = \frac{v}{R}, \quad (4.3)$$

is rearranged to,

$$\frac{dv}{dt} + \frac{v}{RC} = 0, \quad (4.4)$$

so that it is a first order ordinary differential equation, the solution of which is,

$$v(t) = v_0 e^{-\frac{t}{RC}}, \quad (4.5)$$

for a discharging capacitor, where $v_0$ was the original voltage. For a charging capacitor, for
example looking at the heat rise when a circuit is energized, the equation becomes,

$$v(t) = v_{\text{final}} \left( 1 - e^{-\frac{t}{RC}} \right). \quad (4.6)$$

Due to the fact that the function depends on $t/\tau$, if $\tau$ or $t$ are scaled by the same ratio,
the resulting functions will be equivalent. Figure 4.1 demonstrates this property of time scal-
ing. Curve (a) is Equation 4.6 with the physical quantities of the capacitance included. $R$ is
equivalent to $1 \times 10^6 \, \Omega$ and $C$ is equivalent to $2.5 \times 10^{-7} \, \text{J/K}$. Curve (b) is the same equation, however its time axis has been expanded by multiplying by 10. Curve (c) does not scale the time axis, but does scale the capacitance value while holding $R$ constant. The new $C2$ is set to $2.5 \times 10^{-6} \, \text{J/K}$, which is again scaled by 10. The $RC$ time constant in the first equation is .255 and the $RC$ time constant in the second equation in which $C$ has been scaled is 2.5, so the scaling factor of $RC$ is 10. Figure 4.1 shows that scaling either the $RC$ time constant or the time axis by the same ratio will yield equivalent functions. In Section 4.3.2, it will be shown that this can be applied to simulations in order to save significant amounts of CPU time.

Figure 4.1: The transient rise of a point on a hotspot of the overglass has been scaled by time to match capacitive values.
4.3.1 Thermal Capacitance Calculation

If a cubic volume of silicon is considered with edges equivalent to 22.45 µm, then the thickness of the 3DIC tested, the mass of the cube is

\[ m = \frac{2.328 \text{ g}}{\text{cm}^3} \cdot 22.45 \mu\text{m}^3 = 5.228 \times 10^{-14} \text{ kg}, \tag{4.7} \]

where 2.328 g·cm\(^{-3}\) is the density of silicon. The thermal capacity of that mass is computed using the specific thermal capacity of silicon at ambient temperature 298.15 K and pressure 101.3 kPa, which is 705 J/(kg·K), by

\[ C = 705 \frac{J}{\text{kg} \cdot \text{K}} \cdot 5.228 \times 10^{-14} \text{ kg} = 36.862 \times 10^{-12} \frac{J}{\text{K}}. \tag{4.8} \]

However the density of SiO\(_2\) is 2.648 g·cm\(^{-3}\) and the specific heat is 1 J(g·K\(^{-1}\)) \cite{66}, so the thermal capacitance of the same volume is,

\[ m = \frac{2.648 \text{ g}}{\text{cm}^3} \cdot 22.45 \mu\text{m}^3 = 5.945 \times 10^{-14} \text{ kg}, \tag{4.9} \]

\[ C = 1 \frac{J}{\text{g} \cdot \text{K}} \cdot 59.448 \times 10^{-15} \text{ kg} = 59.448 \times 10^{-12} \frac{J}{\text{K}}. \tag{4.10} \]

If the handle is 675 µm and the total thickness of the die including the active 3DIC layers plus the SiO\(_2\) overglass is 698 µm, the assumption can be made that the thermal capacitance of this cube can be considered as 96.7% silicon, with the balance composed of SiO\(_2\). A weighted average was used to calculate the thermal capacitance value accordingly. By comparison, the extracted physical thermal netlist for the simulated 3DIC hotspot is 101 µm ×54 µm. Four thermal capacitances were used per layer for this area; one per each cell. The value of capacitances used were varied between an octive of 10×10\(^{-14}\) and 20×10\(^{-14}\) J/K in order to make use of time compaction. The variance of these values was used to prove feasibility of time compaction for more rapid simulation.
4.3.2 Capacitive Scaling

Transient electrothermal simulations are particularly CPU-time intensive as electrothermal transients occur at the picosecond scale while the thermal transients can be seconds. In the Cornell 3DIC, there is a large array of different frequencies as can be seen in Table 4.1. This is due to the nature of the circuit containing many repeating frequency multipliers and dividers, each operating on the output of another. So, to simulate only one complete cycle on the slowest operating cells of 61 kHz, approximately 16,393 cycles have to be simulated on the fastest switching cell at 1 GHz.

![Image](image.png)

Figure 4.2: Transient electrothermal simulation that takes a long time to rise (a), and thus a long time to compute is time-scaled (b) matching another simulation (c) wherein the RC time constant has been scaled by the same ratio.

Figure 4.1 showed that exponential equations can be time-scaled by reducing the time constant. Using the physical heat capacity of the IC materials resulted in Curve (a) in Fig. 4.2. Both curves in the simulation shown required 8.24 days of simulation using a single core of a 64-bit Intel Xenon machine clocking at 3 GHz. Quicker simulations were obtained by using time-
scaling implemented by reducing the thermal capacitors (and thus the thermal time constant of the thermal circuit) by a factor of 10, allowing steady-state conditions to be reached sooner. This resulted in the thermal response shown as Curve (c) in Fig. 4.2. Dividing the simulated time of Curve (a) by a factor of 10 (the same ratio by which $RC$ was reduced) resulted in the scaled curve (b) being virtually coincident with Curve (c). The $RC$ time constant, $\tau$, of the thermal circuit can only be reduced by changing $C$, whereas changing $R$ would have resulted in incorrect thermal values. Thus, a lower $C$, will reduce $\tau$, and artificially compact the time it takes for temperatures to rise to their steady state conditions. Though both curves (b) and (c) start at 300 K, curve (c) rises quickly from the initial transient due to the reduced thermal capacitance. This validation of the time scaling technique is also supported by the long duration thermal results that are presented in the next section.

4.4 Workflow

The electrical and thermal universes are considered as separate entities during netlist generation, which are later joined via fREEDA’s thermal terminals for absolute communication. The two problems must be independently approached from the onset before joining, however. Figure 4.3 describes the course of actions necessary to produce electrothermal simulations matching known measurements. Electrothermal macromodels are designed which are representative of the functionality of the circuit. Alternately, preexisting standard cell macromodels available with the fREEDA package may be used in the circuit netlists. Parameter information about the process technology node must be included. This information may require additional model writing, however modular design allows for simplistic inclusion.

The physical geometry of the fabricated device must be considered for the physically-aware thermal netlists, which are extracted from the layout by the process described in Section 4.7. In practice, the electrothermal netlist will be tested independently before joining the extracted thermal netlist by use of stand-in resistors connected between thermal terminals and thermal
ground or reference. Once the two commensurable netlists are created, they must be joined via the thermal terminals of the electrothermal fREEDA elements. Matlab helper scripts were created to properly create the long lists of node names for both circuit and subcircuit netlists, which are discussed and listed in Appendix B.

As the final sequence in this workflow, simulations are then run, and in this case, thermal measurements were available for verification of desired results. Calibration of results to account for bad heatsink joints or other irregularities could be made by addition of resistors in parallel with the thermal network.
4.5 Computation of Power Numbers

The power numbers which the electrothermal fREEDA elements were based on were generated independently in HSPICE using the full transistor-level circuit netlists of each cell. The repeater (frequency multiplier) and eater (frequency divider) for each stage in the chain was simulated, so that there was a corresponding power number for each frequency. Before power numbers could be generated, the internal frequencies of each cell had to be determined. The designed output is somewhat different from the measured output as will be described in Chapter 6. Output frequency is the only value which can be directly measured, so with knowledge of this frequency, the rest of the cell operating frequencies were back computed. The backward-computed frequencies used in simulation are listed in Table 4.1. These frequencies are also used in fREEDA electrothermal simulation of the hotspot. The decimal precession of the numbers in Table 4.1 is important because due to the number of times the token frequency is multiplied and divided by two, error will accumulate significantly. Table 4.2\(^2\) lists the results of the HSPICE simulations that were later used to tune \(R_{th}\), a thermal resistance property of each cell in the chain. Additionally there is a controller cell at the initialization of the multiplier chain preceding \texttt{repeater1} that was required to generate tokens and handshake with the first repeater; as well as a bucket cell which performed a similar function with the output of the last eater and provides the necessary handshaking. The bucket and controller cells were neglected from the power number computation because they contribute very little to the total power consumption of the tile.

4.6 Electrothermal Simulation

Electrothermal macromodels were developed based on the functionality of the custom asynchronous cells. The macromodel circuit for the frequency multiplier was based on an XOR

\(^2\)This method was a result of collaboration with Shivam Priyadarshi, and the simulated data was provided by Shivam Priyadarshi
Table 4.1: Standard cell numbers and corresponding simulated frequencies.

<table>
<thead>
<tr>
<th>Cell Name</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeater 1</td>
<td>61.03515625kHz</td>
</tr>
<tr>
<td>Repeater 2</td>
<td>122.0703125kHz</td>
</tr>
<tr>
<td>Repeater 3</td>
<td>244.140625kHz</td>
</tr>
<tr>
<td>Repeater 4</td>
<td>488.28125kHz</td>
</tr>
<tr>
<td>Repeater 5</td>
<td>976.5625kHz</td>
</tr>
<tr>
<td>Repeater 6</td>
<td>1.953125MHz</td>
</tr>
<tr>
<td>Repeater 7</td>
<td>3.90625MHz</td>
</tr>
<tr>
<td>Repeater 8</td>
<td>7.8125MHz</td>
</tr>
<tr>
<td>Repeater 9</td>
<td>15.625MHz</td>
</tr>
<tr>
<td>Repeater 10</td>
<td>31.25MHz</td>
</tr>
<tr>
<td>Repeater 11</td>
<td>62.5MHz</td>
</tr>
<tr>
<td>Repeater 12</td>
<td>125MHz</td>
</tr>
<tr>
<td>Repeater 13</td>
<td>250MHz</td>
</tr>
<tr>
<td>Repeater 14</td>
<td>500MHz</td>
</tr>
<tr>
<td>Repeater 15</td>
<td>1Ghz</td>
</tr>
<tr>
<td>Eater 1</td>
<td>500MHz</td>
</tr>
<tr>
<td>Eater 2</td>
<td>250MHz</td>
</tr>
<tr>
<td>Eater 3</td>
<td>125MHz</td>
</tr>
<tr>
<td>Eater 4</td>
<td>62.5MHz</td>
</tr>
<tr>
<td>Eater 5</td>
<td>31.25MHz</td>
</tr>
<tr>
<td>Eater 6</td>
<td>15.625MHz</td>
</tr>
<tr>
<td>Eater 7</td>
<td>7.8125MHz</td>
</tr>
<tr>
<td>Eater 8</td>
<td>3.90625MHz</td>
</tr>
<tr>
<td>Eater 9</td>
<td>1.953125MHz</td>
</tr>
<tr>
<td>Eater 10</td>
<td>976.5625kHz</td>
</tr>
</tbody>
</table>

with a delay element [67]. As these frequency multipliers are chained together, the delay of each subsequent multiplier must be adjusted in the netlist in order to equal half the output frequency of the previous multiplier in the array. The frequency divider cells are composed of D-flip-flops (DFFs), which in effect function as a 10-bit binary counter. By producing a 1 at the Q output after inputting two 1’s at the clock input, the DFFs will count and divide the input frequency by two.

The individual electrothermal elements making up the macromodels have been optimized by
Table 4.2: Standard cell simulated powers from HSPICE.

<table>
<thead>
<tr>
<th>Cell Name</th>
<th>Static Power (W)</th>
<th>Dynamic Power (W)</th>
<th>Total Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>6.90800E-06</td>
<td>1.06978E-05</td>
<td>1.76058E-05</td>
</tr>
<tr>
<td>Repeater 1</td>
<td>8.96700E-06</td>
<td>1.29210E-05</td>
<td>2.18880E-05</td>
</tr>
<tr>
<td>Repeater 2</td>
<td>9.00300E-06</td>
<td>9.08700E-06</td>
<td>1.80900E-05</td>
</tr>
<tr>
<td>Repeater 3</td>
<td>9.00300E-06</td>
<td>3.16860E-06</td>
<td>1.21716E-05</td>
</tr>
<tr>
<td>Repeater 4</td>
<td>9.00300E-06</td>
<td>5.70300E-06</td>
<td>1.47060E-05</td>
</tr>
<tr>
<td>Repeater 5</td>
<td>9.00300E-06</td>
<td>7.63440E-06</td>
<td>1.66374E-05</td>
</tr>
<tr>
<td>Repeater 6</td>
<td>9.00300E-06</td>
<td>1.05450E-05</td>
<td>1.95480E-05</td>
</tr>
<tr>
<td>Repeater 7</td>
<td>9.00300E-06</td>
<td>1.46310E-05</td>
<td>2.36340E-05</td>
</tr>
<tr>
<td>Repeater 8</td>
<td>9.00300E-06</td>
<td>2.38830E-05</td>
<td>3.28600E-05</td>
</tr>
<tr>
<td>Repeater 9</td>
<td>9.00300E-06</td>
<td>4.08390E-05</td>
<td>4.98200E-05</td>
</tr>
<tr>
<td>Repeater 10</td>
<td>9.00300E-06</td>
<td>7.28970E-05</td>
<td>8.19000E-05</td>
</tr>
<tr>
<td>Repeater 11</td>
<td>9.00300E-06</td>
<td>1.36113E-04</td>
<td>1.45116E-04</td>
</tr>
<tr>
<td>Repeater 12</td>
<td>9.00300E-06</td>
<td>2.60277E-04</td>
<td>2.69280E-04</td>
</tr>
<tr>
<td>Repeater 13</td>
<td>9.00300E-06</td>
<td>4.99857E-04</td>
<td>5.08600E-04</td>
</tr>
<tr>
<td>Repeater 15</td>
<td>9.00300E-06</td>
<td>1.86840E-03</td>
<td>1.87740E-03</td>
</tr>
<tr>
<td>Eater 1</td>
<td>6.04300E-06</td>
<td>1.39472E-03</td>
<td>1.40076E-03</td>
</tr>
<tr>
<td>Eater 3</td>
<td>6.06200E-06</td>
<td>3.50518E-04</td>
<td>3.56580E-04</td>
</tr>
<tr>
<td>Eater 4</td>
<td>6.06200E-06</td>
<td>1.77358E-04</td>
<td>1.83420E-04</td>
</tr>
<tr>
<td>Eater 5</td>
<td>6.06200E-06</td>
<td>9.04360E-05</td>
<td>9.64980E-05</td>
</tr>
<tr>
<td>Eater 6</td>
<td>6.06200E-06</td>
<td>4.65340E-05</td>
<td>5.25960E-05</td>
</tr>
<tr>
<td>Eater 7</td>
<td>6.06200E-06</td>
<td>2.41420E-05</td>
<td>3.02400E-05</td>
</tr>
<tr>
<td>Eater 8</td>
<td>6.06200E-06</td>
<td>1.28560E-05</td>
<td>1.89180E-05</td>
</tr>
<tr>
<td>Eater 9</td>
<td>6.06200E-06</td>
<td>7.80700E-06</td>
<td>1.38690E-05</td>
</tr>
<tr>
<td>Eater 10</td>
<td>6.01300E-06</td>
<td>5.73380E-06</td>
<td>1.17468E-05</td>
</tr>
<tr>
<td>Bucket</td>
<td>6.29700E-07</td>
<td>2.31780E-07</td>
<td>8.61480E-07</td>
</tr>
<tr>
<td>Amplifier</td>
<td>2.03100E-06</td>
<td>3.65700E-06</td>
<td>5.68800E-06</td>
</tr>
</tbody>
</table>
Priyadarshi [68], by reducing the number of unneeded internal state variables. Considering the NAND, the state variables have been reduced to four electrical and one thermal, which totals 20 for the entire frequency multiplier cell. The DFF requires 35 electrical and nine thermal state variables.

![Figure 4.4](image1.png)

Figure 4.4: The electrothermal macromodel for one element in the chain of frequency multipliers utilizes a delay element.

![Figure 4.5](image2.png)

Figure 4.5: The macromodel circuit for the frequency divider chain includes D-flipflops.

Early on a 5 µm resolution simulation was postponed in favor of a 10 µm resolution simulation. Previously, simulations performed with this extractor enabled identification and placement of heat sources from within the layout editor [69]. The layout database was annotated to allow the extractor to identify where to insert the power elements. Thus an easy way to visualize the grid with the correct resistor numbers and node names had not previously been created for such small netlists. Therefore a script was created to draw larger grids for easier connection and 10 µm presented a smaller grid size that was more suitable for debugging because of fewer connections. A number of memory issues were also initially present when simulating so many
elements in the multiple layers of a 5 µm resolution grid. Reducing the grid size, then updating to newer scientific libraries in fREEDA 2.0 eliminated the problem. There is now no limit to the number of elements simulated, other than those imposed by RAM addressable on the system.

