ELECTROTHERMAL SIMULATION AND TEMPERATURE-SENSITIVE RELIABILITY DIAGNOSIS FOR CMOS VLSI CIRCUITS

BY

YI-KAN CHENG

B.S., National Chiao-Tung University, Taiwan, 1991
M.S., University of Southern California, 1993

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1997

Urbana, Illinois

© Copyright by Yi-Kan Cheng, 1997

ELECTROTHERMAL SIMULATION AND TEMPERATURE-SENSITIVE RELIABILITY DIAGNOSIS FOR CMOS VLSI CIRCUITS

Yi-Kan Cheng, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 1997
Sung-Mo (Steve) Kang, Advisor

Over the years, state-of-the-art technologies have continued to push the ultra-large-scale-integrated (ULSI) chip to higher clock speed and packing density. The speed requirement causes large power consumption and the packing density requirement results in large power density (power per unit area). One direct impact of the increasing power density is the dramatic on-chip temperature rise. For a chip under normal operating conditions, the temperature rise can be as much as a few tens of degrees above the ambience. Without good thermal engineering, significantly nonuniform temperature distribution can lead to a considerable on-chip temperature gradient. Although many research efforts have been focusing on the development of low power and new package design for better integrated-circuit (IC) reliability, the thermal problems still exist and deserve more attention. The temperature rise and temperature gradient have strong effects on both chip performance and reliability. Therefore, temperature effects must be taken into consideration for performance and reliability analyses. The temperature information can be obtained by using the computer-aided-design (CAD) tools before the chip has actually been fabricated.

The goal of the work presented in this dissertation is to develop an electrothermal simulation methodology for temperature-profile estimation, hot-spot identification, and circuit reliability prediction for CMOS VLSI chips. This methodology has been implemented in a CAD tool, ILLIADS-T. ILLIADS-T follows a decoupled electrical/thermal simula-

tion flow which has been proven to be very efficient. A temperature-dependent MOS device modeling technique has been developed. This technique accurately models the device mobility at a wide range of temperatures and couples it into a regionwise-quadratic MOS device model. Based on this device model, accurate temperature-dependent power and timing information can be obtained. To calculate the on-chip temperature profile, a chip-level thermal simulation framework has been developed. This framework supports three different thermal simulation techniques which are designed to identify on-chip hot spots, pinpoint the hot-spot temperatures, and profile the full-chip steady-state temperature. A heat-transfer macromodel for the packaging structure is also proposed for the accurate and efficient thermal simulation of the package and heat sink. In order to verify the ILLIADS-T simulation results, a tester chip has been designed and fabricated. Very good agreement between simulation and experiment is observed.

ILLIADS-T has been successfully applied to the electromigration (EM) reliability diagnosis and timing analysis. By considering both transistor and interconnect temperatures, the temperature-sensitive EM-induced mean time-to-failure and critical path timing are estimated and discussed. ILLIADS-T has also been used to simulate the temperature profile of a commercial microprocessor with the most advanced packaging structure. Other issues such as temperature-driven module placement and package design can be also carried out by using the ILLIADS-T electrothermal simulator.

To my parents,
and my dear wife, Hui-Chun

ACKNOWLEDGMENTS

I would like to thank my research advisor, Professor Sung-Mo (Steve) Kang, for his encouragement and inspiration during my three-year course of study. This research would not have been possible without his guidance.

I would like to especially thank Professor Elyse Rosenbaum for numerous invaluable suggestions and discussions during my research. I also thank Professors Farid Najm and Janak Patel for serving on my dissertation committee and providing me with many useful comments on this research.

I appreciate the support received from the Circuit Modeling Group in the Technology Computer-Aided Design (TCAD) Department of Intel Corporation, particularly from Dr. Shiuh-Wuu Lee. I thank all of my officemates, group seniors, and the members in the VLSI reliability group, who have made my life at the University of Illinois very enjoyable.

I would like to express my appreciation and gratitude to my parents and my brother for their encouragement and love, which led me to pursue the doctoral degree. Finally, I owe my deepest appreciation to my wife, Hui-Chun. Without her understanding and love, this research would not have been possible. As a wife and a friend, she has always believed in me and been there for me. I thank her for everything that she has done.

This research was supported in part by grants from Intel Corporation, Semiconductor Research Corporation, and Rome Laboratory.

TABLE OF CONTENTS

CHAPTER PAGE

1 INTRODUCTION 1
1.1 Motivation 1
1.2 Overview of Electrothermal Simulations 3
1.2.1 Previous work 4
1.3 Our Approach 11
1.4 Organization of the Dissertation 16

2 TEMPERATURE-DEPENDENT MOS DEVICE MODELING . . . 17
2.1 Introduction 17
2.2 ILLIADS Fast-Timing Simulator 18
2.2.1 Primitive formation and solutions 18
2.2.2 Simulation strategies 20
2.2.3 Power estimation using ILLIADS 23
2.3 Regionwise Quadratic (RWQ) Modeling 24
2.3.1 Mobility modeling 26
2.3.2 Proposed temperature-dependent mobility model 29
2.3.3 $V_{T0}(T)$ and $\mu_0(T)$ extraction 29
2.3.4 Preliminary mobility and RWQ-fitting results 31
2.3.5 RWQ Model I 35
2.3.6 RWQ Model II 36

3 THERMAL SIMULATION FRAMEWORK AND INCREMENTAL ELECTROTHERMAL SIMULATION 40
3.1 Introduction 40
3.2 Substrate/Package Modeling 41
3.3 Formulation 44
3.3.1 Fast thermal analysis 44
3.3.2 Numerical approach 52
3.3.3 Analytical approach 60
3.3.4 Discussion 62
3.4 Package Simulation 65
3.4.1 Modeling of convective boundaries 65
3.4.2 Modeling of heat flow paths 67
3.5 Incremental Electrothermal Simulation 71

4 **VERIFICATION OF ILLIADS-T AND SIMULATION RESULTS** . 78
4.1 Tester Chip Design and Calibration 78
4.2 Verification of ILLIADS-T 81
4.3 ILLIADS-T Simulation Examples 87

5 **TEMPERATURE-SENSITIVE RELIABILITY AND PERFORMANCE ANALYSIS USING ILLIADS-T** 92
5.1 Motivation 92
5.2 Electromigration Diagnosis 93
5.2.1 Introduction 93
5.2.2 Temperature-dependent electromigration reliability diagnosis flow 95
5.2.3 Interconnect temperature estimation 96
5.2.4 iTEM simulation examples and summary 101
5.3 Timing Analysis 102
5.3.1 Introduction 102
5.3.2 Temperature-dependent gate and RC delays 106
5.3.3 Monte-Carlo power estimation for on-chip temperature profiling . 107
5.3.4 Thermal simulation for timing analysis 109
5.3.5 Simulation results 111

6 **CONCLUSIONS** 115
6.1 Summary 115
6.2 Future Research 117

REFERENCES 119

VITA 125

LIST OF TABLES

Table		Page
2.1	SPICE-model parameters and device dimensions for generating I-V data.	32
3.1	Error function approximations.	47
3.2	Comparison between iTEMP and THUNDER simulation results.	63
3.3	Violation rate by using the FTA method.	64
3.4	Definition of the symbols in Fig. 3.20.	69
4.1	Activation status of Rosc3s.	81
4.2	ILLIADS-T simulation results of the tester chip.	87
4.3	ILLIADS-T simulation results.	89
4.4	Materials and thermal parameters for the packaging structure.	89
4.5	ILLIADS-T simulation results.	91
5.1	The ISCAS85 benchmark circuits.	112
5.2	Simulation results of ISCAS85 benchmark circuits.	113

LIST OF FIGURES

Figure Page

1.1 Applications of electrothermal CAD tools. 2
1.2 Elements of electrothermal simulations. 3
1.3 Coupled electrothermal simulation procedure. 5
1.4 (a) An RC circuit example, (b) the dc circuit for the first-moment generation, and (c) the dc circuit for finding the second moment. 8
1.5 The integrator circuit used to implement the solution of the 3-D heat diffusion equation. 11
1.6 Flowchart of ILLIADS-T electrothermal simulation. 12

2.1 General MOS circuit primitive used in ILLIADS. 18
2.2 Illustrations of SCC formation and topological sort: (a) the original circuit, (b) the digraph representation, and (c) the condensed digraph after topological sort. 21
2.3 Example of transistor merging and internal node elimination. 22
2.4 Primitive mapping for the circuit shown in Fig. 2.3 after the transistor merging process. 23
2.5 DCCB power calculation using ILLIADS. 24
2.6 Regionwise partition of the (V_{ds}, V_{gse}) plane. 25
2.7 (a) Extracted $V_{T0}(T)$, and (b) Extracted $\mu_0(T)$. 30
2.8 RWQ model fits for SPICE-generated data at 27°C: (a) NMOS and (b) PMOS. 32
2.9 RWQ model fits for SPICE-generated data at 100°C: (a) NMOS and (b) PMOS. 33
2.10 Fitted $\mu_0(T)$ using experimental data. 33
2.11 RWQ-fitting result for the experimental data at $T = 30$°C. 34
2.12 RWQ-fitting result for the experimental data at $T = 90$°C. 35
2.13 I-V characteristics at 100°C using RWQ(77). 37
2.14 RWQ-fitting result at 90°C with mobility optimization. 38
2.15 (a) Regionwise mobility fitting, and (b) fitting quality at 90°C. 39
2.16 Output waveforms of a nine-stage inverter chain. 39

3.1 iTEMP thermal simulation framework. 42
3.2 Illustration of effective heat transfer macromodeling. 43
3.3 Method of images. 45
3.4 Error function approximation. 47
3.5 Transformation 1: Constrain the observation point to the first quadrant. . . 48
3.6 Transformation 2: Constrain t_{a1} to be larger than t_{b1}. 49
3.7 Eight cases under six constraints. 49

3.8 FTA example. 50
3.9 (a) Top view of a part of the chip containing heat sources, and (b) 3-D view of grid point (i, j, k). 54
3.10 (a) Analogous thermal circuit to Fig. 3.9(a), and (b) thermal conductances from (i, j, k) to adjacent grids. 55
3.11 Analogy between thermal and electrical circuits. 55
3.12 (a) Top view of a part of the chip comprised of composite materials, and (b) 3-D view of grid point (i, j, k). 57
3.13 (a) Analogous thermal circuit to Fig. 3.12(a), and (b) thermal conductances from (i, j, k) to adjacent grids. 59
3.14 Equivalent thermal circuit at the convective boundary. 59
3.15 Chip structure and heat source locations in the experiment. 63
3.16 Layout of the chip containing three heat sources. 66
3.17 Temperature profiles along the x direction at $y = 500$ μm for three different h^e values. 66
3.18 Unit-level layout of the microprocessor chip. 67
3.19 Cross-sectional view of a flip-chip package. 68
3.20 Equivalent thermal circuit of the flip-chip package. 69
3.21 Method to determine the thermal resistances for heat flowing through the carrier aside to the lids. 70
3.22 A bend structure of the aluminum lid. 70
3.23 On-chip temperature contour for the first experiment. 72
3.24 On-chip temperature contour for the second experiment. 73
3.25 On-chip temperature contour for the third experiment. 74
3.26 Convergence plot for power and temperature. 75
3.27 Illustration of incremental latency; the nominal waveforms are shown in solid lines, while the perturbed waveforms are in dashed lines. 76

4.1 Layout of tester chip, where long blocks are Rosc149s and short blocks are Rosc3s. 79
4.2 Four-terminal configuration for diode measurement. 80
4.3 Diode calibration example (D1). 81
4.4 Simulated temperature profile for Expt. 2. 82
4.5 Simulated temperature profile for Expt. 1. 83
4.6 Comparison between simulated and measured temperatures for D1. 83
4.7 Comparison between simulated and measured temperatures for D2. 84
4.8 Comparison between simulated and measured temperatures for D3. 84
4.9 (a) Measured and (b) simulated waveforms for Expt. 8. 85
4.10 (a) Measured and (b) simulated waveforms for Expt. 7. 86
4.11 (a) Measured and (b) simulated waveforms for Expt. 5. 86
4.12 (a) Measured and (b) simulated waveforms for Expt. 1. 87
4.13 Layout of the 10-bit negative adder. 88
4.14 Layout of the simulated chip. 90

4.15 Packaging structure used in the simulation example. 90
4.16 Output waveforms of the 10-bit negative adder. 91

5.1 The temperature effect on EM reliability. 95
5.2 Simulation flowchart of iTEM. 96
5.3 The interconnect on insulator structure. 97
5.4 Interconnect temperature as a function of its current density, assuming the following physical parameters are used: $t_i = 2\mu$m, $t = 0.5\mu$m, $w = 2\mu$m, $\rho_0 = 3.6 \times 10^{-6}\Omega\cdot$cm, $\beta = 4.04 \times 10^{-3}$K^{-1}, $K_i = 1.835$W/(K$\cdot$m), and $T_s =$ 300 K. 98
5.5 Lumped thermal model for a general interconnect structure. 99
5.6 The procedures for interconnect temperature estimation. 100
5.7 The layout of the 10-bit negative adder. 101
5.8 iTEM-predicted MTF for the 10-bit negative adder. 102
5.9 Generic flowchart of timing analysis. 103
5.10 False path example. 105
5.11 Example of a distributed RC tree. 107
5.12 Thermal boundary conditions used for the temperature-dependent timing simulation. 112
5.13 The temperature profile and the DCCB distribution on the longest path of C6288. The solid lines are the simulated temperature contour and the small diamonds are DCCBs on the longest path. 114

CHAPTER 1

INTRODUCTION

1.1 Motivation

Due to the increasing component density, higher operating speed, and larger scale of integration, the power density and on-chip temperature in integrated circuits continue to increase. Furthermore, the temperature of packaged very-large-scale-integrated (VLSI) circuits can vary by as much as a few tens of degrees from the center to an edge of the chip. Because the failure rate of microelectronic devices depends heavily on the localized operating temperature, hot spots due to high local-power dissipation have become a long-term integrated-circuit (IC) reliability concern in diverse applications such as high-performance microprocessors and digital signal-processing chips. Because of the complexity of a VLSI chip, the verification of chip performance at various operating temperatures relies heavily on computer simulations. Therefore, a new fast and accurate thermal reliability diagnostic tool is required. Once the temperature profile is determined, several important issues can be addressed as shown in Fig. 1.1. It is clear that the thermal engineering can be used not only for reliability checking, but as an additional degree of freedom to enhance circuit performance.

Although many research efforts have been undertaken to deal with the solution of electrothermal problems in devices and small-scale electronic circuits [1][2][3], there have

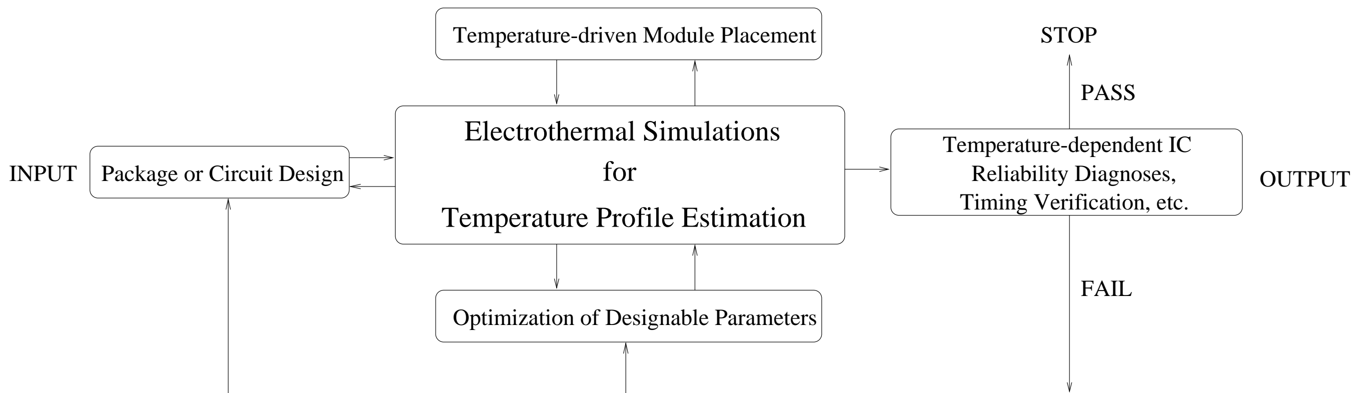


Figure 1.1 Applications of electrothermal CAD tools.

been no attempts to provide electrothermal simulation capability at the chip level. The chip-level tools must avoid the following bottleneck problems:

- simulation inefficiency resulting from commonly used *coupled* electrical and thermal simulations, i.e., coupled electrothermal simulations
- slow execution speed of SPICE-like simulators for computing power dissipation of active devices
- lack of thermal simulator specifically suitable for VLSI chips with complex packaging structures
- improper use of thermal boundary conditions
- dependence on complex models of advanced metal-oxide-semiconductor (MOS) transistors, including temperature models
- lack of automated top-down procedures from the circuit layout to the user-specified output

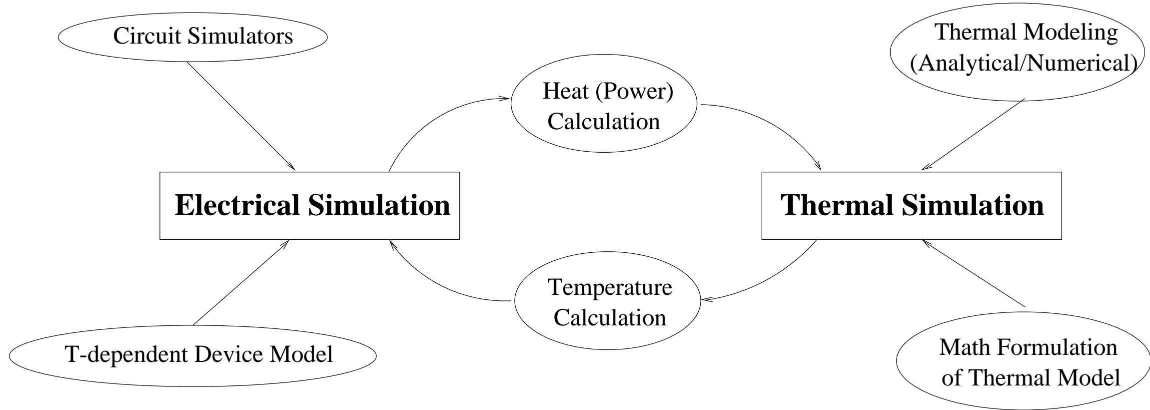


Figure 1.2 Elements of electrothermal simulations.

The work presented in this dissertation is intended to remove the above problems by introducing a new methodology for temperature-profile estimation, hot-spot identification, and the resulting circuit reliability prediction for complementary MOS (CMOS) VLSI chips. An electrothermal simulator, ILLIADS-T, has been developed based on this methodology.

1.2 Overview of Electrothermal Simulations

Simply put, electrothermal simulation consists of electrical and thermal simulations. The purpose of electrical simulation is to obtain the information on power dissipation and performance of devices or circuits. On the other hand, the thermal simulation is used to find the temperature profile and to update all the temperature-dependent physical parameters of the device or circuit model. This is illustrated in Fig. 1.2. The loop in Fig. 1.2 forms the basic mechanism of electrothermal simulation. The electrical

and thermal components must be self-consistent for the system to remain stable, or the thermal runaway effects may occur.

1.2.1 Previous work

Fukahori and Gray [1] comprehensively addressed the simulation of ICs in the presence of electrothermal interaction. Their focus was on the analog circuits where thermal feedback can severely degrade the circuit performance and distort the voltage transfer characteristics.

The electrothermal simulation procedure in [1] is illustrated in Fig. 1.3. A coupled set of nonlinear electrothermal equations is first generated. Next, those equations are represented by a matrix form and then linearized and solved by using the Newton-Raphson method. The linearized circuit matrix contains three parts: (1) Elements corresponding to the electrical circuit (Y_V), (2) elements corresponding to the thermal circuit (Y_{th}), and (3) elements corresponding to the coupling between the two circuits. The thermal circuit was generated by using the finite-difference method for the simplified die-header structure. Elements corresponding to the coupling between the two circuits are the thermally controlled current sources corresponding to the temperature effects on the electrical physical parameters, and the electrically controlled power sources corresponding to the power dependence of the node voltages. Once the matrix is solved and the dc solution is found at time t, as illustrated in Fig. 1.3, the transient solution for the temperature and the node voltage can be calculated by utilizing the preferred integration formula. In [1], the trapezoidal integration technique was employed. The above procedures are similar to those used in circuit simulation programs such as SPICE.