4.7 Extracted RC Thermal Network Approximation

Figure 4.6: The thermal resistances are assembled and numbered by the extractor in this order.

The physical extractor, known as WireX\textsuperscript{3}, creates a resistive netlist from an Open Access layout [70]. The values of resistances are based on geometric proximity and material properties. For example, a short metal interconnect will have a low thermal resistivity, and conversely, an expanse of polysilicon will contribute a relatively high resistance to the grid. At present, the thermal capacitances of quantities of material are neglected by the extractor. To cope with this incomplete feature, estimated thermal capacitances were manually added to the final netlists so that preliminary transient simulations are possible. Automatic thermal capacitance extraction can potentially become part of the extraction script and design flow. The extractor is equally compatible with 2DICs and 3DICs, and an example division of a 3DIC layer is shown in Figure 4.7. The grid size of the final netlist is completely user defined. Figure 4.9 depicts the stackup of the 3DIC and how the resistive terminals are numbered for this example. The extractor starts on the bottom layer and works up. The resistor symbols on the right represent the relative thermal resistances of the layers, as based on size and material. The BOX layer

\textsuperscript{3}WireX was developed by Samsom Melamed, and later modified by the author to output results in fREEDA netlist format. The ability to separate the named 3DIC layers was also added.
is a bulk oxide in the SOI process, which has good electrical characteristics insulating and preventing fourth terminal transistor gate latch-up, but has undesirable thermal properties. Silicon dioxide is a poor thermal conductor and in some applications it can be considered to be an open thermal circuit; not conducting at all. Each tier has its own BOX and active layers, each active layer has been considered in this work. Electrical and thermal compact elements exist within ﬂREEDA which can be networked together and simulated. Figure 4.6 illustrates a basic resistive cell from an extracted layout. The node numbering sequence is important when devising a simulation so that correct connection points as well as measurement points can be realized. Six-resistor network meshes have been used with success in ﬂREEDA in previous works [46, 71], however in these earlier works, resistive values were manually assigned based on size and material. In this evolution of the simulator, the process is now automated.

4.7.1 Procedure for Thermal Netlist Extraction

The general method for generating the physical thermal netlist begins with loading the 3D layout in the layout editor, selecting and cropping an area of interest, flattening the layers and tiers, and finally parsing the file using the WireX scripts to yield a resistive netlist. This section will provide detail so that results presented can be reproduced. The core extraction script is thermExtPreProc.il. Some preprocessing steps must first be taken. In Cadence\textsuperscript{TM}, scripts are loaded by typing into the command box (load "example.script"). The hierarchy must be flattened, wherein the fill layers are added to normal layers, and in the 3D layout a land-to-cut and cut-to-land are combined with vias and equivalent as mask layers. Doping layers are also removed as they are thin and are assumed to have minimal thermal impact. Other consolidation takes place in this step as well.

Cadence was used for all extractions in this work. PDK and OpenAccess environment variables from OpenEDA must be setup. The script findxistor.il determines the device weight, the device count, and the power numbers. It selects the polysilicon gates and assigns
Figure 4.7: A capacitive thermal grid on a separate scale from the resistive thermal grid. The thermal capacitance is lumped together for each cell.

dynamic power, which is input from HSPICE simulations.

The shell script is executed in the following manner,

```
run6.sh Lib/cell/view_size(5.4)
```

For example, an extraction for this work used:

```
./run6.sh cornell_pp_06_11_29_chopped_upper_right_2col chip_m2 layout_processed 5
```

The output of this script is the text of a command for the `x6r.py` script. An example output is:

```
nice +19 ~/local/wirex/x6r_pwr.py cornell_pp_06_11_29_chopped_upper_right_2col chip_m2 layout_processed power 5 | grep -v ERROR | perl ~/local/wirex/other/topetsc.pl >!
```

Backslashes in the above code denote shell line continuations. In the phrase `power 5 | grep`, the ‘5’ specifies the resolution in micrometers of the resistive matrix, which can be changed by hand. The rest of the code after the pipe is not needed for this work, and a customized `x6r.py` script which was written for fREEDA formatted netlists is used. So from that result, the following command was used for extraction,
In this study the resistive matrices are directly connected to fREEDA netlists, thus the commands after the pipe are not needed. `x6r.py` generates the six resistor unit cells used in the netlist.

The run directory is entered where the `cds.lib` with the environmental setups. Running the script without modification will output the modified nodal admittance matrix (MNAM) directly, and enabling `donetlist=1` will output the netlist. Output must be piped to an output file in order to be saved.

To chop a selection of the layout the following menu options are followed: Edit > Basic > Chop. The entire layout is selected by shift-c, and in the libmanager window the following is entered into the command prompt, `(load "chop.il")` and then, `chopFromSelectedShapes(15)`, where `15` is the distance from the selected area in micrometers.

Figure B.1 shows a graphical representation of the grids generated using the script found in Appendix Section B.1.
Figure 4.9: The 3DIC stackup of extracted thermal netlist. The heat flux generated from circuit is injected into the active layer.

4.7.2 Netlist Inclusion

The extracted physical thermal netlists are included in the main circuit netlist as a subcircuit, by use of the .inc function in fREEDA. This allows for modularity in design. With a separate file for the thermal netlist, substitution of a re-extracted netlist is trivial. For example, higher or lower resolutions, thicker metal layers, or a different area on the die have all been separately simulated. For the principal hotspot simulations of the four hottest and fastest switching cells, the subcircuit is included by, .inc hotspot_extracted_101x54_10u.net. Named as such because its physical dimensions are 101 µm by 54 µm with a 10 µm resolution. This resolution was chosen to provide a simple 10×20 temperature simulation grid. A 5 µm resolution grid was also created, and named correspondingly. Remainder overhang is discarded by the extractor, and edges may be ignored when the original dimensions are divisible by the resolution. Once the file has been included into the netlist, the actual connects are interfaced.
Included within the ellipses are 98 similar lines connecting the 5 by 10 nodes of the 10 \( \mu \)m resolution grid for each of the following layers, handle\(_1\), active\(_c\), active\(_b\), active\(_c\), and overglass. While important, these have been omitted for brevity here. The lists of the correct node names to for use in the netlists are generated by a script. In the hotspot\_extracted_-101x54_10u.net file, the subcircuit is defined and connections are made in the following way:

```
*4 connections each: activeC, act_B, act_A, handle1(thermal gnd); overglass
.subckt thermal_network 2031 2013 2017 2035 1481 1463 1467 1485
+681 663 667 685 31 13 17 35
+ 2201
+ 2202
...
```

The connections listed in the subcircuit continue as listed above in order to interface the main circuit with the nodes in the extracted thermal mesh. Since there is a thermal terminal for each electrothermal macromodel, and all of these must interface from the frequency dividers and multipliers to the extracted netlist, those connections are made in the macromodel instantiations, rather than in these subcircuit listings. Therefore, those node numbers must be found and removed from the connection list, and care must be taken to ensure that each file has the list in the same order. Otherwise, among other problems including simulation failure, the end simulation result graphs will be distorted or altogether incorrect since the physical connections will not accurately represent the real world die.
4.8 Complete Electrical-only Simulation

The complete chain of 15 frequency multipliers followed by 10 frequency dividers was successfully electrically simulated in order to verify correct electrical output. The electrical models included custom and standard cells. Simulated data was saved for each stage between cells. In the following figures the change in frequency is shown to increase to a maximum proceeding through the multipliers, and then to decrease by half proceeding through the frequency dividers.

Figures 4.10 through 4.13 show the electrical through a selection of the nodes between the 15 frequency multipliers showing the correct operation. Figure 4.14 shows the output of the first frequency divider after the chain of frequency multipliers, which operates at 1 GHz. Figures 4.15 and 4.16 show more intermediate node voltages between the frequency divider stages, and finally Figure 4.17 shows the output of the last frequency divider, which is in agreement with the output measured from the circuit which will be discussed in Section 6.2.
Figure 4.10: The electrical output of the first frequency multiplier. The scale factor on the horizontal axis is $10^{-5}$.

Figure 4.11: The electrical output of the second frequency multiplier. The scale factor on the horizontal axis is $10^{-5}$.
Figure 4.12: The electrical output of the 7th frequency multiplier. The scale factor on the horizontal axis is $10^{-5}$.

Figure 4.13: The electrical output of the 14th frequency multiplier. The scale factor on the horizontal axis is $10^{-5}$. 
Figure 4.14: The electrical output of the first frequency divider. The scale factor on the horizontal axis is $10^{-5}$.

Figure 4.15: The electrical output of the second frequency divider. The scale factor on the horizontal axis is $10^{-5}$. 

68
Figure 4.16: The electrical output of the eighth frequency divider. The scale factor on the horizontal axis is $10^{-5}$.

Figure 4.17: The electrical output of the tenth and last frequency divider. The scale factor on the horizontal axis is $10^{-5}$.
4.9 Complete Electrical and Thermal Resistive Simulation

The netlist used in Section 4.8 was modified to use electrothermal elements, with the thermal terminal of each cell connected to a thermal resistive value extracted for each cell. Difficulties using an asynchronous circuit of this size, and for the length of simulation time required to see rises in temperature have prevented direct simulation even at the cell-level. For this reason, results were obtained by simulating the entire frequency chain for a sufficient amount of time to validate electrical output and to accomplish simulation results for an initial transient temperature rise for each cell. At this stage, the circuit was broken into two sections; the forward half including the complete frequency multiplier chain, and the rear half including the complete frequency divider chain. These sectional circuit netlists result in solutions which match the initial temperature rises indicated from the complete electrothermal resistive simulation. Results for this section were achieved by combining simulation results in this manner. The complete electrical and thermal resistive simulation netlists are given in Appendix C.3. In addition to simulating the complete electrical behavior, the final temperatures for each cell at the active layers were recorded.

The proper thermal resistivities were calculated by building a netlist which injects heat flux into the active layer of the 5 \( \mu \text{m} \) resolution extracted thermal resistive network.

![Diagram](image)

Figure 4.18: A heat flux is injected into the active layer of the full 5 \( \mu \text{m} \) resolution extracted resistive thermal netlist. Temperatures are then measured at the overglass and active layers to calculate thermal resistances.
With a known current source acting as a heat flux source, and temperature simulation data recorded from the active and over glass surface layers, Ohm’s law is applied to determine the effective thermal resistance for different points across the die. The extracted thermal resistance according to this method was determined to be \( 89225.4 \, \Omega \). This was the number used as the extracted thermal resistance in the complete electrical and thermal resistive simulation. Different locations around the die were determined to be within \( 10 \, \Omega \) of this number. This is due to the fact that 96.7% of the heat path to thermal ground is dominated by the homogeneous silicon substrate, as discussed in Section 4.3.1.

![Thermal Map](image)

Figure 4.19: The full chain of frequency multipliers and dividers was simulated to find final temperature levels. Dimensions of the thermal map are \( 231 \, \mu m \) by \( 126 \, \mu m \).

In the simulation model it was necessary to include a small thermal capacitance in order to achieve steady state results rapidly. Other approaches used average power as thermal input,
however this model uses the more physically-accurate transient heat flux. If the thermal network was resistive-only, the heat flux would immediately dissipate as it is generated and not be stored as temperature. In the electrothermal simulation produces results like those shown in Figure 4.19 the hottest section of the complete circuit is the region containing the four fastest switching cells, which are centered in the bottom center portion of the tile. These four cells are the second to the last frequency multiplier, last frequency multiplier, first frequency divider, and second frequency divider in the chain. This simulation is in agreement with steady state thermal measurements which will be discussed in Section 6.2. An average temperature rises of 8.5 °C is seen in the area surrounding the hotspot.

As applied to a general circuit, this step in the tool flow can potentially be used to identify particular areas needing higher resolution electrothermal simulation. In this case, the four fastest switching cells are manifested as a hotspot in the resultant thermal map and are selected for further detailed simulation.

The steady-state thermal values captured for each custom cell in this simulation can be used as device parameters for a more accurate iteration of the electrical only simulation. The electrical simulation results at the nodes connecting to the hotspot were used as the test signal for input to the next simulation iteration in the tool flow: the hotspot dynamic electrothermal simulation.

### 4.10 Hotspot Dynamic Electrothermal Simulation

By use of the frequency multiplier repeater and frequency divider eater electro-thermal macro-models, the overall required simulation time has been drastically reduced. An individual fREEDA element was created for each of these cells. This task is not necessary with clocked standard cells, however it is easy to quickly develop new elements for custom designs. Transistor level electrical-only simulation of a time interval of the power-on thermal transient over 0.5 ms of an entire die in HPICE is calculated to take 255 years. The electrical-only macro-model with extracted netlists method is estimated to reduce this to only 220 minutes in fREEDA. The
results presented here are the electrothermal simulation of two hottest frequency multipliers and two hottest frequency dividers. The test signal input to this circuit was acquired from the full electrothermal simulation in Section 4.9 at at the corresponding nodes. The resulting run time was 9.5 days. Simulations were carried out on an 8 core 64-bit 3 GHz Intel Xeon machine with 32 GB of RAM.

To simulate the fastest switching frequency divider cell, the electrical macro-model was attached to the extracted physical netlist from that area of the layout to form a heat source point. The resolution of the reduced resistive network is user-specified, and a 5 \( \mu \text{m} \) grid resolution was used for this example because this size will scale well with a full sized chip considering resolution and CPU time trade offs.

4.10.1 Hotspot Dynamic Electrothermal Simulation Results

The approach of attaching electrical behavioral models to extracted thermal netlists is extensible to any technology node or device, including devices fabricated in materials other than silicon. This would be accomplished by simple adjustments in the extraction script so that it is aware of the thermal parameters of the newly provided material. For example, high frequency MMICs in communication circuits may be GaN, SiGe, or GaAs. All of these common materials can be accurately simulated with no extra effort.
Figure 4.20: The transient rise of the hottest point of a cell on the overglass layer.
Figure 4.21: The time evolution transient simulation of the power-on temperature profile of the 3DIC averaged from 20 to 25 ns.
Figure 4.22: The time evolution transient simulation of the power-on temperature profile of the 3DIC.
Figure 4.23: Simulation results of each active layer and the overglass surface layer are graphed at 5 μm.
Figure 4.24: The time evolution transient simulation of the power-on temperature profile of the 3DIC.
4.10.2 Comparison of Thermal Network Resolutions

The electrothermal simulations of the 3DIC hotspot used in Section 4.10.1 were carried out using two separately extracted physical thermal resistive netlists as described by the procedure in 4.7.1. The WireX extraction resolution is user-defined and 5 μm and 10 μm resistive meshes were extracted from the same area chopped from the layout containing the four hottest and fastest switching cells. Since the different resolution resistive meshes have differing grid dimensions, each of the Matlab helper scripts in Appendix B had to be modified to be applicable to each case. The Matlab helper scripts provided the correct grid map for each layer and resolution, and the connections to the extracted thermal subcircuit.

Figure 4.25(a) shows the simulated thermal results of the over glass surface layer at a 10 μm resolution, whereas Figure 4.25(b) shows a separate simulation using a 5 μm extracted resolution. Good matching is shown between the simulations which result in equivalent peak temperatures with a more conforming, less blurred spatial map for the 5 μm simulation. The 5 μm resolution simulation is at the scale of a transistor-level mapping for this circuit yielding accurate spatial information as to where hotspots are generated. Depending on the size and density of the circuitry analyzed, an engineering trade off exists for the choice of simulation resolution. Lower resolution simulations may sufficiently depict problematic temperature regions, while avoiding the increased complexity of connection and increased parsing times of higher resolution extractions. Electrical output for each electrothermal hotspot simulation in this case was identical. However it is hypothesized that for more densely packed or temperature-sensitive circuits such as analog amplifiers, a higher resolution extraction would result in a more affected electrical simulation. The impact of the increased state variables from the higher resolution simulation is discussed in Section 4.11.
Figure 4.25: The thermal results of electrothermal simulation at the overglass layer of the 3DIC hotspot at 5 µm and 10 µm resolutions are compared.
4.11 Transient Computation

Run times were able to be greatly reduced by Kron state variable reduction method [72, 73], otherwise known as network condensation, where internal nodes are collapsed by matrix manipulations and only a dense matrix of external nodes are required. Figure 4.26 demonstrates how this happens from a circuit point of view. Table 4.3 shows that the state variable reduction is indeed effective as networks with 26020 resistive elements simulate just as quickly, or nearly as quickly as networks with 6160 resistive elements. The increase by a factor 4.224 in the number of resistive elements is due to doubling the resolution of the extracted thermal netlist.
from 10 \( \mu m \) to 5 \( \mu m \). The remaining 0.224 increase in resistive elements is due to the overhang
of the chopped layout since its dimensions are 101 \( \mu m \times 54 \mu m \).