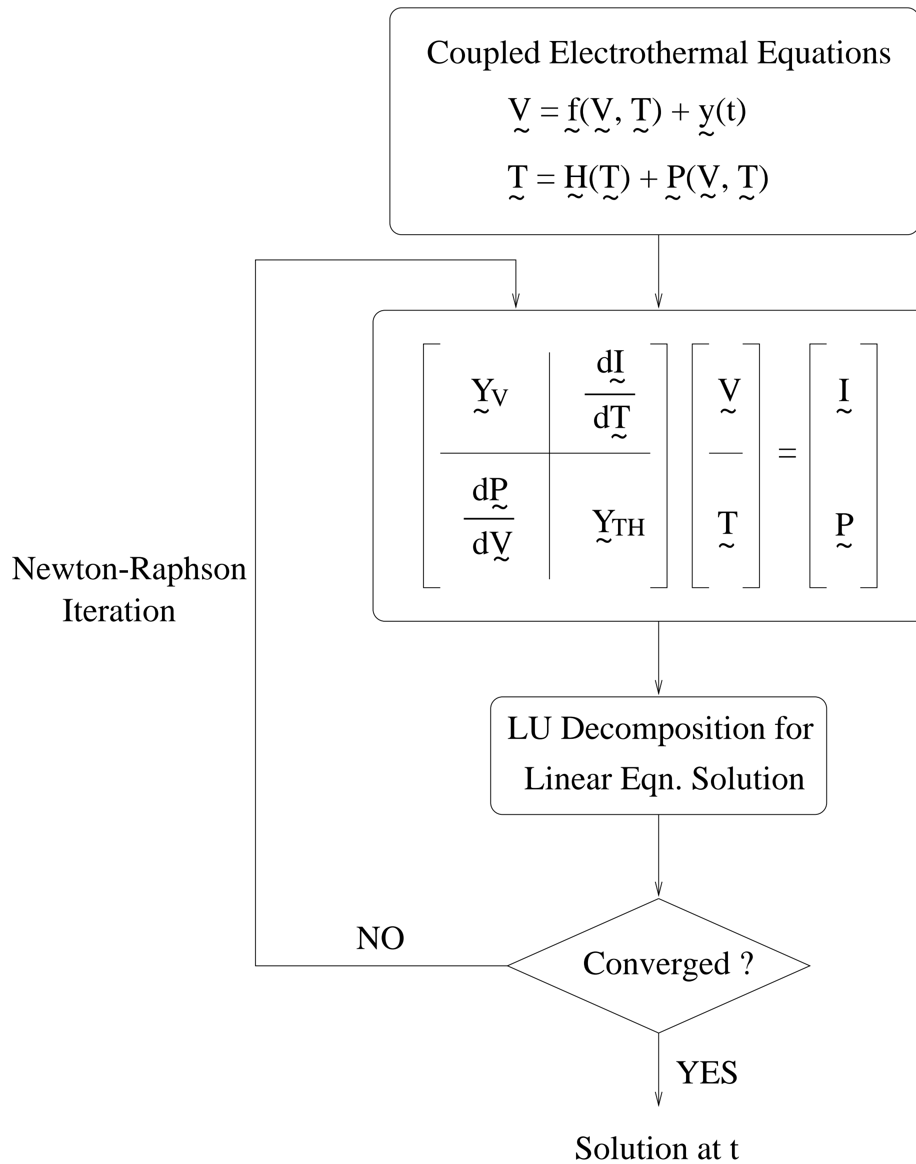


Figure 1.3 Coupled electrothermal simulation procedure.

The electrothermal simulator in [1] was applied to several analog circuits for the prediction of electrothermal interactions, both in dc transfer characteristics and in transient response. It was also pointed out that the simulation time was typically a factor of ten greater than the case when only the electrical effects were considered.

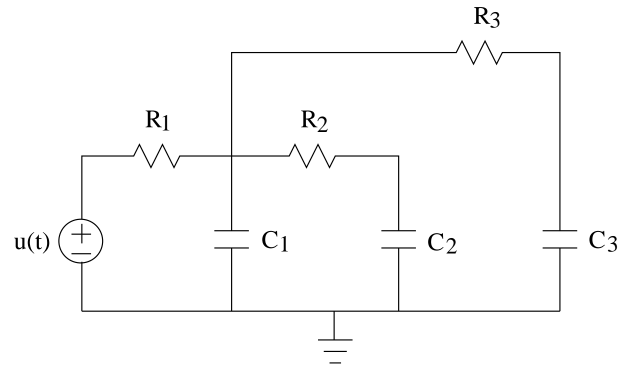
In 1982, a transistor-level electrothermal simulator was developed by Latif et al. [4]. It aimed at finding the temperature-dependent behavior of the power bipolar transistor. The simulated device was partitioned into $n \times m$ (in 2-D case) sections connected in parallel by appropriate base resistors, where each section operated at its own temperature. A temperature-dependent Ebers-Moll model was used for each section. This model included the effects of avalanche multiplication, basewidth modulation, and current-gain variations. The thermal network for the device was generated by using the 3-D finite-difference approach.

Two numerical techniques were proposed in [4] to solve the coupled electrothermal circuits. The first one was called direct method, which is similar to the method proposed earlier in [1]. The second technique was called the relaxation method. This method divided the original problem into electrical and thermal systems. They were solved separately and the solutions were obtained by applying successive relaxation between the two systems. Both techniques had their own advantages and disadvantages. The direct method was more general and powerful for analyzing different problems such as dc, transient, and dc transfer characteristics. However, it was computationally more expensive and might not be able to handle all nonlinearities of the system. The relaxation method was more efficient, but convergence problems might occur under some biasing conditions.

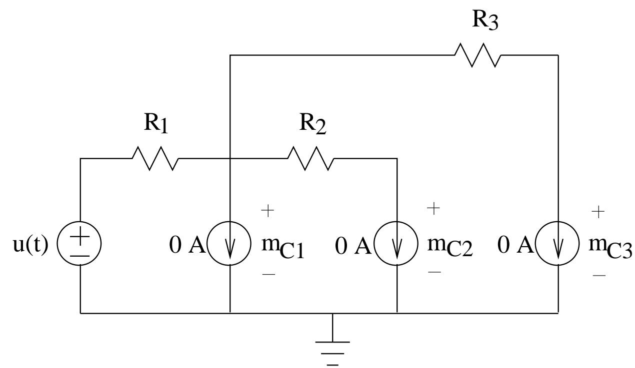
Lee et al. developed a coupled electrothermal simulator for ICs in 1993 [2]. Its purpose was similar to that of [1], but with the focus on improving the simulation efficiency while preserving the accuracy. For the dc analysis, the incomplete Choleski conjugate gradient (ICCG) method was used. For the transient analysis, the macromodeling method based on asymptotic waveform evaluation (AWE) [5] was employed.

The ICCG method is one of the relaxation methods that does not require an expensive LU factorization process to solve the network matrix as in the direct method. By the combination of incomplete Choleski decomposition and conjugate gradient optimization, the ICCG method is known to be very efficient in solving symmetric and diagonally dominant systems such as 3-D interconnect structures or 3-D thermal networks. Simulation results for a 741 operational amplifier showed that the CPU time saved was 93 % by using the ICCG method compared to the direct method [2]. More CPU and memory savings are expected for larger circuits.

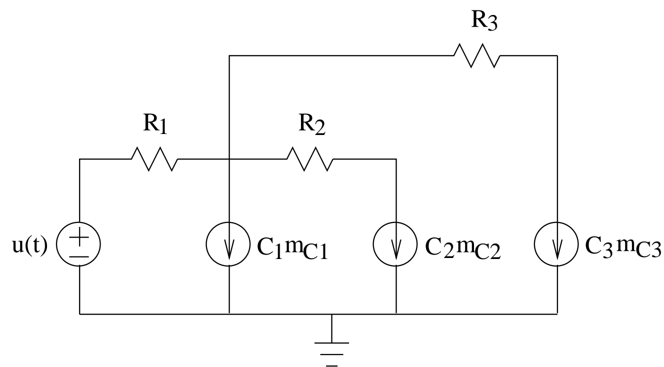
AWE is a technique to find the time-domain response of a linear system by utilizing a reduced set of approximate poles and residues in the frequency-domain transfer function. These poles and residues are determined by applying a moment-matching method such as the Padé approximation [6]. The manner in which moments for a linear system are calculated is to successively perform the dc analysis of the system. For example, consider the RC circuit in Fig. 1.4(a). The first set of moments of the circuit is found by transforming the circuit in Fig. 1.4(a) into Fig. 1.4(b), replacing capacitors with zero-valued constant-current sources, and calculating the voltages across the current sources. The voltages m_{C1}, m_{C2}, and m_{C3} in Fig. 1.4(b) are the resulting first set of moments. The successive generations of higher-order moments are accomplished by setting the driver



(a)



(b)



(c)

Figure 1.4 (a) An RC circuit example, (b) the dc circuit for the first-moment generation, and (c) the dc circuit for finding the second moment.

to zero and replacing each current source with the product of its previous moment and capacitance value. For illustration, the second set of moments for the circuit in Fig. 1.4(a) is found as shown in Fig. 1.4(c).

A linear thermal system can always be described in terms of the state equations in (1.1),

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}, \tag{1.1}$$

where x is the state vector, u is the input vector, y is the output vector, and D is the vector related to the electrothermal coupling. Therefore, the AWE technique can be directly applied to this system to obtain the transient temperature response, which is computationally much more efficient when compared to a conventional time-domain integration method such as in SPICE. The transient electrothermal simulation was performed on the 741 operational amplifier, and the CPU time saved by using the AWE technique was about 85% in comparison to the trapezoidal integration method [2]. Transient simulations were done by Lee et al. for both bulk silicon and silicon-on-insulator (SOI) technologies, and a comparison of thermal effects between the two technologies was made.