The timestep, \( t_{\text{step}} \), was kept constant at 2.5 ps for all simulations in Table 4.3, regardless
of the simulation time, \( t_{\text{stop}} \), because it has been tuned to the frequency of changes in the
simulations for maximum efficiency of time, without error accumulation. The simulation using
the 10u_hotspot_rc_cond netlist has one thermal capacitance removed from the heat flux source
and moved into the resistive thermal network by one node. This has been done to ensure
a true linear element state variable reduction of internal elements, as edge elements are not
collapsed. Similar simulation times indicate that the same condensation is occurring in the
other simulations. Time-sliced and 3D thermal images here.

Figure 4.27 is a graphical interpretation of the data available in Table 4.3. Green and red
data sets for 5 \( \mu m \) and 10 \( \mu m \) resolutions show a high correlation between simulation times
further supporting the RC condensation state variable reduction. A divergence of 11.8% for
longer simulations is attributed to the longer parsing time of longer netlists, the time to write
four times as many output files to disk, or competition for the memory bus with other processes.
Each simulation was run on a dedicated core in a dual quad-core 64-bit Xenon system.
Table 4.3: fREEDA Transient Electrothermal Simulation Times

<table>
<thead>
<tr>
<th>Netlist:</th>
<th>Num. of R Elements</th>
<th>tstop</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10u_hotspot_short</td>
<td>6160</td>
<td>.4e-8</td>
<td>24208.67s (0.28 days)</td>
</tr>
<tr>
<td>10u_hotspot_rc_cond</td>
<td>6160</td>
<td>.4e-8</td>
<td>19021.183s (0.22 days)</td>
</tr>
<tr>
<td>10u_hotspot_M1x2_mid</td>
<td>6160</td>
<td>.3e-7</td>
<td>211239.77s (2.4 days)</td>
</tr>
<tr>
<td>10u_hotspot_mid</td>
<td>6160</td>
<td>.3e-7</td>
<td>202311.17s (2.3 days)</td>
</tr>
<tr>
<td>10u_hotspot_mid_C10</td>
<td>6160</td>
<td>.44e-7</td>
<td>453698s (5.25 days)</td>
</tr>
<tr>
<td>10u_hotspot_long</td>
<td>6160</td>
<td>.8e-7</td>
<td>712394.52s (8.24 days)</td>
</tr>
<tr>
<td>5u_hotspot_short</td>
<td>26020</td>
<td>.4e-8</td>
<td>21525.68s (0.25 days)</td>
</tr>
<tr>
<td>5u_hotspot_mid</td>
<td>26020</td>
<td>.3e-7</td>
<td>203100s (2.35 days)</td>
</tr>
<tr>
<td>5u_hotspot_long</td>
<td>26020</td>
<td>.8e-7</td>
<td>809570.04s (9.37 days)</td>
</tr>
</tbody>
</table>

Figure 4.27: CPU simulation time compared with the simulation stop time, tstop, for 10 µm and 5 µm resolution thermal networks.
4.12 Summary

This chapter introduced and described in detail the methods used in this body of research to generate the electrothermal transient simulation results that are in agreement with measurements. Behavioral macromodels of the basic frequency multiplier and divider cells present in the 3DIC have been created. The process used to extract the thermal physical netlist has been outlined. The WireX extractor splits the 3DIC stackup into layers with resistances based on the material’s thermal properties. The extracted physical thermal netlist is then attached to the electrothermal macromodels with the help Matlab helper scripts that are used to generate a spatial map as well as connection node lists.

Once electrothermal and physical thermal netlists are joined, a transient simulation is performed. A method for reduction of CPU time has been presented wherein the time-to-rise of the thermal circuits is reduced. Rapidly-rising transients may be translated by dilating their time axes by the same ratio that their time constants have been reduced.

The resolution of the discretized grid produced by the WireX layout extractor is user-defined. Resolutions of 5 $\mu$m and 10 $\mu$m have been simulated, which result in equivalent peak temperatures, but more conforming, less blurred spatial thermal maps for the 5 $\mu$m simulations. The increase in resolution also increased the number of elements present in the thermal netlist, however simulation times were shown to be equivalent, because an internal state variable reduction technique was applied.

This chapter described a cell-based approach utilizing macro-models to the electro-thermal simulation of three-dimensional integrated circuits. A 3DIC test circuit has been assembled to verify simulations, and thermally matching measurements of this 3DIC have been made to support the simulation method. The techniques may readily be applied to other designs in single or multi-tiered ICs. In addition to analog circuits, such as amplifiers, which have long required thermal considerations, digital circuits are on the threshold of the thermal barrier. This method could be easily applied to clocked digital circuits. The asynchronous nature of the circuit simulated in this chapter demonstrates the versatility of the method.
Chapter 5

Electrothermal Elements

5.1 Introduction

The ThermalBlockRC fREEDA element is an elegant solution to addressing the electro-thermal properties of an arbitrary block of material. The element consists of thermal resistances extending in the six directions of a three-dimensional block, and a thermal capacitance at the center of the block, which is connected to thermal ground. The naming convention of north, south, east, west, top, and bottom has been used to define resistivity directions within blocks. The capacitor which models heat capacitance is connected to thermal ground, i.e. absolute zero, within the model. This is physically accurate because the heat capacity, or thermal mass, material parameter is the product of the mass of material and the total amount of energy a material will hold per unit temperature. Ambient temperatures, typically 300 K are set by adding an electrically voltage-analogous DC temperature source to the boundary of the bulk. The ThermalBlockRC element is scalable to any size, and therefore can be used for different-sized cells in finite difference analysis. The thermal resistances and capacitance used are extrinsic properties dependent on the thermal conductivity, the specific heat capacity, and amount of material to be modeled. These elements are also extendable to 3DICs because they may be stacked vertically in a three-dimensional array. An example arrangement of a grid of ThermalBlockRC elements
is shown in Figure 5.4. This is a single layer of material that can be connected to other lay-
ers with the top and bottom resistors on the diagonal line. Electrical and thermal compact
elements exist within fREEDA which can be networked together and simulated.

The electrothermal simulation provided by fREEDA assumes continuity of heat flux at
discrete time steps corresponding to the time discretization required to model the electrical
circuit. The time step required is tens of picoseconds. After a power-on event or other thermal
transient, steady-state is achieved after a few seconds of real time, therefore tens of billions of
simulator time steps are required. In the electrical domain alone it has been established that
compact-modeling is essential. Compact models can be viewed as analytic macromodeling and
so it is essential that compact modeling be used in thermal modeling rather than a technique
based on volumetric thermal meshing which requires large matrix solutions. Different material
layers can be simulated in fREEDA by connecting a network of ThermalBlockRC elements or
alternately Batty-based ThermalBlock elements, as shown in Figure 5.1. The ThermalBlockRC
models exist in the thermal world and are linked in the netlist just as electrical circuits would
be described.

5.2 Suite of Electrothermal Elements for fREEDA

The family of electro-thermal elements for fREEDA are divided into three subgroups: RC-
based elements, Batty model-based elements, and active elements. The RC elements reduce
the amount of CPU overhead and memory required for simulation, while elements based on
the Batty model are highly accurate and mathematically elegant with differential heat equation
solutions. The active models include a thermal EKV transistor model as well as a selection of
thermal macro-models such as an inverter and NAND logic gates. Additionally, a standard cell
library was generated based on the FreePDK\textsuperscript{TM} \cite{74} process design kit. The FreePDK used
was based on the Open-Access PDK for the 45 nm technology node.
Figure 5.1: The fREEDA Approach to Electrothermal Modeling with ThermalBlocks connects elements by thermal terminals.

5.2.1 RC-Based Electro-Thermal Elements

RC-based elements offer an appreciable speed-up of computing over more complex models while capturing high fidelity thermal phenomena, such as the long tail effect (described in Section 2.5) if a significant number of RC elements are included.

5.2.2 ThermalBlockRC

The previously developed ThermalBlockRC element contains the thermal capacitance in parallel with one of the directional thermal resistances. While this can be a useful configuration for a single layer of thermal material, this is incorrect for a true volume of material, thus the ThermalBlockRC3D element, described below, was developed for this work. The ThermalBlockRC is a material model based on thermal capacitance and thermal resistivity material parameters. Resistors included in the model are arranged to geometrically express thermal resistances in
the north, south, east, west, top, and bottom directions such that a three-dimensional cube is represented. This model is suitable for single layers of material as the $R_b$, the bottom resistor and $C_{bulk}$ share a common node.

5.2.3 ThermalBlockRC3D

The ThermalBlockRC3D is a material model based on thermal capacitance and thermal resistivity material parameters. Resistors included in the model are arranged to geometrically express thermal resistances in the north, south, east, west, top, and bottom directions such that a three-dimensional cube is represented. This model is extendable to any material to be
modeled in a three-dimensional structure. This model differs from ThermalBlockRC in that $R_b$, the bottom resistor and $C_{bulk}$ are separated so that a three-dimensional matrix of the element may be used or connected to another element. The ThermalBlockRC3D is extensible to all constituents of the system, including package and heatsink material engineering. This is achieved by the addition of another ThermalBlockRC with appropriate material parameters. In this way chips and packages can be engineered to facilitate the most cost effective solution.

Figure 5.4: Connections for a plane of ThermalBlockRC3D elements which can be connected in 3D.
5.2.4 InterconnectRTSH

This element implements an interconnect line as an electro-thermal resistor effects. It is named RTSH to denote that it is a self-eating resistive thermally aware element. The fREEDA command .locate can be used in the netlist to specify the X,Y coordinates of the ends of the interconnect. Electrical and thermal parameters are automatically calculated. This is useful for simulating netlists parsed from layout databases. The resistive thermal interconnect is modeled as resistive only, the metal line is made by a kind of metal, which includes silver, copper, gold, and aluminum that are predefined in the model.

Three-dimensional layout of integrated circuits has advantages in routing [31]. In a 3DIC chip interconnects are embedded in a silicon layer. The interconnects are thermally insulated by layers of insulative BSG cap oxide where the separate silicon wafers are joined in the 3DIC process. The electrically insulative materials used to join the silicon layers is also thermally insulative, and thus presents a tremendous problem for effective heat removal. It is important to be able to simulate the effects of these layers on circuits and overall temperature. To address this challenge directly, the electrothermal interconnect model, InterconnectRTSH depicted in Figure
1(a) has been developed. The interconnect is based on an electro-thermal resistor, whereas it has two electrical terminals, n1 and n2, and two thermal terminals, n3 and n4. It is the typical approach in fREEDA for electro-thermal elements such as InterconnectRTSH to have both electrical and thermal terminals. Interoperability between existing elements is straightforward. Internally in the InterconnectRTSH element, the electrical and thermal quantities are coupled by the power equation,

$$R = \frac{\rho l}{A}$$  \hspace{1cm} (5.1)

And the thermally-aware version of this equation is,

$$R = \frac{\rho l}{A} \left[ 1 + \beta (t - t_0) \right]$$ \hspace{1cm} (5.2)

Where $R$ is the total resistance calculation based on the user-provided parameters for a given material for $\rho$, the resistivity; $l$ the length of the interconnect; and $A$, the area. Equation 5.2 is the electro-thermal resistance of the interconnect with temperature coefficient $\beta$. Resistivities for gold, silver, and copper are included in the element for convenience. The ability to customize any of the element’s parameters allows for simple tuning for process-specific characteristics, such as is necessary for degenerately doped polysilicon interconnects. Material properties for polysilicon have been known to even vary across the same wafer due to bad uniformity in PCVD furnaces. The length may either be given as an argument, or calculated using a netlist locate function. Figure 5.5(b) is a diagram of two connecting interconnects in the same metal layer. Layout rules require an overlap in the mask where there is a connection for DRC satisfaction, whereas after fabrication it is one coherent film. The locate command will eventually store positional data in order to determine heating effects of nearby active devices. In the current form, all heat transfer takes place through thermal terminals.

In order to automate future planned integration with other CAD tools, a new feature, the ‘.locate’ command has been added to fREEDA. The ability to specify node locations within
a circuit netlist so that elements may adjust their properties accordingly is novel and a vast improvement over other SPICE-like simulator engines.

The interconnects are designed to work with active elements, such as electro-thermal transistor models to form the inverter circuit shown in Figure 5.7. The circuit shown in Figure 5.7 contains ten heat-producing elements, eight passive interconnects and two active transistors. Along with ThermalBlockRCs used to model bulk silicon and heatsinks, this paints an accurate picture on the device level. Alternatively, a separate inverter model with thermal terminals has been created, Cmosinv, that can be paired with the interconnects to model the power rails and logical connections present in ICs. Internally, the thermal inverter is modeled by a current source supplying the heat flux, which is matched with the currents flowing though the NMOS and PMOS transistors, including leakage currents. One thermal node of a thermal element should be connected to a thermal reference, or thermal ground. In this case, the thermal ground can be thought of as a perfect heatsink. The ThermalBlockRC, modeling the silicon bulk is then connected to the remaining thermal terminal. Any number of ThermalBlocksRC can then be added in series to simulate whatever layers of thermal material may exist in a device, such as IC package material, and the heatsink.

![Thermal interconnect circuit](image)

**Figure 5.6:** Example of the thermal interconnect circuit.

The a test circuit was constructed to model thermally aware inverters with the Intercon-
In Figure 5.8, two separate interconnects of differing length are driven by an inverting signal for a basic proof of functionality of fREEDA’s ability to make use of spatial information read in from the netlist. It can be seen that the temperature is higher in the longer interconnect because it has a higher thermal resistivity. Due to the presence of a heat capacity of the interconnect material, it can be seen that the resting temperature of the longer interconnect is higher than the shorter interconnect. Figure 5.7 depicts a typical inverter VSLI layout with metal interconnects in blue. Multiple metal and polysilicon layers of interconnect used in VLSI layout design can easily be implemented for more accurate physical representations of inverters. This is a standard inverter layout provided by FreePDK, an open source, Open-Access-based PDK for the 45 nm technology node in accordance with the ITRS road map. The standard inverter is the goal for this path of simulation as it is the basic building block for all logic and can easily be scaled to a larger project.
Figure 5.8: Transient simulations of InterconnectRTSH elements using XY locational information as parameters.

Figure 5.9: Simulated change in resistance along an interconnect as dependent on current and temperature.
The results shown are the heating effects of the interconnects themselves, which is an important part to understanding the entire thermal picture. Heat flux is generated by the current flowing thought the metal layer interconnects when CMOS transistors in the inverter switch. The fREEDA interconnect element has been designed so that they are scalable to any number of modeling needs. Thermal terminals on the InterconnectRTSH element can be connected to an \( RC \)-model or the Batty-based ThermalBlock elements in order to simulate the bulk material.

5.3 Batty Model Based Electro-Thermal Elements

The Batty model [44, 45] is a quasi-analytic approach to compact thermal modeling utilizing Fourier transformations and symmetry to solve the heat diffusion equation. The Batty model is able to quickly solve heat equations after an initial building of matrices. This pre-calculation of the impedance matrix, \( R_{THij}(s) \), is typically computationally more time consuming than the transient simulation.

5.3.1 ThermalBlock

![Diagram of ThermalBlock Element]

Figure 5.10: The symbol and nodal connections for the ThermalBlock Element. After [75].
Figure 5.10 shows the ThermalBlock thermal element which is a vertically matched Thermal $n$-port generated directly from analytical solution of the heat diffusion equation. An N-port element is scalable in order to accommodate $n$ number of nodal connections. The thermal element is represented by thermal impedance matrix $R_{THij}(s)$, with varying temperatures at the top and bottom surface.

### 5.3.2 ThermalHeatsink

![ThermalHeatsink Diagram](image)

Figure 5.11: The symbol and nodal connections for the ThermalHeatsink electrothermal fREEDA element. After [75].

Figure 5.11 shows the thermal element model of a heatsink-mounted thermal $n$-port which is generated directly from a simplified analytical solution of the heat diffusion equation. The simplification arises from the assumption that one side of the thermal body will be held at a constant temperature. The ThermalHeatsink element is represented by the thermal impedance matrix $R_{THij}(s)$, with varying temperatures at the top surface only. The entire bottom surface is considered to be uniform. This element differs from the ThermalBlock element in that it
Thermal Via

Figure 5.12: The symbol and nodal connections for the ThermalShunt element. After [75].

has been designed to always connect to a thermal ground (which acts as a heatsink) on one side of the material block. Only one node is provided for the heatsink connection, though it is considered as a contact for the entire side of the body of thermal material. The reduction in terminals over the ThermalBlock element results in a considerable reduction in complexity of the heat equation and results in a reduction in required computation time. Equation 2.13, the heat diffusion equation, and Equation 2.14, the boundary heat equation, both become invariant in the $x$ and $y$ dimensions on the right hand sides of the equations. This element is suitable for use as bulk silicon, die packaging material, or at a coarser level, a physical metal heatsink.

5.3.3 ThermalShunt

Figure 5.12 shows the thermally-aware shunt which is modeled by a vertically matched thermal 2-port. Following the same progression of simplification from the ThermalBlock to the ThermalHeatsink element, the dimension of lateral variance has been reduced for both sides of the ThermalShunt element. This simplification is responsible for the change from an $n$-port of the more complex elements to the 2-port. The ThermalShunt element is represented by a thermal impedance matrix, $R_{THij}(s)$ with varying temperature at the top and/or bottom surface. Only one heat generating device should be connected to this element due to the fact that there is no other outlet for the heat flux, which will lead to ill-conditioning of the matrix.