A new circuit-level electrothermal simulator, iETSIM, was introduced by Díaz et al. in 1994 [3]. It simulates the transient electrothermal effects, with an emphasis on the electrical overstress (EOS) and electrostatic discharge (ESD) applications. ESD is one of the most prevalent causes for IC failures due to the high-current and short-duration stress. Under such a stress, the breakdown phenomenon of a device is important. Because the second breakdown is thermally originated, electrothermal simulation is essential for an accurate ESD-induced failure analysis.

iETSIM is a coupled transient electrothermal simulator. To find the node voltages and circuit temperatures, a set of coupled electrothermal equations is formed and solved by using the standard modified nodal analysis (MNA) technique as shown in Fig. 1.3. For the electrical part, a new model and algorithm for avalanche breakdown were developed for accurate ESD/EOS simulation. This new algorithm was shown to be much simpler, more robust and more efficient than the algorithms introduced earlier in [7]. For the thermal part, a novel temperature model based on an electrical analog implementation of the time-dependent 3-D heat-diffusion equation was developed. It employed the solution of the 3-D heat-diffusion equation derived by Dwyer et al. [8]. For a heat source with dimensions $a \times b \times c$ and a constant power value P_0, the transient temperature distribution due to this source can be written as [8]

$$T(\vec{r}, t) = T_0 + \frac{P_0}{\rho C_p abc} \int_0^t G(x, a, \tau) G(y, b, \tau) G(z, c, \tau) d\tau. \tag{1.2}$$

In (1.2), $\vec{r}$ is the location of the observation point with respect to the center of the heat source, T_0 is the ambient temperature, ρ is the mass density, C_p is the specific heat, and $G(x, a, \tau)$, $G(y, b, \tau)$, and $G(z, c, \tau)$ are the Green's functions.

In iETSIM, the integral over time in (1.2) is evaluated by using an electrical equivalent integrator circuit shown in Fig. 1.5. In this circuit, a power monitor (P_0) and a time-dependent resistor (R) are provided to convert power to the temperature rise above the ambient temperature T_0. The time-dependent resistor can be obtained from (1.2) and is given as

$$R(\vec{r}, t) = \frac{\rho C_p abc}{CG(x, a, t) G(y, b, t) G(z, c, t)}, \tag{1.3}$$

where C is chosen so that the matrix entries become more even, and its typical value is 1 μF.

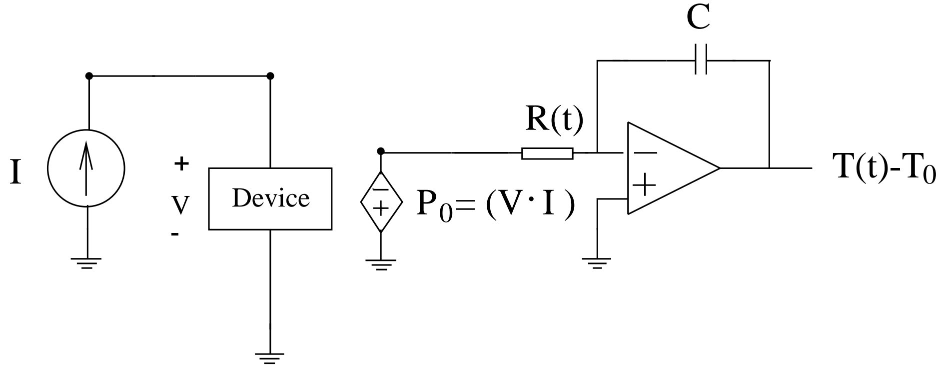


Figure 1.5 The integrator circuit used to implement the solution of the 3-D heat diffusion equation.

By using the implementation in Fig. 1.5, iETSIM is more efficient than using the relaxation method in [4], especially when circuits contain more than one device or when the temperature gradients are steep under ESD stress. In order to simulate the complete coupling between various heat sources in an ESD protection circuit, the current summation property of the integrator can be used as suggested by the superposition theory [3].

1.3 Our Approach

A simplified flowchart of ILLIADS-T electrothermal simulation procedure is shown in Fig. 1.6. The main features of ILLIADS-T are listed below.

1. To achieve the computational time efficiency required by large circuits, ILLIADS-T uses a fast-timing simulator, ILLIADS, to calculate the power dissipated by each *logic gate*. Each gate is then viewed as a heat source in our thermal simulation.

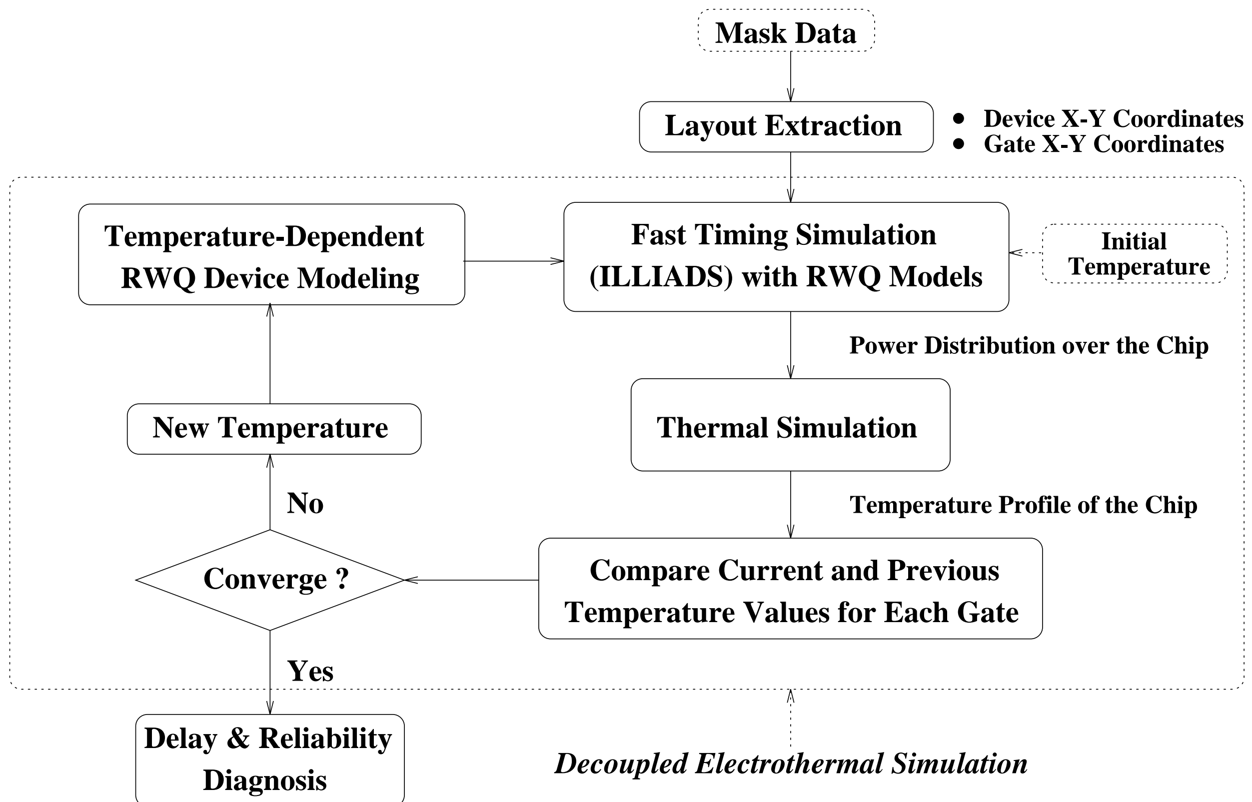


Figure 1.6 Flowchart of ILLIADS-T electrothermal simulation.

ILLIADS provides the following advantages: (1) The speedup of ILLIADS over SPICE-like programs increases linearly with the circuit size as measured in terms of the transistor count, (2) the speedup can be further enhanced by introducing the *incremental* electrothermal simulation technique, and (3) an accurate temperature-dependent modeling method for the MOS device is developed based on the region-wise quadratic (RWQ) modeling technique [9]. With this method, the accuracy of delay and power values estimated by ILLIADS is comparable to SPICE for a wide range of temperatures (27°C - 120°C).

2. Existing *coupled* electrothermal simulators are inefficient in nature [1][2][3]. The total simulation time is first divided into many small time intervals, then the power and temperature values are updated and coupled for each time interval. This approach is only suitable for the transient simulation on small circuits. ILLIADS-T, which is designed to find the chip-level steady-state temperature distribution and the resulting circuit performance, uses a much more efficient approach. It starts with an initial guess of the average chip temperature and then calculates the average power for each gate based on the current waveform drawn from the power supply. Next, the gate power values are fed to the thermal simulator to estimate the temperature profile. The temperature profile is then used to update the device model parameters for the second round of power calculation. This process continues until convergence is obtained and the steady-state temperature profile has been found. Note that our approach *decouples* the power and temperature calculations. This approach is justified because the chip temperature does not follow the instantaneous power dissipation; instead, it is virtually constant after reaching the

steady state. In addition, the time required for the on-chip temperature to reach its steady state is several orders of magnitude longer than the clock signal period in digital circuits [10]. Therefore, the average power rather than the instantaneous power is used in our temperature calculation.

3. Because existing electrothermal simulators were developed mainly for the temperature profile estimation of SSI or MSI circuits [1][2][3], thermal boundary conditions (BCs) were simplified. Moreover, the 1-D/2-D thermal simulations were usually adopted. For VLSI/ULSI chips with complex packaging structures, the simplified BCs and 1-D/2-D approaches may not be valid. To handle this problem, we have developed a new thermal simulator, iTEMP, to solve 3-D heat equations for the chip substrate, while modeling the packages and heat sinks as *effective* thermal resistances. iTEMP can handle various BCs at any side of the chip so that it removes the limitations on conventional electrothermal simulators. A hierarchical approach is also developed in our thermal simulation framework to quickly identify the on-chip hot spots and to subsequently pinpoint the hot-spot temperatures.

4. In order to achieve the top-down automation, once users specify the chip dimensions, packaging materials, device I-V data and thermal parameters, ILLIADS-T needs only the layout description file (in CIF or GDSII format) to find the steady-state temperature profile, and the corresponding circuit performance and reliability.

5. By using the RWQ modeling technique instead of the complex MOS models as in [11], temperature-dependent power and delay estimation can be done in ILLIADS-T even when only measured data are available and the MOS models have not been

fully developed or characterized. This makes ILLIADS-T device-model-independent and thus, applicable to advanced CMOS technologies.

Referring back to Fig. 1.6, the primary input to ILLIADS-T is the layout description file of the target VLSI chip. A layout extractor has been developed to obtain the electrical circuit that the layout represents, as well as to identify the location of each device. Our layout extractor employs the *scanline extraction* algorithm [12]. Its time complexity is $O(N \log N)$ with an expected memory space complexity of $O(\sqrt{N})$, where N is the number of rectangles in the layout description. A standard device specification in the netlist generated by our layout extractor is shown as follows:

MOS_name ND NG NS NB MODEL_name ⟨L=VAL⟩ ⟨W=VAL⟩ ⟨ AD=VAL⟩ ⟨ AS=VAL⟩ ⟨PD=VAL⟩ ⟨PS=VAL⟩ XMIN YMIN XMAX YMAX,

where XMIN, YMIN, XMAX, and YMAX define the bounding box of a MOS device, and MODEL_name specifies a particular RWQ model for a MOS device.

ILLIADS-T then calculates the bounds of each logic gate according to the coordinates of the bounding boxes of MOS devices within this gate. Next, the average power dissipation from each gate at the initial temperature is calculated by ILLIADS. iTEMP will take as input the power values and the coordinates of heat sources to calculate the on-chip temperature profile by solving the heat equations, in particular, the average temperature of each gate is found. At this stage, each gate has its updated local temperature and ILLIADS must be rerun to find the new average power values under the new temperature distribution. This iterative procedure stops when the updated temperature of each gate no longer has any significant change from the previous value. Empirical results

shown in Chapter 4 indicate that this process is efficient and usually converges after two or three iterations. Note that in CMOS circuits, the short-circuit power accounts for approximately 25% of the total IC power consumption [13]. The temperature-induced variations of the short-circuit power and/or the switching activity are what necessitate a few iterations during ILLIADS-T simulation.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, we present the temperature-dependent MOS device models for fast-timing simulation. A new mobility model for the RWQ MOS modeling is also proposed. A thermal simulation framework is demonstrated in Chapter 3. This framework contains three distinct thermal simulation techniques and their detailed modeling and mathematical formulation are presented. The package simulation method is also described in this chapter. In order to verify the simulation results, we have designed a tester chip and had it fabricated and packaged by MOSIS. The tester chip design and the comparison between measurement and ILLIADS-T simulation results will be presented in Chapter 4. Other ILLIADS-T simulation examples are also given in Chapter 4. Applications of ILLIADS-T to temperature-sensitive electromigration (EM) diagnosis and timing analysis are presented in Chapter 5. Conclusions and suggestions for future research are given in Chapter 6.

CHAPTER 2

TEMPERATURE-DEPENDENT MOS DEVICE MODELING

2.1 Introduction

The shortcoming of typical fast-timing simulation approaches lies in the lack of simulation accuracy. A major source of inaccuracy is due to the inadequate mapping of gates or subcircuits to the circuit macromodels. The other source is due to the use of overly simplified MOS transistor models for the submicron devices with various short-channel effects. To solve the first problem, ILLIADS uses a new circuit primitive that reduces the mapping error [14]. To solve the second problem, ILLIADS uses the RWQ modeling technique [9] for accurate submicron device modeling. To take into account the temperature effects, we have further developed the temperature-dependent RWQ device models. In the following sections, we will first review the fast-timing simulator ILLIADS and the general RWQ technique, followed by the enhanced temperature-dependent RWQ device models.

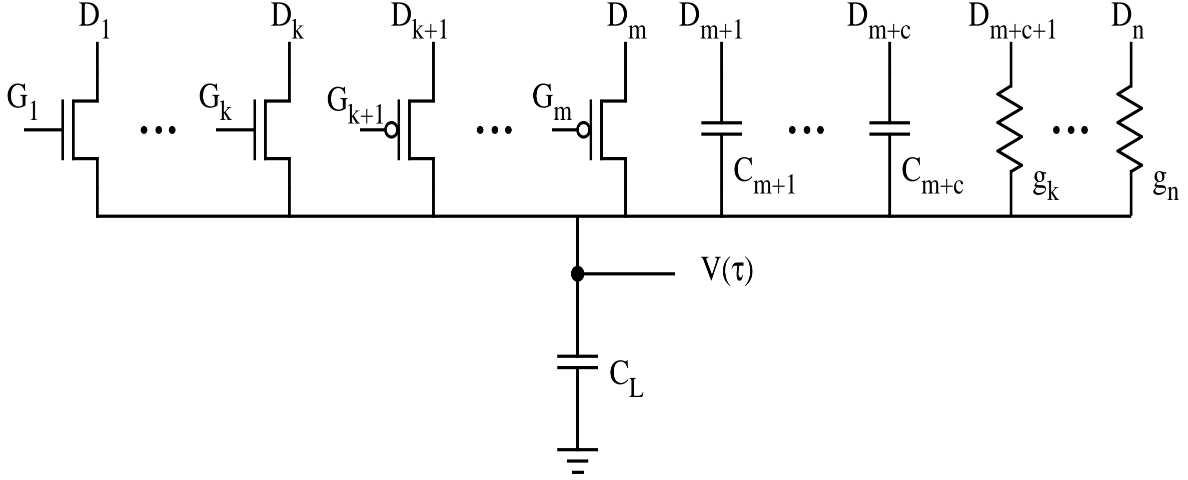


Figure 2.1 General MOS circuit primitive used in ILLIADS.

2.2 ILLIADS Fast-Timing Simulator

2.2.1 Primitive formation and solutions

The generic circuit primitive for MOS digital circuits is shown in Fig. 2.1. The primitive contains multiple branches of NMOS and PMOS transistors, linear coupling capacitances C_k, and linear conductances g_k. The output node voltage and the loading capacitance are denoted as $V(\tau)$ and C_L, respectively, and the applied terminal voltages are represented by D_k and G_k.

When the drain current i_k of MOS transistor k is modeled by a quadratic function of its terminal voltages, we have

$$i_k = f((V - D_k), (G_k - V - V_{T0}), (G_k - D_k - V_{T0})), \;\; 1 \leq k \leq m, \tag{2.1}$$

where V_{T0} is the threshold voltage. The well-known Shichman-Hodges model is a special case of (2.1). The current equations for the linear capacitors and conductors are given

by

$$i_k = C_k \frac{d(D_k - V)}{d\tau}, \quad m < k \leq m + c, \tag{2.2}$$

and

$$i_k = g_k(D_k - V), \quad m + c < k \leq n, \tag{2.3}$$

respectively, where c is the number of capacitors in the primitive and n is the total number of parallel branches in the primitive. When the waveforms of the applied terminal voltages D_k and G_k are linearized piecewise, we can write the state equation of the output node of the primitive in Fig. 2.1 as

$$C_L \frac{dV}{d\tau} = \sum_{\text{MOS}} i_k + \sum_{\text{CAP}} C_k \frac{d}{d\tau}(D_k - V) + \sum_{\text{COND}} g_k(D_k - V), \quad V(0) = V_0. \tag{2.4}$$

Substituting (2.1), (2.2) and (2.3) into (2.4), (2.4) can be rewritten as

$$\frac{dV}{d\tau} = kV^2 + (p_1\tau + p_0)V + (q_2\tau^2 + q_1\tau + q_0), \quad V(0) = V_0. \tag{2.5}$$

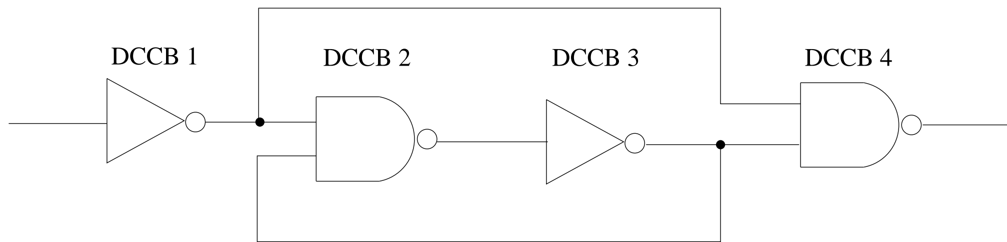
The coefficients k, p_1, p_0, q_2, q_1, and q_0 can be written in terms of the MOS transistor model parameters, capacitances, conductances, and input signals. The details can be found in [15].

Equation (2.5) belongs to the class of Riccati differential equations (RDE) [16]. In the most general case, the analytical solutions for the RDE can be found by using the hypergeometric functions [17]. An alternative approach is to use the power series method as done in ILLIADS. Detailed solutions of (2.5) and its degenerate forms are presented in [15].

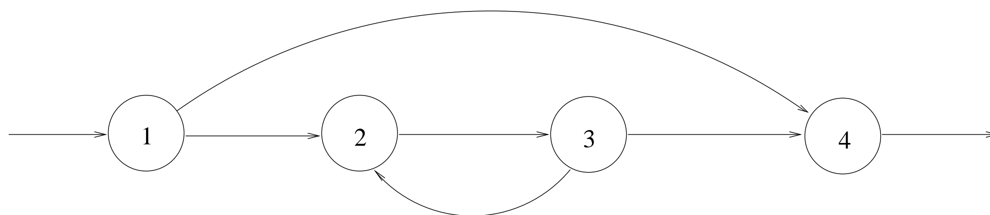
2.2.2 Simulation strategies

Given a circuit netlist, ILLIADS first partitions the circuit and groups the dc-connected blocks (DCCBs) using the breadth-first or depth-first search. A DCCB consists of a set of circuit nodes and elements that are connected through dc paths. Next, a directed graph (digraph) is constructed based on the connectivity of each DCCB. In order to detect the feedback loops, the vertices of the digraph are further partitioned into strongly connected components (SCCs). Each SCC contains DCCBs which can traverse to one another in the digraph. After SCC partitioning, the circuit graph consists of *condensed* vertices which are either DCCBs or SCCs. The topological sort is then performed to obtain their temporal order. The SCC partitioning and the topological sort can be done simultaneously using the modified Tarjan's algorithm [18]. The procedures for SCC formation and topological sort are illustrated by the example in Fig. 2.2.