The ThermalShunt element should be used whenever only a single dimension of temperature change is considered. This may be for a thermal via structure in an IC, a lateral interconnect,
or potentially an entire two-dimensional layer in a thermal analysis. The ThermalShunt offers considerable reduction in pre-calculation overhead because of the reduction of heat spreading in two dimensions on each side of the thermal material, therefore it should be used whenever the level of detail required permits and computation time has become intractable. It is for these reasons that the ThermalShunt is just as flexible as the ThermalBlock element in that it can be used to model a die package or a physical heatsink, in cases where it is assumed that temperature will be invariant across the surface of the thermal material. By the time heat diffuses from an active layer of semiconductor through a silicon handle of typical thickness and encounters a package or metal heatsink, lateral diffusion is likely at equilibrium.

5.4 Comparison of the Batty Model with an RC Thermal Model

Electrothermal simulations of a passive thermal material, in this case silicon, using two different models are compared in this section. The Batty-based ThermalHeatSinkSpot is used as it accepts an internal spot or single terminal point heat injection from an electrothermal element. The element simulates an $n$-port grid array substrate with $n \times n$ surface heating elements; there is no equivalent SPICE element. The temperatures at the thermal terminals of the element are calculated by Fourier-space solution of the heat equation.

To compare the functionality of the two elements the thermal terminal of the electrothermal inverter element was connected to each material model block separately. The schematic for these electrothermal circuits is related in Figure 5.13, where either the Batty model or the thermal $RC$ model is used for the ThermalBlock. The netlist for the batty netlist is given in Appendix C.1.1. Similarly, a comparable netlist was built using the six resistance cube with a thermal capacitance model. The $RC$-based thermal netlist can be referred to in Appendix C.1.2.
Figure 5.14 shows the digital output of the electrothermal CMOS inverter element which is attached by its thermal thermal to either a Batty thermal block or an $RC$ thermal block in these examples. The rises and falls in the output coincide with the spikes in heat flux shown in Figure 5.15. The heat flux from the Batty model and the $RC$ simulations were identical. Figure 5.16 shows the heat rise of the silicon substrate at a small time scale so that short temperature spikes are visible. Figure 5.17 shows the same simulation, however using the 6-$RC$ resistive cube model. The same texture to the temperature rise spikes is visible, with the difference of some shape of the curves. The Batty model temperature rises are thinner, with the area under their curves amounting to less. This results in less energy transfer over time which is evident by a slowed rise to steady-state compared to the $RC$ circuit. This rise to steady-state for the Batty-based model is shown in Figure 5.18, while the same event for the $RC$-based model is shown in Figure 5.19. Due to differences in temperature calculation by the models, they are not one-to-one compatible and some parameters were adjusted for better model comparison and axis adjustment, as visible in netlists used to generate these results, which can be referred to in Appendix C.1. For the reasons stated above out the differences in the shape of the small time scale temperature spikes between the models, the long time scale
Figure 5.14: The digital output of the electrothermal CMOS inverter model is verified to correspond with the heat flux resulting from dynamic switching.

Figure 5.15: The simulated heat of the electrothermal CMOS inverter attached to the Batty thermal block.
Figure 5.16: The simulated heat rise of a CMOS inverter using the Batty model for a block of silicon.

Figure 5.17: The simulated heat rise of a CMOS inverter using an RC cube model for a block of silicon.
Figure 5.18: A long simulation of the heat rise of a CMOS inverter using the Batty model for a block of silicon.

Figure 5.19: A long simulation of the heat rise of a CMOS inverter using an $RC$ cube model for a block of silicon.
simulation of the Batty-based model takes a longer time to rise to steady state (exhibiting the long tail effect) when compared to the $RC$-based model.

The simulation data presented in Figure 5.19 does not exhibit the long tail effect as Figure 5.19 does. This difference is due to the fact that only one pole in the $RC$ circuit used. It is expected that as more poles are gained by adding additional $RC$ cubes to the circuit that the simulated results will converge to a common long tail rise.
5.5 Summary

This Chapter presented a discussion of the existing Batty-based electrothermal elements and RC-based electrothermal elements which were created to further abilities in simulation. The active electrothermal elements used to build the circuits in this work were also presented including the CmosinvT inverter element. The electrothermal simulation of the inverter can be further elaborated upon by adding interconnect elements.

RC-based thermal models allow Kron linear element reduction to be applied to simulated networks. The immediate benefit of fREEDA RC reduction is the reduction of CPU time from many hours to a matter of minutes. This is contrasted to the Batty model [44], where the majority of time is the pre-calculation when matrices are created and filled, rather than the actual solving time.

This chapter described an RC block approach to the electro-thermal simulation of three-dimensional integrated circuits. Special attention has been afforded to the problematic metal interconnect. The Interconnectrtsh element has been described to address this problem, which is able to use locational data provided in the netlist for geometric calculations. The aim is to allow engineers to effectively design around critical temperatures and thus maximize performance. Extensible, scalable electro-thermal tools have been shown to work over a range of simulation applications to facilitate the needs of designers.
Chapter 6

Electrical and Thermal Measurements

6.1 Introduction

In the past, the thermal profile of integrated circuits could not be measured directly with infrared thermal imaging due to inequivalent emissivities of materials. It was necessary to use an absorptive ink to approximate a blackbody so that the infrared emissions can be used to infer temperature. The impact and effect of this thermal imaging technique was investigated [76], by comparing measurements with detailed thermal simulations with and without the surface treatment.

Infrared thermal imaging is commonly used to experimentally extract the thermal profile of ICs. A multitude of materials can be found within ICs, including metals, Si, and SiO$_2$, each presenting a different emissivity, $\epsilon$. With known or calibrated $\epsilon$’s, the infrared intensity is directly related to the device temperature. The reliability of thermal imaging is dependent on the properties, including emissivity, of the surface materials and measurement environment [77, 78, 79, 80, 81]. The reliability and usefulness of thermal imaging can be improved dramatically by using a black ink coating with the intention of creating a superficial blackbody. A theoretical
blackbody has $\epsilon = 1$, while real objects have $\epsilon < 1$. In this case, a uniform $\epsilon$ across the surface is more important than blackbody approximation. The ink is a composite with the major component being carbon black that exhibits both scattering and conduction loss at infrared frequencies. The typical spectral range of infrared detectors used with ICs is 2-5.4 $\mu$m, the skin depth in the carbon black composite is hundreds of microns, while the thickness of inks tends to be in the range of 1-5 $\mu$m. Having low conductivity, there will be transmission of infrared through the ink and so it can only approximate a blackbody. As a result the effective emissivity from the surface of the ink has a dependence on the underlying material.

For the measurements presented in this chapter, an InfraScope™ made by Quantum Focus Instruments Corporation was used for imaging. The InfraScope applies automated emissivity correction to microscopy, which eliminates the need for a uniform blackbody at the surface of the material. The photon flux density emitted at a surface is a cubic function of the temperature. Therefore, the results of imaging a bare uninked substrate are higher resolution, more accurate, crisper measurements with less heat spreading. The spectral response of the apparatus is 2-5.4 $\mu$m (mid-wave IR), a pixel resolution of 1.6 $\mu$m and a temperature sensitivity of 0.1 degrees at 80°C.

Typically in optical microscopy, an object is illuminated and reflected rays are focused by the optics to produce a real image. In IR microscopy photons emitted directly from the object imaged are focused by the objective optics. In a dark room there is a tremendous amount of infrared wavelength photons emitted, which are not visible to the human eye, so thermal imaging inherently involves a high degree of background noise. This is another benefit to using the heated chuck for heating the sample well above room temperature. This is so effective that an isolation shroud around the microscope enclosure is not necessary. Thermal measures taken with the shroud open and closed yielded the same results.

Glass, which is traditionally used in optics, is opaque to infrared spectra. Often salt is used in infrared spectroscopy, but NaCl is highly hygroscopic and requires special storage in vacuum desiccators. This is not possible with microscope objectives because they are precisely
machined, as well as mounted. To allow for transmission of infrared wavelengths, custom microscope objectives are implemented which have been engineered from materials such as silicon, germanium, sapphire and zinc selenide. An indium antimonide (InSb) CCD detector is cooled with liquid nitrogen. InSb is a photovoltaic narrow-gap semiconductor with energy gaps at 0.17 eV at 300 K and 0.23 eV at 80 K, and by the de Broglie relations,

\[ \lambda = \frac{h}{\hbar k} \]  

\[ E = \hbar \omega, \]  

thereby emits current when impinged by electromagnetic radiation in the infrared spectrum range. In the above equations \( h \) is Plank’s constant, \( \hbar \) is \( h/(2\pi) \), \( E \) is energy, \( \lambda \) is wavelength, and \( k \) is the angular wavenumber.

### 6.2 3DIC Electrical Measurements

Electrical testing took place on a hotplate heated to 80°C, which was the background temperature used during thermal measurements. The background temperature can potentially effect functional frequencies in circuits. As discussed in Section 2.3.1, conductivity is a function of temperature. Asynchronous circuits can be sensitive to changes in operating temperature due to increases in interconnect resistivity and decreases in semiconductor mobility. The hotplate temperature was verified by a contact thermocouple and an IR thermometer.

Measurements were taken at header pins installed on the PCB with 50 Ω test probes and a Tektronix TDS3052 real-time sampling oscilloscope. There are two modes of operation for the chip, single-trigger events, and free-run mode. A push button was installed which would be used to generate a single token. The chip worked well in free-run mode, so this was the mode used for all measurements. A single pulse output would not be easy to capture electrically, and would
Figure 6.1: The setup for electrical measurements at temperature of the 3DIC.

Figure 6.2: Thermal imaging of the 3DIC in situ at Wright-Patterson Air Force Base.
not generate useful thermal information. A discrete NAND IC was installed for debouncing signals before arriving at the chip as a trigger event. To trigger free-run mode, a side switch on jumper J16 is slid to the on position. For stabilization during thermal measurement in situ, where small movement would greatly affect the calibration of the microscope, wire leads were attached to the pins of the slid switch so that electrical connection could be made away from the board, triggering the token generation.

Table 6.1 lists the output frequencies measured at the header pin corresponding to each tile. These average of the frequencies is 598.32 Hz, which correlates to halving of the frequencies ten times by the chain of frequency dividers from the maximum internal frequency of 1 GHz. A benefit of the design of the outputs preceded by frequency dividers is that RF test probes and equipment is not necessary. The standard deviation is $\sigma = 73.34$, which can be attributed to variation in fabrication processes of the wafers making up the different tiers, as well as variation across the individual wafers in processing.

6.3 3DIC Thermal Measurements and Comparison

Quantum Focus has employed patented and proprietary algorithms in their imaging software as well was the TMViewer$^\text{TM}$ software, which applied to each pixel data individually. These compensate for the efficiency in photon emission of the differing materials and generates the contrast in the thermal map images. A high degree of temperature sensitivity and resolution is possible with the InfraScope. Due to the texture of the surface, there are some reflections where the surface is not normal to area of the die. There may be reflections, or the emissions are not captured by the CCD.

6.3.1 Steady State Thermal Measurements

Each of the nine tiles of the uncapped 3DIC was imaged using the InfraScope thermal microscope. Thermal linearity was confirmed by increasing observing temperature rises with two
Table 6.1: Measured frequencies of tile electrical output.

<table>
<thead>
<tr>
<th>Tile Position</th>
<th>Tier</th>
<th>Frequency (KHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>0 1</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>0 2</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>1 0</td>
<td>A</td>
<td>802</td>
</tr>
<tr>
<td>1 1</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>1 2</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>2 0</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>2 1</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>2 2</td>
<td>A</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>0 0</td>
<td>B</td>
<td>540</td>
</tr>
<tr>
<td>0 1</td>
<td>B</td>
<td>498.5</td>
</tr>
<tr>
<td>0 2</td>
<td>B</td>
<td>570</td>
</tr>
<tr>
<td>1 0</td>
<td>B</td>
<td>513.8</td>
</tr>
<tr>
<td>1 1</td>
<td>B</td>
<td>Nonfunctional</td>
</tr>
<tr>
<td>1 2</td>
<td>B</td>
<td>587.2</td>
</tr>
<tr>
<td>2 0</td>
<td>B</td>
<td>515.5</td>
</tr>
<tr>
<td>2 1</td>
<td>B</td>
<td>509.7</td>
</tr>
<tr>
<td>2 2</td>
<td>B</td>
<td>575.7</td>
</tr>
<tr>
<td>0 0</td>
<td>C</td>
<td>594.5</td>
</tr>
<tr>
<td>0 1</td>
<td>C</td>
<td>591.5</td>
</tr>
<tr>
<td>0 2</td>
<td>C</td>
<td>663.8</td>
</tr>
<tr>
<td>1 0</td>
<td>C</td>
<td>610.5</td>
</tr>
<tr>
<td>1 1</td>
<td>C</td>
<td>629.5</td>
</tr>
<tr>
<td>1 2</td>
<td>C</td>
<td>651.8</td>
</tr>
<tr>
<td>2 0</td>
<td>C</td>
<td>633.9</td>
</tr>
<tr>
<td>2 1</td>
<td>C</td>
<td>647.3</td>
</tr>
<tr>
<td>2 2</td>
<td>C</td>
<td>634.6</td>
</tr>
</tbody>
</table>
different background temperatures, with the heated chuck between 80°C and 100°C. The temperature rise of hotspots was consistently 9-10°C above the applied background temperature, confirming consistent power output over those that range, and the accuracy of the emission algorithm performed by the TmViewer software.

While there was no way to control which tiles were operational, some tiles did exhibit electrical failure due to previous burn-in testing by Akopyan et al. [61] in a study on processing variability between tiers. Circumstantially, this allowed for observation of functional circuits on tier C while the underlying circuits on tier A or B were non-functional. While the token generators are relatively low power circuits, which are spaced 6340 nm, there has been an appreciable difference of less than a degree. The fastest frequency repeater and frequency divider cells on the lower cells A and B, also have a less measurable effect on the surface overglass because they are positioned closer to the heatsink. For this reason, the functionality
Figure 6.4: The white box is the area selected in software to be averaged over time in the transient measurement.

and thermal impact of tier A is less measurable than tier B, while tier C is directly measurable.
Figure 6.5: The white box is the area selected in software to be averaged over time in the transient measurement.

Figure 6.6: The white box is the area selected in software to be averaged over time in the transient measurement.
Figure 6.7: Thermal measurements of the hotspots of individual tiles are compared with their near-IR images and layouts.
6.3.2 Variance in Three-Dimensional Heat Sources

It was observed that the temperature was cooler in the middle tile in thermal measurements, as shown in Figure 6.8. Electrical measurements taken from tier B, the center tier, presented in Table 6.1 confirm that this site was non-functional. The fact that a center layer was non-functional for a measurement site was taken advantage of to further prove simulation agreement. The hotspot dynamic electrothermal simulation presented in Section 4.10 was modified not to include tier B to produce the results shown in Figure 6.9. In simulation results tier B shows lower spread out temperatures conducted from tiers A and C. Tier C shows a lower overall temperature than a nominal tile with a functional tier C in agreement with measurements.
Figure 6.9: Electrothermal simulation results with tier B deactivated matches measurements of a tile with a non-functional tier B.
6.3.3 Transient Thermal Measurements

A video capture mode of the software was utilized to capture transients of each tile. Figure 6.11 relates the measured front end of the transient is represented in order to match simulations. This is after the power is turned on, and $V_{dd}$ is applied to the chip, wherein there is a slight temperature rise, likely due to leakage currents. The token generator circuit was triggered \textit{in situ} once the recording had started.

![Figure 6.10: The white box is the area selected in software to be averaged over time in the transient measurement.](image)

Figure 6.10: The white box is the area selected in software to be averaged over time in the transient measurement.
Figure 6.11: The measurement of a thermal transient of a hotspot on the 3DIC averaged over a square area of pixels.

Figure 6.12: The simulated thermal transient rise of a the hotspot of the 3DIC matches measured data as it nears steady-state.
6.4 Summary

Electrical measurements confirmed correct functionality of the majority of the chip’s tiles. Each identically replicated token generator circuit tile was measured on each tier. Some tiles on the lower tiers were found not to be functional, as shown in Table 6.1. Previous work by Akopyan et. al. [61] in fabrication variability permitted chip burn-in to disable some repeated circuits on tiers A and B, so that the differences could be measured. These non-functional locations were used for further verification of simulations, where tier B was simulated as non-operational. Electrical measurements were performed at ambient temperature as well as raised to an 80°C background to explore frequency dependence on temperature. Thermal mapping was an important component of model development due to the need for validation of accuracy.

The combination of electrothermal models and an extracted physical thermal netlist have been shown to be accurate within 15% of measured data. Volumetric effects of heat sources where measured and compared against simulated results. Thermal mapping has confirmed that due to dynamic power dissipation, the majority of the heat flux is generated in the two hottest cells of the token generator circuit. The impact of heat sources existing throughout the volume of the 3DIC die (excepting the silicon handle) has been explored.

For the first time, temporal and volumetric transient simulations have been matched with measurements. While thermal measurement by imaging is not possible beyond the overglass surface of the die, the surface measurements which were performed match with electrothermal simulation results of the surface. At-depth electrothermal simulations through each layer of the 3DIC show temperatures to rise higher than would otherwise be expected considering only surface measurements. Transient simulations in particular show peak temperature intensities to rise sharply above the average. Furthermore, simulated transient temperature spikes cannot be directly measured because of the timescales in which they occur. The temperature spikes are important considerations with regard to performance, aging, and reliability.
Chapter 7

Conclusions and Future Work

7.1 Summary of Research and Original Contributions

This dissertation presented a new method of dynamic electrical and thermal co-simulation. Contributions to the abilities of spatial thermal modeling in three dimensions have been made. The electrothermal analysis presented used the finite element method and a reduced-order model based on cuboids with accurately extracted physical thermal conductivities. The end goal was to provide a general simulation tool to designers, which could be extended to a broad selection of projects which require attention to thermal performance. The work described here has resulted in the first availability of spatial and temporal simulation data. This highly detailed information is indispensable during the design process of VLSI systems. Furthermore, in addition to the surface temperature, temperatures throughout the active layers of a 3DIC can be explored with this approach. A novel technique for combining an accurate extraction of physical thermal data with a behavioral macromodel of a thermally-aware circuit has been outlined.

A cell-based approach to electrothermal modeling was developed and implemented. The complexity of the electrical circuit used in this experiment mandated that cell-based macromodels be used to solve an otherwise intractable problem, which was further complicated by
coupling with thermal models. The asynchronous circuit used in simulations was originally designed using an automated behavioral synthesis, and not designed at the individual transistor level. This synthesis using behavioral models lead to a natural circuit hierarchy that could be exploited in analysis. The wide range of frequencies present in this circuit made it a good simulation case in order to verify differences in temperature due to dynamic power, but it required long simulation to verify correct output. Thus the example used was a significant test of the dynamic electrothermal simulation procedure.

The Kron-like [72] state variable reduction of internal linear elements (implemented in the fREEDA SVTran2 analysis [82]) was used in this work. Large electrothermal simulations that use extracted physical thermal resistive networks, such as the work presented in this dissertation, are excellent candidates for the increased performance provided by state variable reduction. Test cases using 5 μm and 10 μm extracted resolutions contained 26,020 and 6,160 resistive elements respectively, which were both simulated for CPU times of six hours to 9.4 days. Results showed that the number of internal linear elements did not adversely affect simulation time thus demonstrating the importance of state variable reduction for large thermal circuits.

Electrothermal elements were designed and implemented in simulations. A locationally-aware interconnect model was designed which can read in geometric information from a netlist and calculate its thermally-dependent electrical resistivity as well as power dissipation in terms of temperature. This model represents an important step toward linking circuits, which rely only on connection graphs, to the physical domain which is geometrically dependent. Additionally, the ThermalBlockRC3D model was developed to model blocks of thermal materials in three dimensions. This model is extensible in application because many cuboids with small resistance values can be used to discretize a small volume in high detail, or larger cuboids can be used on a coarser level for macro-scale modeling. A comparative analysis of a more complex Batty-based thermal model and the $RC^*$-based resistive cell model was presented and determined that for some cases, an $RC^*$-based model can provide accurate results with a much lighter computation time requirement. Computation efficiency is a paramount concern when
considering large thermal networks, which is further complicated with the time evolution involved. The flexibility of the ThermalBlockRC3D element allows for unilateral modeling of active semiconductor regions, packages, and heatsinks, all using the same macromodel. The choice of thermal block model type, number, and size comes as an engineering trade off between accuracy, computation time, and simplicity of simulation.

An experimental procedure was designed to validate the electrothermal simulations of the 3DIC. The experimental measurements included steady-state thermal imaging of the 3DIC surface, as well as transient thermal data collection through the power-on transient. In order to collect precise measurements, an infrared microscope was employed which did not require coating the object to be thermally measured with black thermal ink to ensure uniform emissivity. Furthermore, the method used to directly measure the phonon emission from the 3DIC die required sample attachment to a heated chuck for calibration. This required the design of a new heatsink that could connect through to the chip package though the PCB. The heatsink designed for this work consists of a large copper block with the test board mounted above. A copper pillar rises from the block through a hole cut beneath the socket in the PCB which does not interfere with any of the layers of routing. This heatsink design was validated by board-level Mechanica simulations to ensure that the majority of the heat flux was localized within the copper pillar, and not through the package pins in to the test board. The large scale thermal simulations confirmed successful board design for accurate measurements.

A method of time compaction in which thermal capacitances are reduced, and the resulting time error corrected for, lead to rapid transient simulation. It was shown mathematically that reduction of the $RC$ time constant of the differential equations which describe capacitive charging and discharging has the same effect as dividing the time axis by the same factor. It was then shown that time compaction by reducing thermal capacitance can be successfully applied to simulated electrothermal circuits. In simulated circuits, if thermal resistance was changed temperature results would be affected, therefore thermal capacitance was scaled down by one tenth which affects the time domain only. The time axis of the simulated data obtained
using these electrothermal netlists was then scaled by the same factor of ten and it was shown that these results matched simulations with unaltered thermal capacitances. Time compaction has been an important development for rapid determination of final steady state temperatures with transient thermal flux perturbation. Previously reported work used time averaged power dissipation models for heat injection allowing simpler, but less physically accurate, steady-state determination; whereas this work uses physically-accurate heat flux which is smoothed in time by the thermal capacitance of a material. Application of the time compaction and scaling technique used in this dissertation showed a computation time reduction of a factor of nine.

An analysis flow for full electrothermal simulation has was developed at the design flow level orchestrating the work flow between the extractor and previously developed fREEDA models. The fREEDA netlist extraction of a physical resistive thermal netlist which is based on the thermal material parameters of the 3DIC layers is described in detail. The correct connectivity of the electrothermal circuit and the extracted resistive thermal netlist is an important step in the work flow where the two netlists were joined with the help of Matlab helper scripts. This dissertation described the necessary steps to arrive at electrothermal simulated results.

The design flow includes three levels of simulation which were successfully applied, which are a full electrical only simulation, a full electrical and thermal extracted resistive simulation, and a hotspot dynamic electrothermal simulation. The simulations progress in the level of thermal detail resultant, and each builds from the previous step.

A study of thermal simulation results using varying extracted thermal netlist resolutions was presented. A thermal resistive mesh was extracted at 5 and 10 µm grid spacings and compared. Temperature intensity and geometry agreed between results with some heat spreading effects seen using the 10 µm resolution mesh, while the 5 µm resolution mesh is closer to the scale of the transistors and therefore more spatially accurate.

Accurate steady state, as well as transient simulations, were validated by thermal measurements while the circuit was operating. The experimental data enabled, for the first time, volumetric and temporal electrothermal simulations which could be matched to measured data.
A maximum rise in the temperature in the 3DIC hotspot was 8.5°C, while the maximum rise in temperature simulated in the high resolution simulation of the hotspot was 8.3°C. This is a favorable degree of agreement with 2.4% error. The fact that this method has been applied to an asynchronous non-standard cell design further demonstrates its efficacy. The method also applies, more accessibly, to clocked circuits using standard cells.

### 7.2 Future Research

In the research presented here the resolution of the extracted thermal netlist was required to be provided by the user. In future implementations it could be useful for the resolutions of areas of interest to be increased automatically.

One of the most time consuming and tedious tasks in this simulation effort were the steps of connecting the thermal and electrothermal netlists, even with helper scripts which generated subcircuit connection nodes. Numbered grid maps had to be referred to while choosing sites for heat injection from the electrothermal elements. This method was error-prone due to the fact that five separate places in the netlists had to be manually edited for each thermal connection. Scripting to automate connections between electrical and thermal networks, and to process results accordingly, would be very helpful and ease larger simulations as well as application to other projects.

One simplification that was made to simulations was to inject the heat flux from each macromodel cell as a point source into the extracted thermal network. In fact, this is not physically accurate, as the heat producing elements in the real-world circuit do have dimension and cover some area. In the circuit modeled here there are 96 and 77 transistors in the frequency multiplier and frequency divider cells, respectively, and each transistor contains a heat producing region. The behavioral macromodels reduced the complexity to 44 state variables per divider cell and 20 state variables per multiplier cell for simulation feasibility on the time scales required to observe thermal effects. Thus it would not be wise to include each individually, but rather to divide and spread the heat flux throughout the cells. Internally, eight heat points are available.
within the DFF subcircuit, which could be easily used. Due to the complexities of placing heat spots, or connecting the extracted thermal subcircuit, an automated script would make this step more accessible.

The time required for the power-on transient to reach steady state is many orders of magnitude higher than that of electrical events, up to milliseconds compared to a picosecond or less for the clock cycle, some moving window power averages have been considered [5]. A simulation package which can partition the circuit into two parts, electrical and thermal, and update the slower-changing thermal circuit less often could result in a significant reduction in the CPU time required. Additionally, a simulator which can automatically vary its time steps between fast and slow events where a higher degree of error is permissible would be advantageous.
REFERENCES


APPENDICES
Appendix A

Asynchronous 3DIC Design

A.1 Asynchronous Automated Design

The asynchronous circuits used in the 3DIC (which were developed at the Asynchronous VLSI and Architecture group at Cornell University [61]) were individually synthesized by defining their behavior. Handshake expressions (HSE) are defined by waiting for a signal “a”, [a], assigning “a” as high or low by a+ or a-. If-then statements are evaluated as S1 then S2 by, S1:S2, and statements such as S1 and S2 in parallel are described by, S1,S2. The handshake expression is defined as,

- Wait for signal "a": [a]
- Put signal "a" to digital high: a+
- Put signal "a" to digital low: a-
- Execute statement S1 and THEN statement S2 (semicolon): S1;S2
- Execute statement S1 in parallel with statement S2 (comma): S1,S2

As an example, an HSE can be described by [a]; b+ [ a] ; bas:, which is interpreted as the following. Wait for “a” to be true, then set “b” then wait for “a” to become false, then set “b” false. After developing an HSE, the HSE is compiled into transistors. In order to archive this task the schematic must be described as a set of production rules. Pull-up and pull-down CMOS transistors are specified as a text based schematic, for example, an inverter with inputs
“a” and “b”:

a -> b-
¬a -> b+

The first line of the inverter can be read as: if “a” is true then “b” becomes false (0 V).
The second line of the inverter can be read as: if “not a” is true, then “b” becomes true (1.8 V).
Logical gates such as the NAND are specified by:

a&b -> c-
¬a | ¬b -> c+
or NOR
a | b -> c-
¬a & ¬b -> c+

Dynamic gates be interpreted as a domino precharge inverter, by the following.

a & clk -> b-
¬clk -> b+

To eliminate charge sharing, which becomes an issue in dynamic domino logic and reduces Vt, compilers automatically add required bleeder or keeper transistors. The compilation of the frequency divider circuit used in the 3DIC follows.

Inputs:
L.d[0] (Input coming from the previous stage)
R.e (Acknowledgment of the next stage)
Reset (A global reset signal that sets every node to its initial state)
_Reset (A perfect inverse of the Reset signal)

Outputs
.e (acknowledge that I will send to the previous Stage)
.d[0] (The output that goes to the next stage)
Intermediate signals: x.d[0], x.d[1], y.d[0], y.d[1], _o.d[0], _o.d[1], r1, peek (unused).
HSE of a frequency divider:
[[Re&Ld];Rd+;Le-;[¬Re&¬Ld];Rd-;Le+;[Ld];Le-;[¬Ld];Le+]

This block of code is interpreted as the following. Wait for Re and Ld (the next stage to have acknowledged and the previous to have set some data), then set Rd+ (the data sent to the next stage).
stage) high. Enable is then set to the previous stage low (Le−), then wait for the next stage to acknowledge ∼Re, and wait for the previous stage to put the data rail down, ∼Ld. After these events, set the data to the next stage “low” by, Rd−, and then send the acknowledgment to the previous stage as high, Le+. Afterwards, wait for the next data signal from the previous stage by, Ld, and send an acknowledgment to the previous stage by setting it low, Le−. Wait for the previous stage to acknowledge by setting the data low ∼Ld, and finally the last acknowledgment to the previous stage is sent by setting Le+.

Note that for each two tokens of the previous stage L.d only one token is produced at the output R.d. This handshake expression enables simplistic to compilation of production rules using well-published methods [61]. The result the compiled production rules is1:

\[
\begin{align*}
\text{Reset} & \rightarrow x.d[1]- \\
\neg \text{Reset} & \rightarrow x.d[0]+ \\
\text{Reset} & \rightarrow y.d[1]- \\
\neg \text{Reset} & \rightarrow y.d[0]+ \\
\text{Reset} & \rightarrow R.d[0]- \\
\neg \text{Reset} & \rightarrow o.d[1]+ \\
\neg \text{Reset} & \rightarrow o.d[0]+ \\
\text{R.e} & \land L.d[0] & x.d[0] & y.d[0] \rightarrow o.d[0]- \\
\neg o.d[0] & \rightarrow R.d[0]+ \\
R.d[0] & \rightarrow L.e- \\
R.d[0] & \land x.d[0] \rightarrow y.d[0]- \\
\neg y.d[0] & \land (\neg r1 & x.d[1] | \text{Reset}) \rightarrow y.d[1]+ \land x-coupled \\
\text{R.e} & \land L.d[0] & \neg y.d[0] \rightarrow o.d[0]+ \\
o.d[0] & \land y.d[1] \rightarrow R.d[0]- \\
\text{R.d[0]} & \land \neg r1 \rightarrow L.e+ \\
L.e & \land y.d[1] & \land o.d[0] \rightarrow x.d[0]- \\
\neg x.d[0] & \land (L.e & y.d[0] & \land o.d[1] | \text{Reset}) \rightarrow x.d[1]+ \land x-coupled \\
\text{L.d[0]} & \land x.d[1] & \land y.d[1] \rightarrow o.d[1]- \land y.d[1]? \\
\neg o.d[1] & \rightarrow r1+ \\
r1 & \rightarrow L.er1 \\
& \land x.d[1] \rightarrow y.d[1]-
\end{align*}
\]

1This code for asynchronous design is adapted from code written by Carlos Tadeo Ortero at the Asynchronous VLSI and Architecture group at Cornell University [61].
\(\neg y.d[1] \& \neg (R.d[0] \& x.d[0]) \rightarrow y.d[0]+ \) // x-coupled

// reset-phase affects x
\(\neg L.d[0] \& \neg y.d[1] \rightarrow _o.d[1]+\)
\(_o.d[1] \& y.d[0] \rightarrow r1-\)

L.e \& y.d[0] \& \_o.d[1] \rightarrow x.d[1]-

\(\neg x.d[1] \& \neg (L.e \& y.d[1] \& \_o.d[0]) \rightarrow x.d[0]+ \) // x-coupled

With these production rules, schematics can easily be drawn with code. Automatic tools exist which generate spice-style netlist. Similar to the previous process, the frequency multiplier is described by the HSE, \([\text{Re&Ld}; \text{Le}-; \text{Rd}+; [ \text{Re}]; \text{Le}+; \text{Rd}-; [\text{Re}]; \text{Rd}+; [\text{Re}]; \text{Rd}-].\)

Intermediate signals are \_le, \_o[0], \_o[1], x.d[0], x.d[1], y.d[0], y.d[1], and \_le. The production rules are \(^2\):

\begin{align*}
\text{Reset} & \rightarrow x.d[0]- \\
\neg \text{Reset} & \rightarrow x.d[1]+ \\
\text{Reset} & \rightarrow y.d[0]- \\
\neg \text{Reset} & \rightarrow y.d[1]+ \\
\neg \text{Reset} & \rightarrow \_o[0]+ \\
\neg \text{Reset} & \rightarrow \_o[1]+ \\
\text{Reset} & \rightarrow \_o[0]- \\
\text{Reset} & \rightarrow \_o[1]- \\
\_o[0] & \rightarrow \_le- \\
\_le & \rightarrow L.e- \\
\_o[0] \& \_o[1] & \rightarrow R.d[0]+ \\
\_o[0] \& \_o[1] & \rightarrow R.d[0]- \\
\text{set-phase affects y} \\
R.e \& x.d[0] \& y.d[0] \& \_\text{Reset} & \rightarrow \_o[1]- \\
\_o[1] & \rightarrow \_o[1]+ \\
x.d[0] \& \_o[1] & \rightarrow y.d[0]- \\

// reset-phase affects x
\text{R.e} \& \_x.d[1] \& \_y.d[0] \rightarrow \_o[1]+ \) // eww, 3p

\_o[1] \& y.d[1] \rightarrow \_o[1]- \\
\_o[1] \& \_o[1] \& \_x.d[0]- \\

// set-phase affects y

\text{R.e} \& x.d[0] \& x.d[1] \& y.d[1] \rightarrow \_o[0]- \\
\_\_o[0] \rightarrow \_o[0]+ \\
x.d[1] \& \_o[0] \rightarrow y.d[1]-
\end{align*}

\(^2\)This code for asynchronous design is adapted from code written by Carlos Tadeo Ortero at the Asynchronous VLSI and Architecture group at Cornell University [61].
\neg y.d[1] \& \neg(x.d[0] \& o[1] \mid \text{Reset}) \rightarrow y.d[0]^+ // x\text{-coupled}

// reset-phase affects x
\neg R.e \& \neg L.d[0] \& \neg x.d[0] \& \neg y.d[1] \rightarrow _o[0]^+ // eww, 4p

_o[0] \& y.d[0] \rightarrow o[0]^{-}

_o[0] \& y.d[0] \rightarrow x.d[1]^{-}

\neg o[0] \& \neg x.d[1] \& \neg(_o[1] \& y.d[1] \mid \text{Reset}) \rightarrow x.d[0]^+ // x\text{-coupled}
Appendix B

Matlab Helper Scripts

B.1 Extracted Thermal Netlist Grid Drawer

After generating an electrical netlist, and extracting a thermal netlist, this is the first script that should be used to visualize the thermal grid. Without having a numerical representation of where the resistors or nodes are, thermal connections can not be connected in a geometrically sensible way. The way this netlist is currently configured, it draws the resistive grid and numbers the resistors. The user then has to determine where in the grid thermal connections are to be made, then the resistor numbers must be cross-checked with the extracted thermal netlist to find a node number associated with them. A potentially useful script would be one which also lists all of the node numbers automatically.