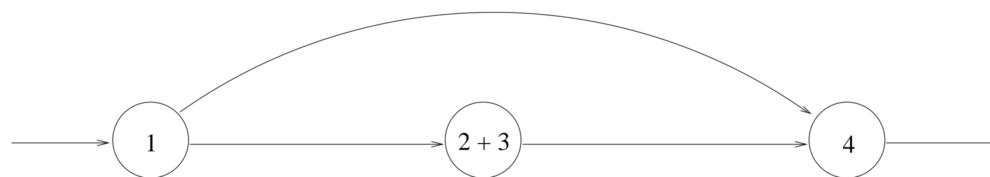
In ILLIADS, internal nodes (nodes that connect the dc paths of only one type of transistor, either NMOS or PMOS) are usually eliminated by merging serial or parallel transistors and forming the equivalent transistor. When necessary, internal nodes can be simulated with the trade-off of execution time. For serial merging of two transistors with transconductances β_1 and β_2, the transconductance of the equivalent transistor is given by $\beta_{eq} = \beta_1\beta_2/(\beta_1 + \beta_2)$. The equivalent gate signal is taken to be the weaker segment of the two gate signals (i.e., lower voltage for NMOS transistors and higher voltage for PMOS transistors). For parallel transistor merging, the equivalent transconductance is given by $\beta_{eq} = \beta_1 + \beta_2$, and the equivalent gate signal is the stronger segment of the two gate signals. Figure 2.3 illustrates the transistor merging and the internal node elimination process.



(a)



(b)



(c)

Figure 2.2 Illustrations of SCC formation and topological sort: (a) the original circuit, (b) the digraph representation, and (c) the condensed digraph after topological sort.

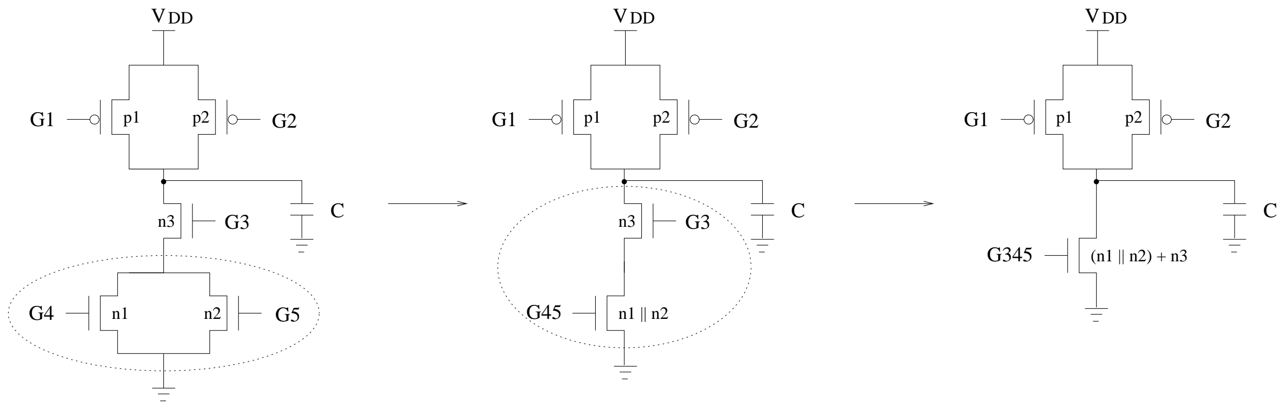


Figure 2.3 Example of transistor merging and internal node elimination.

After internal node elimination, the remaining circuit is mapped into a primitive with the generic structure shown in Fig. 2.1. For example, the circuit after node elimination in Fig. 2.3 is mapped into the primitive given in Fig. 2.4. Next, the state equation is formed and the corresponding RDE is solved analytically. The analogous output waveform can thus be obtained and it serves as the input to the next DCCB after piecewise linearization. For the circuit containing SCCs, the DCCBs inside the SCC are first ordered by a greedy algorithm where DCCBs with the most external inputs and the least feedbacks are put in front of the queue. The waveform-relaxation method [19] is used to simulate SCC. The waveform-relaxation algorithm implemented in ILLIADS uses the partial waveform and time convergence and the dynamic windowing technique [14].

Note that the main factor for ILLIADS simulation efficiency is to solve the RDE (2.5) analytically. To formulate this equation, the drain current of the MOS transistor is assumed to follow the quadratic dependency on it terminal voltages, as in the well-known

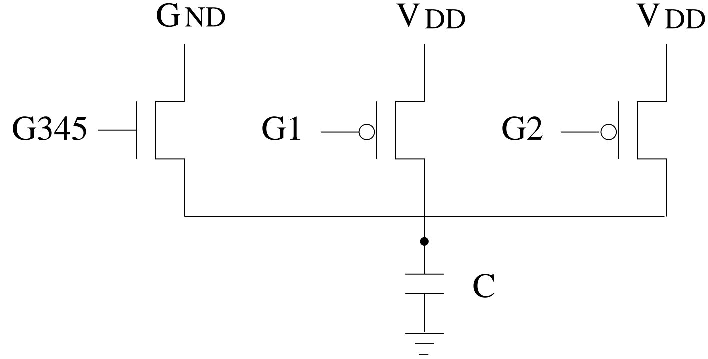


Figure 2.4 Primitive mapping for the circuit shown in Fig. 2.3 after the transistor merging process.

Shichman-Hodges model. However, this model is not accurate for submicron devices. Therefore, it becomes imperative to develop a new quadratic-current modeling technique to improve the simulation accuracy in ILLIADS. This technique, called RWQ technique [9], will be reviewed in Section 2.3.

2.2.3 Power estimation using ILLIADS

Given an input pattern, we calculate the average power for each DCCB using ILLIADS. Each DCCB is then treated as a heat source in thermal simulation. The procedure for our power calculation is shown in Fig. 2.5. It is similar to the power-meter method proposed in [20]. However, instead of building the power-meter circuitry, ILLIADS directly solves the RDE for each DCCB primitive and finds the current waveform drawn from the power supply for all branches that are connected to it. Like the power-meter method, ILLIADS can accurately calculate both *dynamic* and *short-circuit* power.

begin

```
for (all DCCBs in the circuit) {
    for (all simulated nodes in the DCCB) {
        find the voltage waveform by solving the Riccati
        equation from the state equation of the node;
    }
    for (all elements in DCCB connected to VDD)
        find the current waveform drawn from VDD;
    I_avg ← average the current waveform over the simulation period;
    P_avg ← I_avg × VDD;
}
```

end

Figure 2.5 DCCB power calculation using ILLIADS.

2.3 Regionwise Quadratic (RWQ) Modeling

The RWQ modeling procedure takes as input a set of data points (V_{ds}, V_{gse}, I_{ds}) that have been obtained either from measured data of a test device or by exercising (using SPICE, for example) a particular analytical or empirical MOS I-V model, where $V_{gse} = V_{gs} - V_{T0}$ and V_{T0} is the zero-bias threshold voltage of a MOS device. Next, we optimally partition the (V_{ds}, V_{gse}) plane into a number of regions and numerically fit a quadratic model of I_{ds} in terms of V_{ds} and V_{gse} in each region using the data points in that region. One example of the partitioned (V_{ds}, V_{gse}) plane is shown in Fig. 2.6. For a given regionwise partition, the following quadratic model of I_{ds} is fitted to the data in

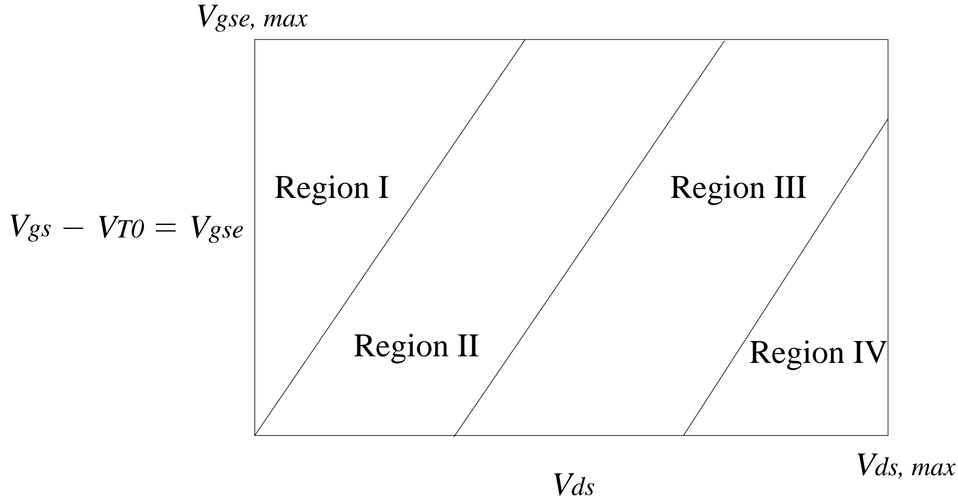


Figure 2.6 Regionwise partition of the (V_{ds}, V_{gse}) plane.

the kth region,

$$\frac{I_{ds}}{\beta} = (\alpha_0^{(k)} + \alpha_1^{(k)} V_{gse} + \alpha_2^{(k)} V_{ds} + \alpha_3^{(k)} V_{gse}^2 + \alpha_4^{(k)} V_{gse} V_{ds} + \alpha_5^{(k)} V_{ds}^2), \quad k = 1, \ldots, n_r, \quad (2.6)$$

where $\beta = \frac{1}{2}\mu_0 C_{ox} \frac{W}{L}$, n_r is the number of regions chosen for best fitting, and α's are fitting parameters in the kth region. In (2.6), there are two temperature-dependent physical parameters, μ_0 and V_{T0}. The temperature dependency of V_{T0} is relatively weak when compared with that of μ_0, and we adopt the temperature model of V_{T0} which is used in the SPICE Level 1 MOS model. For example, $V_{T0}(T)$ of an NMOSFET with an n^+-polysilicon gate is

$$V_{T0}^n(T) = -\frac{E_g}{2q} + \frac{kT}{q} \ln \frac{N_a}{n_i} - \frac{Q_{tot}}{C_{ox}} + \frac{2}{C_{ox}} \sqrt{\epsilon_{si} kT (\ln \frac{N_a}{n_i}) N_a}. \quad (2.7)$$

Similarly, $V_{T0}(T)$ of a PMOSFET with an n^+-polysilicon gate is

$$V_{T0}^p(T) = -\frac{E_g}{2q} - \frac{kT}{q} \ln \frac{N_d}{n_i} - \frac{Q_{tot}}{C_{ox}} - \frac{2}{C_{ox}} \sqrt{\epsilon_{si} kT (\ln \frac{N_d}{n_i}) N_d}, \quad (2.8)$$

where

$$n_i = 3.9 \times 10^{16} T^{1.5} e^{-(E_g/2kT)}, \quad \text{and} \tag{2.9}$$

$$E_g = 1.179 - 9.025 \times 10^{-5} T - 3.05 \times 10^{-7} T^2. \tag{2.10}$$

The total charge density Q_{tot} includes the surface state charge density (Q_{ss}), fixed charge density (Q_F), and the threshold voltage-adjustment implant density (Q_{imp}). The C_{ox} term is the oxide capacitance, E_g is the energy band gap of silicon, and ϵ_{si} is the permittivity of silicon.