User input is required to find the first resistor number of each desired layer for connections. In this case all of the correct starting resistor numbers are available at the top of the script for the user to uncomment as needed for each layer and for 10µm and 5µm resolutions. The following is a reprinting of the GridDrawer5u.m file.

```
clear;
clf;

hold on;
```
x=0;
y=0;
row=0;
col=0;
%active C 5u
%node=22801;
%active C 10u
%node=5401;
%active B 10u
%node= 3916
%active A 10u
%node=1756
%overglass 10u
%node=5941;
%handle1
%node=1;

%active C 5u
%node=22801;
%active B 5u
%node=16531;
%active A 5u
%node=7411;
%overglass1 5u
node=25081;

%xmax=10;
%ymax=5;
xmax=20;
ymax=10;

%textbox width and h
w=1/20;
h=1/10;
%lable locations
ylablex=0.09;
ylabley=0.08/2;
xlablex=0.123;
xlabley=0.055/2;

while(col<ymax)
%draw the last x row lower so they are visible on graph
if(col==ymax-1)
y=y-.2;
end
while(row<xmax)
    %not lableing zline for clarity
    node=node+1;
    %y line
    if(col~=ymax-1)
        if(col==ymax-2)
            %this draws the y line of the top row slightly shorter
            %for matlab graphing
            line([x x], [y y+4.8]);
        else
            line([x x], [y y+5]);
        end
        annotation('textbox', [ylablex, ylabley, w, h], 'String', node, 'FontSize', 8, 'LineStyle', 'none', 'Color', [1 0 1], 'HorizontalAlignment', 'center');
        ylablex=ylablex+0.03835*2;
        ylablex=ylablex+0.0387;
        node=node+1;
    end
    %x line
    if(row~=xmax-1)
        line([x x+5], [y y]);
        %blue x annotation
        annotation('textbox', [xlablex, xlabley, w, h], 'String', node, 'FontSize', 9, 'LineStyle', 'none', 'Color', [0 1 1], 'HorizontalAlignment', 'center');
        xlablex=xlablex+0.03835*2;
        xlablex=xlablex+0.042;
    end
    row=row+1;
end

ylablex=0.09;
% ylabley=ylabley+.08;
% ylabley=ylabley+.09*2;
ylabley=ylabley+.09;
   xlablex=0.123;

%  xlabley=xlabley+.1;
xlabley=xlabley+.09;
y=y+5;
col=col+1;
row=0;
x=0;
end
Figure B.1: The output of the grid drawer script for the numbered grid of the Active_C layer.
B.2 Netlist and Node Number Generator

The following Matlab script is used to generate three parts of the netlist. The connections in the main netlist to the subcircuit file, the output file names for each transient result for each desired layer in the 3DIC (Overglass, Active_A, Active_B, and Active_C), and the connections in the subcircuit file to the main file. Two versions of this netlist were created; one for a 10µm resolution extracted thermal netlist, and one for a 5µm resolution extracted thermal netlist. This script could be adapted in the future to work automatically with any size netlist. In its current state, much user interaction is required by finding the first node number at the start of each desired layer. The following is a reprinting of the SubcircuitConnections5u.m file. SubcircuitConnections10u.m is also available.

```matlab
% List nodes 5u

n=4001:4135
fprintf('Connections\n');

% Overglass connections
filenumber=1;
n=1;
while(n<201)
    fprintf('+"og%0.0f" \n',n);
    filenumber=filenumber+1;
    n=n+1;
end

% Active C connections
filenumber=1;
n=1;
while(n<201)
    fprintf('+"act_c%0.0f" \n',n);
    filenumber=filenumber+1;
    n=n+1;
end
```

143
%active B connections
filenumber=1;
n=1;
while(n<201)

    fprintf('+"act_b%0.0f" \n’,n);
    filenumber=filenumber+1;
    n=n+1;
end

%Active A connections
filenumber=1;
n=1;
while(n<201)

    fprintf('+"act_a%0.0f" \n’,n);
    filenumber=filenumber+1;
    n=n+1;
end

fprintf('Output files\n');
%overglass Output files
n=1;
filenumber=1;
%change to 50
while(n<201)

    fprintf('.out plot term 4001 vt in "OverGlass1.out"\n');
    fprintf('.out plot term "og%0.0f" vt in "OverGlass%0.0f.out"\n’, ... 
n, filenumber);
    filenumber=filenumber+1;
    n=n+1;
end

%Active C output files
n=1;
filenumber=1;
while(n<201)

    fprintf('.out plot term "act_c%0.0f" vt in "c%0.0f.out"\n’, n, ... 
    filenumber);
    filenumber=filenumber+1;
    n=n+1;
end
%Active B output files
n=1;
filenumber=1;
%change to 50
while(n<201)
    fprintf('.out plot term "act_b%0.0f" vt in "b%0.0f.out"
', n, ...
    filenumber);
    filenumber=filenumber+1;
    n=n+1;
end

%Active A output files
n=1;
filenumber=1;
%change to 50
while(n<201)
    fprintf('.out plot term "act_a%0.0f" vt in "a%0.0f.out"
', n, ...
    filenumber);
    filenumber=filenumber+1;
    n=n+1;
end

fprintf('Subcircuit Connections
');
fprintf('*Overglass subckt
');
n=8801;
while(n<9002)
    fprintf('+%0.0f 
',n);
    filenumber=filenumber+1;
    n=n+1;
end

fprintf('*Active C subckt
');
n=8001;
while(n<8201)
    fprintf('+%0.0f 
',n);
    filenumber=filenumber+1;
    n=n+1;
end

fprintf('*Active B Subckt
');
n=5801;
while(n<6001)
    fprintf('%0.0f \n',n);
    filenumber=filenumber+1;
    n=n+1;
end

fprintf('*Active A subckt\n');
n=2601;
while(n<2801)
    fprintf('%0.0f \n',n);
    filenumber=filenumber+1;
    n=n+1;
end
B.3 Time Averaging and 3D Graphing Thermal Plots

The Matlab script listed below is that which generates the three-dimensional temperature graphs of the layers of interest in the 3DIC. Each point on the plots represents a large data file which is imported and time-averaged over some duration. This can be adjusted to examine the transient behavior as time advances. The following is a reprinting of the 3DICGraphTimeAve.m file.

```matlab
clear;

layernum=1;
while(layernum<5)
    filenumber=1;
    x=1;
    y=1;
    switch layernum
        case 1
            layer = 'c';
        case 2
            C=ave_temps;
            layer = 'b';
        case 3
            B=ave_temps;
            layer = 'a';
        case 4
            A=ave_temps;
            layer = 'overglass';
    end

    while(filenumber<201)
        while(x<11)
            while(y<21)
                filename = [layer num2str(filenumber) '.out'];
                M = dlmread(filename,'	');
                %M = M(12001:28001,2);
                %in general use:
```
%M = M(10000:12001,2);
%M = M(1:2,2);
%for really quick ones
M = M(1538:1600,2);
ave_temps(x,y) = mean(M);
filenumber=filenumber+1;
y=y+1;
end
y=1;
x=x+1;
end

% pcolor(ave_temps);
layernum=layernum+1;
if (layernum == 5)
    break
end

% figure;
end
og=ave_temps;

%set(gca,'XTick',[1:100])
ticks = [’ 10’; ’ ’; ’ 20’; ’ ’; ’ 30’; ’ ’; ’ 40’; ...
’ ’; ’ 50’; ’ ’; ’ 60’; ’ ’; ’ 70’; ’ ’; ’ 80’; ...
’ ’; ’ 90’; ’ ’; ’100’; ’ ’];
yticks = [’ 10’; ’20’; ’30’; ’40’; ’50’];

%opengl neverselect
%camlight right
subplot(2,2,1);
%imagesc(A);
surf(A);
shading(’interp’);
zlabel(’Temperature (K)’);title(’Active A’);
ylabel(’Width (\mu)’);xlabel(’Length (\mu)’);
set(gca,’XTick’,[1:20]); set(gca,’YTick’,[1:100]);

set(gca,’XTickLabel’,ticks); set(gca,’YTickLabel’,ticks);
%caxis([3 4])
subplot(2,2,2);
surf(B);
%imagesc(B);
shading('interp');
zlabel('Temperature C');title('Active B');
ylabel('Width (\mu)');xlabel('Length (\mu)')
set(gca,'XTick',[1:20]); set(gca,'YTick',[1:100]);

set(gca,'XTickLabel',ticks); set(gca,'YTickLabel',ticks);
%caxis([3 4])
subplot(2,2,3);
surf(C);
%imagesc(C);
shading('interp');
zlabel('Temperature C');title('Active C');
ylabel('Width (\mu)');xlabel('Length (\mu)')
set(gca,'XTick',[1:20]); set(gca,'YTick',[1:100]);

set(gca,'XTickLabel',ticks); set(gca,'YTickLabel',ticks);
%caxis([3 4])
subplot(2,2,4);
surf(og);
%imagesc(og);
shading('interp');
zlabel('Temperature C');title('Over Glass');
ylabel('Width (\mu)');xlabel('Length (\mu)')
set(gca,'XTick',[1:20]); set(gca,'YTick',[1:100]);
set(gca,'XTickLabel',ticks); set(gca,'YTickLabel',yticks);
%caxis([3 4])

%shading('interp');
Appendix C

fREEDA Netlists

C.1 Netlists for Thermal RC Comparison to the Batty Model

The following netlists were used to model the thermal effects resultant from connecting an electrothermal CMOS inverter to a simulated block of silicon substrate.

C.1.1 Batty Model Thermal N-Port Netlist

The following lists the InvBatty.net file of Batty model netlist for comparison to an RC-based simulation.

** Electrothermal Inverter attached to Silicon **
** thermalheatsinkspot Batty model transient analysis **
******************************************************************************
.tran2 tstop=4e-6 tstep=10e-9 out_steps=1
.options initTmp=300 deltat=10e-9 nsteps=400 temp=300.
.ref "tref"
.ref 0

*Connecting inverter with thermal terminals to thermal heatsink (block of material)*
*Electrothermal connections are:*
* vdd in o g th-j th-r*
cmosinvt:Q1 1 2 3 0 3000 "tref" thermal=1 td=0.1e-6
**wn=16e-6 ln=13e-6 wp=50e-6 lp=13e-6
c: c1 3 0 c=1e-15

vpulse: Vin 2 0 v1=0 v2=5 per=1e-6 pw=.5e-6 tr=0.05e-6 tf=0.05e-6

vsource: vdd 1 0 vdc=5

*.model silicon thermalheatsinkspot

* thermal
thermalheatsinkspot: SiliconBlock 3000 6000
+ Ntimesteps=nsteps
+ dt=deltat
+ Tambient=initTmp narray=1 ndevices=2 w=100e-6 l=100e-6 d=1000e-6
* ks: Thermal conductivity in (W/m.K).
++ ks=1.1
+ ks=149
*b: Exponent in power law temperature dependence of thermal conductivity
+b=0.0342
*c: Specific heat (J/kg.K).
+c=1150
+c=700
+time_d=1
*Dimensions can be specified:
++ x1=0.25e-6
++ xr=0.25e-6
++ yu=1.0e-6
++ yd=1.0e-6
vsource: vport 6000 "tref" vdc=initTmp

.out plot term 2 vt in "pulse.out"
.out plot term 3 vt in "inv_output.out"
.out plot element "thermalheatsinkspot: SiliconBlock" 0 ut in
+"temperature2.out"
.out plot element "thermalheatsinkspot: SiliconBlock" 0 it in "heatflux2.out"

.end
C.1.2  **RC-Based Model Thermal Netlist**

The following lists the `RCInv.net` file of the RC model netlist for comparison to the Batty-based simulation.

```plaintext
** Electrothermal Inverter attached to Silicon  **
** thermalblock RC model transient analysis **
************************************************

.tran2 tstop=1e-3 tstep=10e-9 out_steps=1
.options initTmp=300 deltat=1us nsteps=100000 temp=300.
.ref "tref"
 ref 0

*Thermal inverter with thermal terminals to bulk
ccmosinv3:Q1 1 2 3 0 3000 "tref" thermal=1 td=0.1e-6
++\textit{wn}=16e-6 \textit{ln}=13e-6 \textit{wp}=50e-6 \textit{lp}=13e-6

c\text{\texttt{c1}} 3 0 c=1e-15

vpulse:Vin 2 0 v1=0 v2=5 per=1e-6 pw=.5e-6 tr=0.05e-6 tf=0.05e-6

vsource:Vdd 1 0 vdc=5

*Cube of 6 Rs centered around 3000, heat injection of inverter
*The bottom is connected to heatsink, thermal ground
.options rcube=150
r\text{\texttt{rta1}} 3000 "tref" r=1200
r\text{\texttt{rta2}} 3000 "aleft" r=rcube
r\text{\texttt{rta3}} 3000 "aright" r=rcube
r\text{\texttt{rta4}} 3000 "afront" r=rcube
r\text{\texttt{rta5}} 3000 "aback" r=rcube
r\text{\texttt{rta6}} 3000 "atop" r=rcube
*c\text{\texttt{cta}} 3000 "tref" c=7e-10
c\text{\texttt{cta}} 3000 "tref" c=200e-9

*Connect sides of cube to thermal gnd for adiabatic conditions
.options rsides=2500
r\text{\texttt{rleft}} "aleft" "tref" r=rsides
r\text{\texttt{raright}} "aright" "tref" r=rsides
r\text{\texttt{raback}} "aback" "tref" r=rsides
r\text{\texttt{rafront}} "afront" "tref" r=rsides
```

152
C.2 Netlist for Complete Electrical-Only Simulation of the Frequency Chain

The following netlist, FullChainElec.net, simulates the complete chain of 15 frequency multipliers followed by 10 frequency dividers. Macromodels based on standard and custom cells were employed as subcircuits, which also follow this section.

C.2.1 Main Netlist for Electrical-Only Simulation of the Frequency Chain

**** Netlist for Full Chain of Multipliers
*and Dividers Electrical Simulation
*By T. Robert Harris
*97567.576u 22.151s 27:07:12.38 99.9% 0+0k 0+0io 0pf+0w
*25 hours of simulation time
***********************************
*Transient Sim
.tran2 tstop=34e-6 tstep=10e-11
.options initTmp=300
*Include Cell Subcircuits
.inc nanddff_subckt.net
.inc xor_subckt.net

vpulse: Vin 2 0 v1=0 v2=1.8 per=32e-6 pw=16e-6 tr=2e-11 tf=2e-11
**** Vdd connection
vsource: Vdd 1 0 vdc=1.8

*** Chain of 15 Frequency Multipliers
xxor1 1 0 2 3 4 XOR_1
vpulse:d1 3 0 v1=0 v2=1.8 per=32768e-9 pw=16384e-9 tr=2e-11 tf=2e-11
+td=8192e-9

xxor2 1 0 4 5 6 XOR_1
vpulse:d2 5 0 v1=0 v2=1.8 per=16384e-9 pw=8192e-9 tr=2e-11 tf=2e-11 td=4096e-9

xxor3 1 0 6 7 8 XOR_1
vpulse:d3 7 0 v1=0 v2=1.8 per=4096e-9 tr=2e-11 tf=2e-11 td=2048e-9

xxor4 1 0 8 9 10 XOR_1
vpulse:d4 9 0 v1=0 v2=1.8 per=2048e-9 pw=1024e-9 tr=2e-11 tf=2e-11 td=1024e-9

xxor5 1 0 10 11 12 XOR_1
vpulse:d5 11 0 v1=0 v2=1.8 per=1024e-9 pw=512e-9 tr=2e-11 tf=2e-11 td=512e-9

xxor6 1 0 12 13 14 XOR_1
vpulse:d6 13 0 v1=0 v2=1.8 per=512e-9 tr=2e-11 tf=2e-11 td=256e-9

xxor7 1 0 14 15 16 XOR_1
vpulse:d7 15 0 v1=0 v2=1.8 per=256e-9 tr=2e-11 tf=2e-11 td=128e-9

xxor8 1 0 16 17 18 XOR_1
vpulse:d8 17 0 v1=0 v2=1.8 per=128e-9 pw=64e-9 tr=2e-11 tf=2e-11 td=64e-9

xxor9 1 0 18 19 20 XOR_1
vpulse:d9 19 0 v1=0 v2=1.8 per=64e-9 pw=32e-9 tr=2e-11 tf=2e-11 td=32e-9

xxor10 1 0 20 21 22 XOR_1
vpulse:dx10 21 0 v1=0 v2=1.8 per=32e-9 pw=16e-9 tr=2e-11 tf=2e-11 td=16e-9

xxor11 1 0 22 23 24 XOR_1
vpulse:dx11 23 0 v1=0 v2=1.8 per=16e-9 pw=8e-9 tr=2e-11 tf=2e-11 td=8e-9

xxor12 1 0 24 25 26 XOR_1
vpulse:dx12 25 0 v1=0 v2=1.8 per=8e-9 pw=4e-9 tr=2e-11 tf=2e-11 td=4e-9

xxor13 1 0 26 27 28 XOR_1
vpulse:dx13 27 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11 td=2e-9

xxor14 1 0 28 29 30 XOR_1
vpulse:dx14 29 0 v1=0 v2=1.8 per=2e-9 pw=1e-9 tr=2e-11 tf=2e-11 td=1e-9

xxor15 1 0 30 31 32 XOR_1
vpulse:dx15 31 0 v1=0 v2=1.8 per=1e-9 pw=.05e-9 tr=2e-11 tf=2e-11 td=.05e-9

*Chain of 10 Frequency Dividers

xdff1 1 72 32 74 72 0 DFF_1
xdff2 1 722 74 724 722 0 DFF_1
xdff3 1 732 724 734 732 0 DFF_1
xdff4 1 742 732 744 742 0 DFF_1
xdff5 1 752 742 754 752 0 DFF_1
xdff6 1 762 752 764 762 0 DFF_1
*** Resistances

r:R1 2 0 r=1e5
r:R2 3 0 r=1e5
r:R3 4 0 r=1e5
r:R5 5 0 r=1e5
r:R6 6 0 r=1e5
r:R7 7 0 r=1e5
r:R8 8 0 r=1e5
r:R9 9 0 r=1e5
r:R10 10 0 r=1e5
r:R11 11 0 r=1e5
r:R12 12 0 r=1e5
r:R13 13 0 r=1e5
r:R14 14 0 r=1e5
r:R15 15 0 r=1e5
r:R16 16 0 r=1e5
r:R17 17 0 r=1e5
r:R18 18 0 r=1e5
r:R19 19 0 r=1e5
r:R20 20 0 r=1e5
r:R21 21 0 r=1e5
r:R22 22 0 r=1e5
r:R23 23 0 r=1e5
r:R24 24 0 r=1e5
r:R25 25 0 r=1e5
r:R26 26 0 r=1e5
r:R27 27 0 r=1e5
r:R28 28 0 r=1e5
r:R29 29 0 r=1e5
r:R30 30 0 r=1e5
r:R31 31 0 r=1e5
r:R32 32 0 r=1e5

* Plot electrical output for each stage
* Frequency multipliers output

.options gnuplot
.out plot term 2 vt in "vin.out"
.out plot term 4 vt in "4.out"
.out plot term 6 vt in "6.out"
The following netlist, nandff_subckt.net is a necessary component for the full electrical simulation of the complete frequency chain in the 3DIC.

```
*Netlist of DFF cell
.subckt DFF_1 "vdd" "d" "clk" "q" "qb" "gnd"

_cmosinv:clkinv "vdd" "clk" "clkb" "gnd" ln=0.18e-6 wn=1.8e-6
lp=0.18e-6 wp=2.4e-6
```

C.2.2 DFF Subcircuit Netlist for Complete Electrical-Only Simulation

The following netlist, nandff_subckt.net is a necessary component for the full electrical simulation of the complete frequency chain in the 3DIC.
C.2.3 XOR Subcircuit Netlist for Complete Electrical-Only Simulation

The following netlist, xor_subckt.net is a necessary component for the full electrical simulation of the complete frequency chain in the 3DIC.

*Netlist of XOR cell

.circuit XOR_Thermal "vdd" "gnd" "In1" "In2" "out"

.circuit cmos2nand:and1 "vdd" "d" "clk" 1 "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and2 "vdd" "clk" 1 2 "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and3 "vdd" 1 4 3 "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and4 "vdd" 2 3 4 "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and5 "vdd" 3 "clkb" 5 "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and6 "vdd" 4 "clkb" 6 "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and7 "vdd" 5 "qb" "q" "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

.circuit cmos2nand:and8 "vdd" 6 "q" "qb" "gnd" ln=0.18e-6 wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

* Intermediate nodal capacitance

.circuit c:c1 1 "gnd" c=1e-14

.circuit c:c2 2 "gnd" c=1e-14

.circuit c:c3 3 "gnd" c=1e-14

.circuit c:c4 4 "gnd" c=1e-14

.circuit c:c5 5 "gnd" c=1e-14

.circuit c:c6 6 "gnd" c=1e-14

.circuit c:c7 "q" "gnd" c=1e-14

.circuit c:c8 "qb" "gnd" c=1e-14

.circuit c:c9 "clkb" "gnd" c=1e-14

.ends
C.3 Netlist for Complete Electrothermal Simulation of the Frequency Chain

The following netlist, FullChainThermal.net simulates the complete chain of 15 frequency multipliers followed by 10 frequency dividers with an extracted resistive thermal network.

**** Netlist for Full Electrothermal Chain
* of Multipliers and Dividers Electrical Simulation
*By T. Robert Harris
*
***********************************
*Transient simulation
.tran2 tstop=1e-6 tstep=10e-11

.ref "tref"
*Set temperature to ambient room temperature
.options initTmp=300
*Include Electrothermal Cell Subcircuits
.inc nanddff_subckt.net
.inc xor_subckt.net

vpulse:Vin 2 0 v1=0 v2=1.8 per=32e-6 pw=16e-6 tr=2e-11 tf=2e-11
d=8192e-9
**** Vdd connection
vsource:Vdd 1 0 vdc=1.8

*Chain of 15 Electrothermal Frequency Multipliers
xxor1 1 0 2 3 4 1002 "tref" XDR_Thermal

vpulse:d1 3 0 v1=0 v2=1.8 per=32768e-9 pw=16384e-9 tr=2e-11 tf=2e-11
+td=8192e-9
xor2 1 0 4 5 6 2002 "tref" XOR_Thermal
vpulse:d2 5 0 v1=0 v2=1.8 per=16384e-9 pw=8192e-9 tr=2e-11 tf=2e-11 td=4096e-9
xor3 1 0 6 7 8 3002 "tref" XOR_Thermal
vpulse:d3 7 0 v1=0 v2=1.8 per=8192e-9 pw=4096e-9 tr=2e-11 tf=2e-11 td=2048e-9
xor4 1 0 8 9 10 4002 "tref" XOR_Thermal
vpulse:d4 9 0 v1=0 v2=1.8 per=4096e-9 pw=2048e-9 tr=2e-11 tf=2e-11 td=1024e-9
xor5 1 0 10 11 12 5002 "tref" XOR_Thermal
vpulse:d5 11 0 v1=0 v2=1.8 per=2048e-9 pw=1024e-9 tr=2e-11 tf=2e-11 td=512e-9
xor6 1 0 12 13 14 6002 "tref" XOR_Thermal
vpulse:d6 13 0 v1=0 v2=1.8 per=1024e-9 pw=512e-9 tr=2e-11 tf=2e-11 td=256e-9
xor7 1 0 14 15 16 7002 "tref" XOR_Thermal
vpulse:d7 15 0 v1=0 v2=1.8 per=512e-9 pw=256e-9 tr=2e-11 tf=2e-11 td=128e-9
vpulse:d8 17 0 v1=0 v2=1.8 per=256e-9 pw=128e-9 tr=2e-11 tf=2e-11 td=64e-9
xor8 1 0 16 17 18 8002 "tref" XOR_Thermal
vpulse:d9 19 0 v1=0 v2=1.8 per=128e-9 pw=64e-9 tr=2e-11 tf=2e-11 td=32e-9
xor9 1 0 18 19 20 9002 "tref" XOR_Thermal
xor10 1 0 20 21 22 10002 "tref" XOR_Thermal
vpulse:d10 21 0 v1=0 v2=1.8 per=64e-9 pw=32e-9 tr=2e-11 tf=2e-11 td=16e-9
vpulse:d11 23 0 v1=0 v2=1.8 per=32e-9 pw=16e-9 tr=2e-11 tf=2e-11 td=8e-9
xor11 1 0 22 23 24 11002 "tref" XOR_Thermal
vpulse:d12 25 0 v1=0 v2=1.8 per=16e-9 pw=8e-9 tr=2e-11 tf=2e-11 td=4e-9
xor12 1 0 24 25 26 12002 "tref" XOR_Thermal
vpulse:d13 27 0 v1=0 v2=1.8 per=8e-9 pw=4e-9 tr=2e-11 tf=2e-11 td=2e-9
xor13 1 0 26 27 28 13002 "tref" XOR_Thermal
vpulse:Vin4 29 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11 td=1e-9
xor14 1 0 28 29 30 14002 "tref" XOR_Thermal
vpulse:Vin5 31 0 v1=0 v2=1.8 per=2e-9 pw=1e-9 tr=2e-11 tf=2e-11 td=.05e-9
xor15 1 0 30 31 32 15002 "tref" XOR_Thermal

*Chain of 10 Electrothermal Frequency Dividers
xdff1 1 72 32 74 72 0 1000 "tref" DFF_Thermal1
xdff2 1 722 74 724 722 0 2000 "tref" DFF_Thermal1
xdff3 1 732 724 734 732 0 3000 "tref" DFF_Thermal1
xdff4 1 742 732 744 742 0 4000 "tref" DFF_Thermal1
xdff5 1 752 742 754 752 0 5000 "tref" DFF_Thermal1
xdff6 1 762 752 764 762 0 6000 "tref" DFF_Thermal1
xdff7 1 772 762 774 772 0 7000 "tref" DFF_Thermal1
xdff8 1 782 772 784 782 0 8000 "tref" DFF_Thermal1
xdff9 1 792 782 794 792 0 9000 "tref" DFF_Thermal1
xdff10 1 102 92 104 102 0 10000 "tref" DFF_Thermal1

*** Thermal Resistance and Capacitance for dffs
.options r_extracted=89225.4 c_extracted=1e-12
*** Thermal Resistance and Capacitance for XORs

r:Rtx1 1002 1001 r=r_extracted
c:Ctx1 1002 1001 c=c_extracted
r:Rtx2 2002 1001 r=r_extracted
c:Ctx2 2002 1001 c=c_extracted
r:Rtx3 3002 1001 r=r_extracted
c:Ctx3 3002 1001 c=c_extracted
r:Rtx4 4002 1001 r=r_extracted
c:Ctx4 4002 1001 c=c_extracted
r:Rtx5 5002 1001 r=r_extracted
c:Ctx5 5002 1001 c=c_extracted
r:Rtx6 6002 1001 r=r_extracted
c:Ctx6 6002 1001 c=c_extracted
r:Rtx7 7002 1001 r=r_extracted
c:Ctx7 7002 1001 c=c_extracted
r:Rtx8 8002 1001 r=r_extracted
c:Ctx8 8002 1001 c=c_extracted
r:Rtx9 9002 1001 r=r_extracted
c:Ctx9 9002 1001 c=c_extracted
r:Rtx10 10002 1001 r=r_extracted
c:Ctx10 10002 1001 c=c_extracted
c:Ctx11 11002 1001 c=c_extracted
r:Rtx12 12002 1001 r=r_extracted
c:Ctx12 12002 1001 c=c_extracted
r:Rtx13 13002 1001 r=r_extracted
c:Ctx13 13002 1001 c=c_extracted
r:Rtx14 14002 1001 r=r_extracted
c:Ctx14 14002 1001 c=c_extracted
r:Rtx15 15002 1001 r=r_extracted
c:Ctx15 15002 1001 c=c_extracted

vsource:vt1 1001 "tref" vdc=initTmp

*** Resistances
r:R1 2 0 r=1e5
r:R2 3 0 r=1e5
r:R3 4 0 r=1e5
r:R5 5 0 r=1e5
r:R6 6 0 r=1e5
r:R7 7 0 r=1e5
r:R8 8 0 r=1e5
r:R9 9 0 r=1e5
r:R10 10 0 r=1e5
r:R11 11 0 r=1e5
r:R12 12 0 r=1e5
r:R13 13 0 r=1e5
r:R14 14 0 r=1e5
r:R15 15 0 r=1e5
r:R16 16 0 r=1e5
r:R17 17 0 r=1e5
r:R18 18 0 r=1e5
r:R19 19 0 r=1e5
r:R20 20 0 r=1e5
r:R21 21 0 r=1e5
r:R22 22 0 r=1e5
r:R23 23 0 r=1e5
r:R24 24 0 r=1e5
r:R25 25 0 r=1e5
r:R26 26 0 r=1e5
r:R27 27 0 r=1e5
r:R28 28 0 r=1e5
r:R29 29 0 r=1e5
r:R30 30 0 r=1e5
r:R31 31 0 r=1e5
r:R32 32 0 r=1e5
* Plot electrical output for each stage
* Frequency multipliers output
  .options gnuplot
  .out plot term 2 vt in "vin.out"
  .out plot term 4 vt in "4.out"
  .out plot term 6 vt in "6.out"
  .out plot term 8 vt in "8.out"
  .out plot term 10 vt in "10.out"
  .out plot term 12 vt in "12.out"
  .out plot term 14 vt in "14.out"
  .out plot term 16 vt in "16.out"
  .out plot term 18 vt in "18.out"
  .out plot term 20 vt in "20.out"
  .out plot term 22 vt in "22.out"
  .out plot term 24 vt in "24.out"
  .out plot term 26 vt in "26.out"
  .out plot term 28 vt in "28.out"
  .out plot term 30 vt in "30.out"
  .out plot term 104 vt in "32.out"

* Frequency dividers output
  .out plot term 74 vt in "dff1.out"
  .out plot term 724 vt in "dff2.out"
  .out plot term 734 vt in "dff3.out"
  .out plot term 744 vt in "dff4.out"
  .out plot term 754 vt in "dff5.out"
  .out plot term 764 vt in "dff6.out"
  .out plot term 774 vt in "dff7.out"
  .out plot term 784 vt in "dff8.out"
  .out plot term 794 vt in "dff9.out"
  *.out plot term 104 vt in "dff10.out"

* Thermal nodes for multipliers
  .out plot term 1002 vt in "mult1_temp.out"
  .out plot term 2002 vt in "mult2_temp.out"
  .out plot term 3002 vt in "mult3_temp.out"
  .out plot term 4002 vt in "mult4_temp.out"
  .out plot term 5002 vt in "mult5_temp.out"
  .out plot term 6002 vt in "mult6_temp.out"
  .out plot term 7002 vt in "mult7_temp.out"
  .out plot term 8002 vt in "mult8_temp.out"
  .out plot term 9002 vt in "mult9_temp.out"
  .out plot term 10002 vt in "mult10_temp.out"
  .out plot term 11002 vt in "mult11_temp.out"
C.4 3DIC Hotspot Electrothermal Main Circuit Netlist

The electrothermal network for the 3DIC hotspot in the netlist 3DICHotspot.net follows.