2.3.1 Mobility modeling

The mobility μ_0 is a strong function of temperature, and two different $\mu_0(T)$ formulae have been implemented in our RWQ MOS device model. The first formula is the one commonly used in the SPICE model:

$$\mu_0(T) = \mu_0(300) \times (\frac{T}{300})^{-1.5}. \tag{2.11}$$

Because the on-chip temperature could be as high as 120°C and the device size keeps shrinking in state-of-the-art VLSI technologies, the simple mobility model in (2.11) may not be sufficient to cover a wide range of temperatures. However, it is rather difficult to devise an analytical formula to accurately calculate the channel mobility because of the complex quantum effects [21]. Although some empirical models such as SPICE BSIM3 use as many as eight fitting parameters to obtain a fairly good fit for the mobilities of a fixed technology, the technology dependency and scaling properties are not well-understood due to very little physical insight. Owing to above reasons, we have developed a physically based, semiempirical mobility model for the temperature range of 300 - 400 K, which is

the normal temperature range for most circuits. Because this model is for use in the fast-timing simulator, it must be accurate yet simple. In addition, this model should be scaled only with temperature and not with the transverse electric field E_{eff}, although the physical channel mobility actually depends on both temperature and transverse electric field. This is because the transverse field dependency is already taken into account by V_{gse} in (2.6) at the RWQ fitting stage, i.e., $E_{eff} \propto V_{gse}$ [22].

The carrier mobility is related directly to the mean-free time between collisions, which in turn is determined by the various scattering mechanisms. The three most important mechanisms are Coulomb, lattice, and surface-roughness scatterings.

2.3.1.1 Coulomb (impurity) scattering

Coulomb scattering results when a charge carrier travels past an ionized impurity. The effect of Coulomb scattering at high temperature is small because the carriers are moving faster and, therefore, scatter less. However, this cannot be neglected because the oxide charges contribute to the scattering at room temperature or higher [23]. In [23], it is proven that the Coulomb-scattering-limited mobility μ_C follows

$$\mu_C \propto T/N_I, \tag{2.12}$$

where T is temperature and N_I is the charge density at the Si-SiO_2 interface.

2.3.1.2 Lattice (phonon) scattering

Lattice scattering results from thermal vibrations of the lattice atoms at any temperature above zero. These vibrations disturb lattice periodic potential and allow energy to be transferred between the carriers and the lattice. For intermediate inversion-layer

concentrations ($Q_N/q = 0.5 \sim 5 \times 10^{12}/cm^2$), the channel mobility has been observed to have the following relationship with E_{eff} and T [24]:

$$\mu_L \propto T^{-n} E_{eff}^{-1/\gamma}, \tag{2.13}$$

where

$$E_{eff} = \frac{(0.5 \cdot Q_N + Q_D)}{\epsilon_{si}}. \tag{2.14}$$

The Q_D term is the depletion charge density, $\gamma = 3 \sim 6$, and $n = 1 \sim 1.5$, depending on the crystallographic orientation and the strength of intervalley and intersubband scattering.

2.3.1.3 Surface-roughness scattering

Surface-roughness scattering results from the asperities at the Si-SiO_2 interface at high electron concentrations. The dependence of the surface-roughness scattering-limited mobility μ_{SR} on E_{eff} is given by [25]

$$\mu_{SR} \propto E_{eff}^{-2}. \tag{2.15}$$

Because the probability of a collision ($1/\tau_c$) taking place in unit time is the sum of the probabilities of collisions due to various scattering mechanisms, we have

$$\frac{1}{\tau_c} = \frac{1}{\tau_{c,C}} + \frac{1}{\tau_{c,L}} + \frac{1}{\tau_{c,SR}},$$

or equivalently (Mathiessen's rule):

$$\frac{1}{\mu_n} = \frac{1}{\mu_C} + \frac{1}{\mu_L} + \frac{1}{\mu_{SR}}. \tag{2.16}$$

2.3.2 Proposed temperature-dependent mobility model

Based on (2.12), (2.13), (2.15) and (2.16), we propose the following temperature-dependent mobility model:

$$\begin{aligned} U(T) &= \mu_0^{-1}(T) \\ &= a_1 T^{-1} + a_2 T^{a_3} + a_4 \\ &= A_1[(\frac{T}{300})^{-1} - 1] + A_2[(\frac{T}{300})^{A_3} - 1] + A_4, \end{aligned} \tag{2.17}$$

where $U(T)$ is defined as the inverse of the mobility $\mu_0(T)$ for convenience, and A_1, A_2, A_3 and A_4 are the fitting parameters which will be determined by using the nonlinear least-square fitting to the extracted $\mu_0(T)$. Note that the dependencies on E_{eff} in (2.13) and (2.15) are merged into A_2 and A_4 in (2.17).

2.3.3 $V_{T0}(T)$ and $\mu_0(T)$ extraction

In this section, the methods used to extract $V_T(T)$ and $\mu_0(T)$ from the experimental I_{ds}-V_{ds} and I_{ds}-V_{gs} data are presented. In the linear region of MOSFET I_{ds}-V_{ds} characteristics where V_{ds} is small,

$$I_{ds}(T) = \mu_0(T, E_{eff}) C_{ox} \frac{W}{L} (V_{gs} - V_{T0}(T)) V_{ds}, \tag{2.18}$$

where $E_{eff} \approx 0.5 \cdot Q_N/\epsilon_{si} = 0.5 \cdot C_{ox}(V_{gs} - V_{T0}(T))/\epsilon_{si}$ if $Q_N \gg Q_D$ in (2.14). In other words, $\mu_0(T, E_{eff})$ is approximately a function of T and $(V_{gs} - V_{T0}(T))$. In our experiment, V_{ds} is chosen to be 5 mV. The gate voltage at which $I_{ds}(T)$ in (2.18) is zero is found by extrapolating the tangent at the point of the inflection of the I_{ds}-V_{gs} curve to zero current. The threshold voltage $V_{T0}(T)$, according to (2.18), is then equal to this

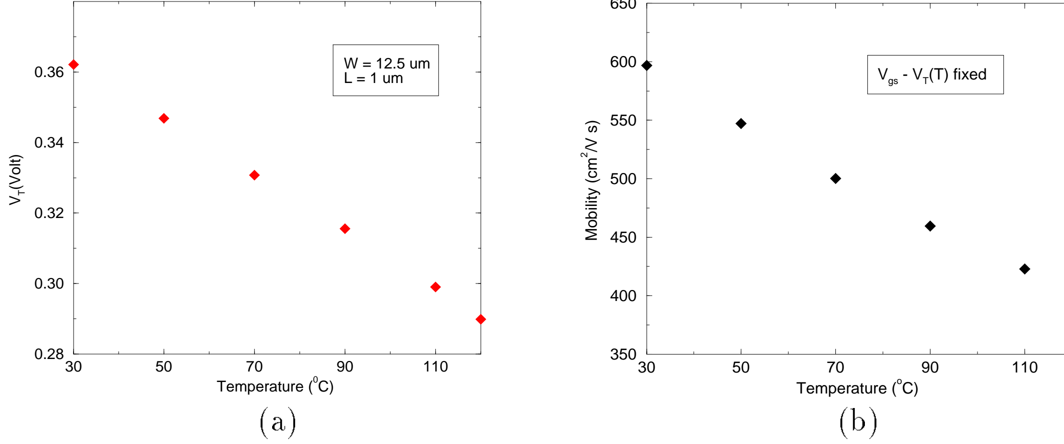


Figure 2.7 (a) Extracted $V_{T0}(T)$, and (b) Extracted $\mu_0(T)$.

V_{gs}. The extracted $V_{T0}(T)$ values of a sample device from $T = 30°\mathrm{C}$ to $T = 120°\mathrm{C}$ are shown in Fig. 2.7(a).

In our mobility study, two methods are used to extract $\mu_0(T)$. One is to directly use (2.18) and calculate $\mu_0(T, E_{eff})$ using the measured $I_{ds}(T)$ data for different V_{gs} and T. This method allows two physical quantities, V_{gs} and T, to vary simultaneously. However, if we are interested in the mobility variation subject only to T, as in the case of RWQ modeling, we have to set $(V_{gs} - V_{T0}(T))$ constant during $\mu_0(T)$ extraction. Following (2.18), we further assume that the E_{eff} and T dependencies of μ_0 are functionally separable. In other words, $\mu_0(T, E_{eff})$ can be expressed as $\mu_0(T) \times \mu_0^{(1)}(E_{eff})$. Because for $Q_N \gg Q_D$, $E_{eff} \propto (V_{gs} - V_{T0}(T))$, if we keep $V_{gs} - V_{T0}(T)$ constant during the mobility extraction, then

$$\frac{I_{ds}(T)}{I_{ds}(300K)} = \frac{\mu_0(T)}{\mu_0(300K)} \cdot \frac{\mu_0^{(1)}(V_{gs} - V_{T0}(T))}{\mu_0^{(1)}(V_{gs} - V_{T0}(300K))} \cdot \frac{(V_{gs} - V_{T0}(T))}{(V_{gs} - V_{T0}(300K))}$$
$$= \frac{\mu_0(T)}{\mu_0(300K)}.$$

Therefore, we have

$$\mu_0(T) = \mu_0(300K) \cdot \frac{I_{ds}(T)}{I_{ds}(300K)}, \tag{2.19}$$

where $\mu_0(300K)$ is either the measured or the user-specified mobility at room temperature. The extracted $\mu_0(T)$ values for the same device are shown in Fig. 2.7(b).