*** fREEDA 3DIC_hotspot.net
*** Electrothermal 3DIC Hotspot
***By T. Robert Harris
***2 repeaters, 2 eaters. outputs 1GHz
****3 active layers

*Set ambient thermal reference to room temp 300K
.options initTmp=300
.ref "tref"

vsource:vt1 1001 "tref" vdc=initTmp

*Include electrothermal macromodel subcircuits
.inc xor_subckt.net
.inc nanddff_subckt.net
.inc hotspot_extracted_101x54_10u.net

*Simulation length options
*.tran2 tstop=1.5e-7 tstep=.25e-11
*.tran2 tstop=.8e-7 tstep=.25e-11
.tran2 tstop=.3e-7 tstep=.25e-11

* 202311.166u 2.631s 56:12:55.62 99.9% 0+0k 0+0io 0pf+0w
* 2.3 days (output from unix time command)

*input from pervious repeater step
vpulse:Vin3 28 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11
*500MHz

**** Vdd connection
vsource:Vdd 1 0 vdc=1.8

*ACTIVE LAYER C

***** Delay and XOR instantiation
vpulse:Vin4 29 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11 td=1e-9
xxor14 1 0 28 29 30 14000 "tref" XOR_Thermal

r:R28 28 0 r=1e5
r:R29 29 0 r=1e5
r:R30 30 0 r=1e5
r:R31 31 0 r=1e5
r:R32 32 0 r=1e5

** Source Resistance and Capacitance
*r:Rin0 212 0 r=1e3
*r:Rin1 213 0 r=1e3
*c:Cin2 212 0 c=1e-15
*c:Cin3 213 0 c=1e-15

*** Thermal Resistance and Capacitance
*Thermal Rs are for testing. Comment out
*when connected to extracted thermal netlist
*r:Rtx14 14000 1001 r=20e3
c:Ctx14 14000 1001 c=10e-13
*r:Rtx15 15000 1001 r=20e3
c:Ctx15 15000 1001 c=10e-13

164
**DFFs: Eaters have the following connections:**
"vdd" "d" "clk" "q" "qb" "gnd" "temp" "tref"
xdff1 1 2 32 4 2 0 1000 "tref" DFF_Termal1
xdff2 1 22 4 24 22 0 2000 "tref" DFF_Termal1

*** DFF Thermal Resistance and Capacitance
*r:Rtd1 1000 1001 r=20e3
c:Ctd1 1000 1001 c=10e-13
*r:Rtd2 2000 1001 r=20e3
c:Ctd2 2000 1001 c=10e-13

*ACTIVE LAYER B
vpulse:VinB "b28" 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11

***** Delay and XOR instantiation
vpulse:Vinb4 "b29" 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11 td=1e-9
xxorb14 1 0 "b28" "b29" "b30" "b14000" "tref" XOR_Termal

c:csmoothb "b30" 0 c=.1e-14
vpulse:Vin5b "b31" 0 v1=0 v2=1.8 per=2e-9 pw=1e-9 tr=2e-11 tf=2e-11
+td=0.5e-9
xxor15b 1 0 "b30" "b31" "b32" "b15000" "tref" XOR_Termal

r:R28b "b28" 0 r=1e5
r:R29b "b29" 0 r=1e5
r:R30b "b30" 0 r=1e5
r:R31b "b31" 0 r=1e5
r:R32b "b32" 0 r=1e5

** Source Resistance and Capacitance
*r:Rin0 212 0 r=1e3
*r:Rin1 213 0 r=1e3
*c:Cin2 212 0 c=1e-15
*c:Cin3 213 0 c=1e-15

*** Thermal Resistance and Capacitance
r:Rtx14b "b14000" 1001 r=20e3
c:Ctx14b "b14000" 1001 c=10e-13
r:Rtx15b "b15000" 1001 r=20e3
c:Ctx15b "b15000" 1001 c=10e-13
**DFFs**

"vdd" "d" "clk" "q" "qb" "gnd" "temp" "tref"

xdff1b 1 "b2" "b32" "b4" "b2" 0 "b1000" "tref" DFF_Thermal1

xdff2b 1 "b22" 4 "b24" "b22" 0 "b2000" "tref" DFF_Thermal1

*** DFF Thermal Resistance and Capacitance

r:Rtd1b "b1000" 1001 r=20e3
c:Cntd1b "b1000" 1001 c=10e-13

r:Rtd2b "b2000" 1001 r=20e3
c:Cntd2b "b2000" 1001 c=10e-13

*ACTIVE LAYER A

vpulse:VinA "a28" 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11

***** Delay and XOR instantiation

vpulse:Vina4 "a29" 0 v1=0 v2=1.8 per=4e-9 pw=2e-9 tr=2e-11 tf=2e-11 td=1e-9

xxora14 1 0 "a28" "a29" "a30" "a14000" "tref" XOR_Thermal

c:csmootha "a30" 0 c=.1e-14

vpulse:Vin5a "a31" 0 v1=0 v2=1.8 per=2e-9 pw=1e-9 tr=2e-11 tf=2e-11
+td=0.5e-9

xxor15a 1 0 "a30" "a31" "a32" "a15000" "tref" XOR_Thermal

r:R28a "a28" 0 r=1e5

r:R29a "a29" 0 r=1e5

r:R30a "a30" 0 r=1e5

r:R31a "a31" 0 r=1e5

r:R32a "a32" 0 r=1e5

** Source Resistance and Capacitance

*r:Rin0 212 0 r=1e3

*r:Rin1 213 0 r=1e3

*c:Cin2 212 0 c=1e-15

*c:Cin3 213 0 c=1e-15

*** Thermal Resistance and Capacitance

*r:Rtx14a "a14000" 1001 r=20e3
c:Ctx14a "a14000" 1001 c=10e-13

*r:Rtx15a "a15000" 1001 r=20e3
c:Ctx15a "a15000" 1001 c=10e-13

**DFFs
"vdd" "d" "clk" "q" "qb" "gnd" "temp" "tref"

xdff1a 1 "a2" "a32" "a4" "a2" 0 "a1000" "tref" DFF_Thermal1
xdff2a 1 "a22" 4 "a24" "a22" 0 "a2000" "tref" DFF_Thermal1

*** DFF Thermal Resistance and Capacitance
*r:Rtd1a "a1000" 1001 r=20e3
c:Ctd1a "a1000" 1001 c=10e-13
*r:Rtd2a "a2000" 1001 r=20e3
c:Ctd2a "a2000" 1001 c=10e-13

*4 connections each: activeC, activeB, handle1(thermal gnd); overglass
*4001+134=4135

*Connections for extracted thermal network
xtnetwork 14000 15000 1000 2000 "b14000" "b15000" "b1000" "b2000"
+"a14000""a15000" "a1000" "a2000" 1001 1001 1001 1001
+"og1"
+"og2"
+"og3"
+"og4"
+"og5"
+"og6"
+"og7"
+"og8"
+"og9"
+"og10"
+"og11"
+"og12"
+"og13"
+"og14"
+"og15"
+"og16"
+"og17"
+"og18"
+"og19"
+"og20"
+"og21"
+"og22"
+"og23"
+"og24"
+"og25"
+"og26"
+"og27"
+"og28"
/"og29" /"og30" /"og31" /"og32" /"og33" /"og34" /"og35" /"og36" /"og37" /"og38" /"og39" /"og40" /"og41" /"og42" /"og43" /"og44" /"og45" /"og46" /"og47" /"og48" /"og49" /"og50" /"act_c1" /"act_c2" /"act_c3" /"act_c4" /"act_c5" /"act_c6" /"act_c7" /"act_c8" /"act_c9" /"act_c10" /"act_c11" /"act_c12" **"act_c13" /"act_c14" /"act_c15" /"act_c16" **"act_c17" /"act_c18" /"act_c19" /"act_c20" /"act_c21" /"act_c22" 168
"act_c23"
"act_c24"
"act_c25"
"act_c26"
"act_c27"
"act_c28"
"act_c29"
"act_c30"
"act_c31"
"act_c32"
"act_c33"
"act_c34"
"act_c35"
"act_c36"
"act_c37"
"act_c38"
"act_c39"
"act_c40"
"act_c41"
"act_c42"
"act_c43"
"act_c44"
"act_c45"
"act_c46"
"act_c47"
"act_c48"
"act_c49"
"act_c50"
"act_b1"
"act_b2"
"act_b3"
"act_b4"
"act_b5"
"act_b6"
"act_b7"
"act_b8"
"act_b9"
"act_b10"
"act_b11"
"act_b12"
"act_b13"
"act_b14"
"act_b15"
"act_b16"
;++"act_b17"
;++"act_b18"
;++"act_b19"
;++"act_b20"
;++"act_b21"
;++"act_b22"
;++"act_b23"
;++"act_b24"
;++"act_b25"
;++"act_b26"
;++"act_b27"
;++"act_b28"
;++"act_b29"
;++"act_b30"
;++"act_b31"
;++"act_b32"
;++;++"act_b33"
;++"act_b34"
;++"act_b35"
;++"act_b36"
;++;++"act_b37"
;++"act_b38"
;++"act_b39"
;++"act_b40"
;++"act_b41"
;++"act_b42"
;++"act_b43"
;++"act_b44"
;++"act_b45"
;++"act_b46"
;++"act_b47"
;++"act_b48"
;++"act_b49"
;++"act_b50"
;++"act_a1"
;++"act_a2"
;++"act_a3"
;++"act_a4"
;++"act_a5"
;++"act_a6"
;++"act_a7"
;++"act_a8"
;++"act_a9"
;++"act_a10"
+"act_a11"
+"act_a12"
+"act_a13"
+"act_a14"
+"act_a15"
+"act_a16"
+"act_a17"
+"act_a18"
+"act_a19"
+"act_a20"
+"act_a21"
+"act_a22"
+"act_a23"
+"act_a24"
+"act_a25"
+"act_a26"
+"act_a27"
+"act_a28"
+"act_a29"
+"act_a30"
+"act_a31"
+"act_a32"
+"act_a33"
+"act_a34"
+"act_a35"
+"act_a36"
+"act_a37"
+"act_a38"
+"act_a39"
+"act_a40"
+"act_a41"
+"act_a42"
+"act_a43"
+"act_a44"
+"act_a45"
+"act_a46"
+"act_a47"
+"act_a48"
+"act_a49"
+"act_a50"
+ thermal_network

*Leave commented for a large number of outputs
*or risk crashing X11.
# Data Plotting Commands

- `.out plot term "act_a6" vt in "a6.out"
- `.out plot term "act_a7" vt in "a7.out"
- `.out plot term "act_a8" vt in "a8.out"
- `.out plot term "act_a9" vt in "a9.out"
- `.out plot term "act_a10" vt in "a10.out"
- `.out plot term "act_a11" vt in "a11.out"
- `.out plot term "act_a12" vt in "a12.out"
- `.out plot term "act_a13" vt in "a13.out"
- `.out plot term "a15000" vt in "a13.out"
- `.out plot term "act_a14" vt in "a14.out"
- `.out plot term "act_a15" vt in "a15.out"
- `.out plot term "act_a16" vt in "a16.out"
- `.out plot term "act_a17" vt in "a17.out"
- `.out plot term "a10000" vt in "a17.out"
- `.out plot term "act_a18" vt in "a18.out"
- `.out plot term "act_a19" vt in "a19.out"
- `.out plot term "act_a20" vt in "a20.out"
- `.out plot term "act_a21" vt in "a21.out"
- `.out plot term "act_a22" vt in "a22.out"
- `.out plot term "act_a23" vt in "a23.out"
- `.out plot term "act_a24" vt in "a24.out"
- `.out plot term "act_a25" vt in "a25.out"
- `.out plot term "act_a26" vt in "a26.out"
- `.out plot term "act_a27" vt in "a27.out"
- `.out plot term "act_a28" vt in "a28.out"
- `.out plot term "act_a29" vt in "a29.out"
- `.out plot term "act_a30" vt in "a30.out"
- `.out plot term "act_a31" vt in "a31.out"
- `.out plot term "act_a32" vt in "a32.out"
- `.out plot term "act_a33" vt in "a33.out"
- `.out plot term "a14000" vt in "a33.out"
- `.out plot term "act_a34" vt in "a34.out"
- `.out plot term "act_a35" vt in "a35.out"
- `.out plot term "act_a36" vt in "a36.out"
- `.out plot term "act_a37" vt in "a37.out"
- `.out plot term "a20000" vt in "a37.out"
- `.out plot term "act_a38" vt in "a38.out"
- `.out plot term "act_a39" vt in "a39.out"
- `.out plot term "act_a40" vt in "a40.out"
- `.out plot term "act_a41" vt in "a41.out"
- `.out plot term "act_a42" vt in "a42.out"
- `.out plot term "act_a43" vt in "a43.out"
- `.out plot term "act_a44" vt in "a44.out"
- `.out plot term "act_a45" vt in "a45.out"
A D-flip-flop (DFF) is a basic memory cell which in this macromodel subcircuit has been composed of reduced state variable electrothermal NAND gates. The following is a listing of the file `nandff_subckt.t.net`.

```plaintext
.subckt DFF_Thermal1 "vdd" "d" "clk" "q" "qb" "gnd" "temp" "tref"
  cmosinv:clkinv "vdd" "d" "clk" "q" "qb" "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand1 "vdd" "d" "clk" 1 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand2 "vdd" "d" "clk" 1 2 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand3 "vdd" 1 4 3 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand4 "vdd" 2 3 4 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand5 "vdd" 3 "clk" "q" "gb" "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand6 "vdd" 4 "clk" 5 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand7 "vdd" 5 "q" "qb" "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6
  cmos2nandt:nand8 "vdd" 6 "q" "qb" "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6

* Intermediate nodal capacitance
  c:c1 1 "gnd" c=1e-14
  c:c2 2 "gnd" c=1e-14
```

C.4.1 Electrothermal DFF Subcircuit Netlist

A D-flip-flop (DFF) is a basic memory cell which in this macromodel subcircuit has been composed of reduced state variable electrothermal NAND gates. The following is a listing of the file `nandff_subckt.t.net`.
C.4.2 Electrothermal XOR Subcircuit Netlist

The frequency multiplier utilizes an XOR gate in addition to a time delayed element or time delayed pulse element. The XOR gate subcircuit is composed of reduced state variable electro-thermal NAND gates. The following is a listing of the file xor_subckt.t.net.

*** Thermal Netlist of XOR Gate in fREEDA

.subckt XOR_Thermal "vdd" "gnd" "In1" "In2" "out" "temp" "tref"

  cmos2nandt:nand1 "vdd" "In1" "In2" 4 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6 rth=2e2
  cmos2nandt:nand2 "vdd" "In1" 4 5 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6 rth=2e2
  cmos2nandt:nand3 "vdd" "In2" 4 6 "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6 rth=2e2
  cmos2nandt:nand4 "vdd" 5 6 "out" "gnd" "temp" "tref" ln=0.18e-6
  +wn=1.8e-6 lp=0.18e-6 wp=2.4e-6 rth=2e2

* Intermediate nodal capacitance
  c:c1 4 "gnd" c=1e-15
  c:c2 5 "gnd" c=1e-15
  c:c3 6 "gnd" c=1e-15
  c:c4 "out" "gnd" c=1e-15
.ends

178
C.5 3DIC Hotpost Thermal Subcircuit

This is the extracted resistive physical thermal netlist which is attached to the main circuit file. It has been omitted due to length in this version of the dissertation, but is available in both resolutions as hotspot_extracted_101x54_10u.net and hotspot_extracted_101x54_5u.net.
Appendix D

Other Measurements

D.1 Introduction

The following documents the Thermal Verification study of a resistive thermal test chip. The test chip considered here is a test vehicle fabricated in the MITLL 3DIC process [60] with resistive elements and large power transistors designed to be heat sources.

This appendix presents static thermal images captured experimentally using thermal equipment at Wright Patterson Air Force Base. The measurements were used to provide experimental validation of thermal modeling software. The test chip was designed by Filipp Akopyan, Carlos Tadeo Ortega Otero, David Fang, Sandra Jackson and Rajit Manohar of Cornell University.

D.2 Description

Power transistors were fabricated in extra space on the periphery of a chip so that their electro-thermal effects could be studied. Tantalum nitride, gate polysilicon, and resistive polysilicon materials were used. Also included were tantalum nitride resistors, termed power resistors, which acted as heat sources. The resistive heating sections of the chip were also measured. Figure D.1 is a thermal image of a section of the die focusing on the region with a power resistor. Figure D.1 shows power resistors at two different temperatures with temperatures
established by applying different voltages to the resistor. Also visible are wire bonds seen as green shadows on the pads in the image. A greater magnification of the thermal image is shown in Figure D.2.

Figure D.1: Thermal image of a section of the die area with power resistor. The two parts of the figure are at different temperatures.

In Figure D.2 the underlying grid of resistive material is visible through the top layer of the device. The grid is only visible by thermal imaging and is not apparent visually.

The temperature measured along a line through the power resistor is shown in Figure D.3. Thermal bunching is an important effect to be understood in thermal analysis. Current flowing in close proximity to another source realizes the additive nature of heat flux, allowing an accumulation of higher temperatures. This is important in determining qualitatively how much of an area needs to be simulated to determine the effect on another area. Due to CPU and memory limitations, it is ideal to neglect as much as possible while remaining completely accurate. This histogram, Figure D.3, shows that it is safe to neglect heat sources over 9 µm away per
Figure D.2: Thermal image of a section of the die area with a power resistor. The two parts of the figure are at different temperatures.

$40^\circ C$ rise in this particular material and process. The total heat flux can be calculated from the resistor’s voltage and current, see Figures D.4 and D.5.

Figure D.3: The thermal profile is showed as a profile of the power resistor.

These results were used in static thermal analysis of the test chip. They can further be used
Figure D.4: Graphed measurements of temperature vs. voltage measured at the power resistor IC.

for calibration of simulations of differing materials.
Figure D.5: Graphed measurements of temperature vs. current measured at the power resistor IC.