2.3.4 Preliminary mobility and RWQ-fitting results

In our first example, SPICE is used as a surrogate of an experiment to generate the I_{ds}-V_{ds} data sets at different temperatures, and the RWQ modeling technique is applied to fit these data sets. The SPICE-model parameters and device dimensions are shown in Table 2.1. We fit the SPICE data at room temperature for both NMOS and PMOS devices, and the quality of fitted I_{ds} is shown in Fig. 2.8. The I_{ds}-V_{ds} curves at T = 100°C are constructed by using the RWQ-fitting parameters α_0 - α_5 obtained at room temperature and the value of $\mu_0(T = 100)$ calculated according to (2.11). The results are compared with SPICE data in Fig. 2.9. It can be seen from Fig. 2.9 that the I_{ds}-V_{ds} data generated by the RWQ model generally match the SPICE data, but that deviations occur in the high-V_{gs}/high-V_{ds} region.

The above modeling inaccuracy is also observable when the RWQ model is used to fit the real experimental data. For example, the measured I-V data of an NMOSFET with dimensions of $w = 12.5$ μm and $l = 1$ μm are used for the RWQ fitting. The $\mu_0(T)$ values are determined by the extraction procedure presented in Section 2.3.3. Nonlinear least-square fitting of the A_1, A_2, A_3, and A_4 parameters in (2.17) is accomplished using the Levenberg-Marquart algorithm [26]. The results are shown in Fig. 2.10 and the best fitting parameters are given in the inset. Note that we do not include the mobility

Table 2.1 SPICE-model parameters and device dimensions for generating I-V data.

Parameter Description	Parameter Symbol	Units	Level 3 NMOS	Level 3 PMOS
Length	L	μm	0.8	0.8
Width	W	μm	1.6	1.6
Oxide thickness	t_{ox}	$\mathring{A}$	173.0	173.0
Surface mobility	μ_0	$cm^2/V \cdot s$	614.9	170.6
Zero-bias threshold voltage	V_{T0}	V	0.7805	-0.8482
Substrate doping	N_b	cm^{-3}	4.365e16	2.597e16
Maximum drift velocity	$VMAX$	m/s	1.49e5	1.82e5
Saturation field factor	$KAPPA$	-	9.51e-2	3.22e-2

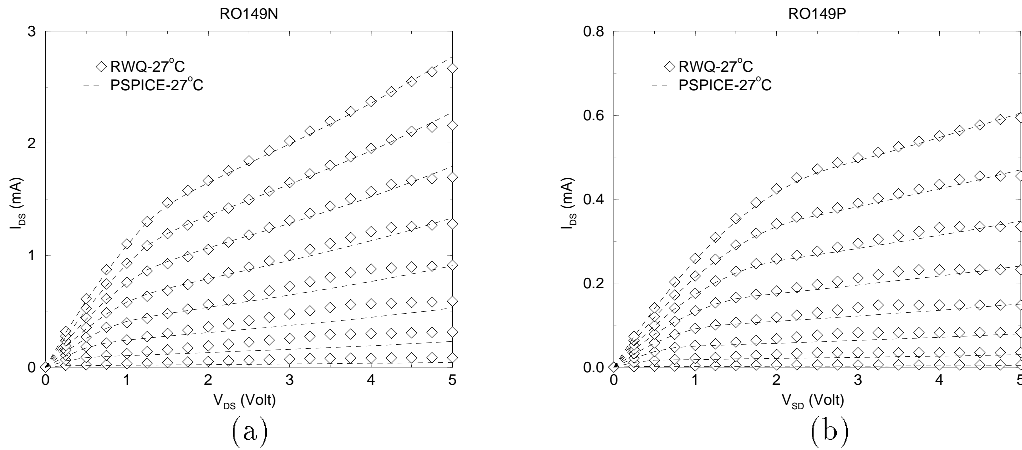


Figure 2.8 RWQ model fits for SPICE-generated data at 27°C: (a) NMOS and (b) PMOS.

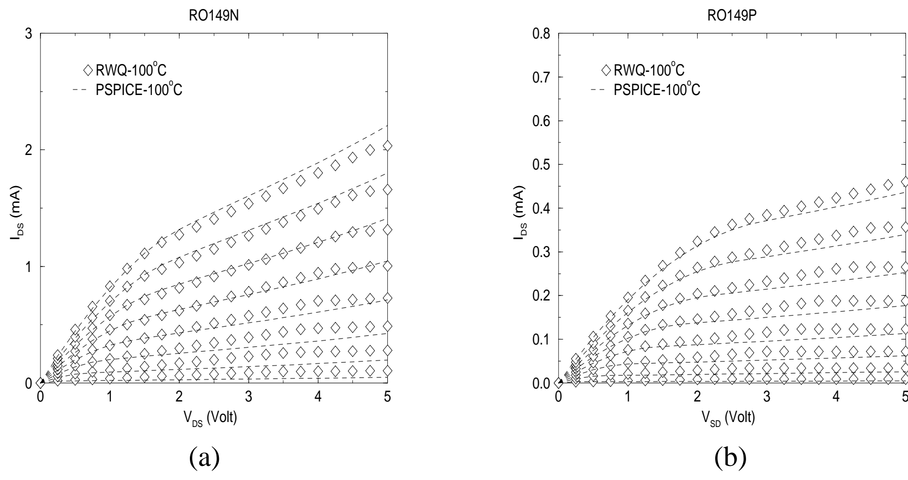


Figure 2.9 RWQ model fits for SPICE-generated data at 100°C: (a) NMOS and (b) PMOS.

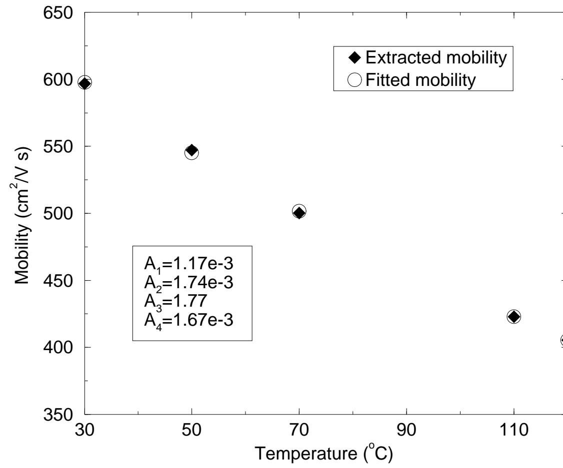


Figure 2.10 Fitted $\mu_0(T)$ using experimental data.

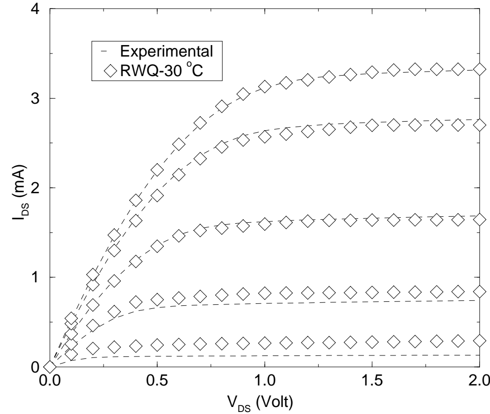


Figure 2.11 RWQ-fitting result for the experimental data at $T = 30°C$.

at 90°C in our fitting so that we can utilize this value to compare with the model-predicted mobility at 90°C. The RWQ model at room temperature is compared with measured data in Fig. 2.11. The mobility model in (2.17) is used to predict $\mu_0(T = 90)$ as $458.2\,cm^2/(V \cdot s)$, which is very close to $459.5\,cm^2/(V \cdot s)$ obtained by extraction. The I_{ds}-V_{ds} curves at $T = 90°C$ are constructed by using the RWQ-fitting parameters α_0 - α_5 obtained at room temperature and the value of $\mu_0(T = 90)$. The results are compared with experimental data in Fig. 2.12.

From Fig. 2.12, we can see that the RWQ model fits experimental data very well in the *linear* region. However, deviation is observed in the saturation region, which is similar to the case in Fig. 2.9. The reason for the deviation is that the RWQ model takes into account the mobility as a proportional constant. In other words, the RWQ model can capture the mobility scaling effect only when a linear relationship exists between I_{ds} and $\mu_0(T)$, as in the linear region of MOS I-V characteristics. This also implies that the physically extracted mobility may *not* be adequate for our purpose. In the following

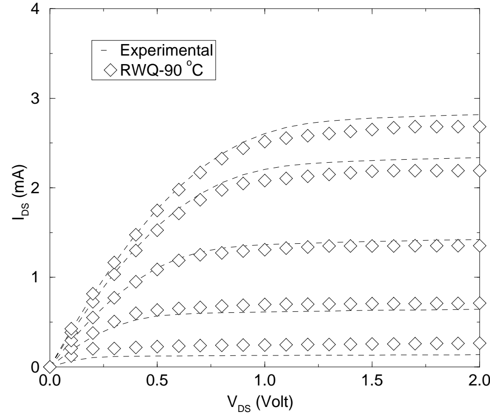


Figure 2.12 RWQ-fitting result for the experimental data at $T = 90°\text{C}$.

sections, we will introduce two different temperature-dependent RWQ device models which we have developed to accurately fit the given device I-V data for *all* regions. We denote these two models as RWQ Model I and RWQ Model II, respectively. RWQ Model I uses the physical mobility $\mu_0(T)$, and multiple sets of α_0 - α_5 must be provided for this model at certain temperatures. RWQ Model II uses the optimized mobility $\mu_0(T)$, and only one set of α_0 - α_5 needs to be provided.

2.3.5 RWQ Model I

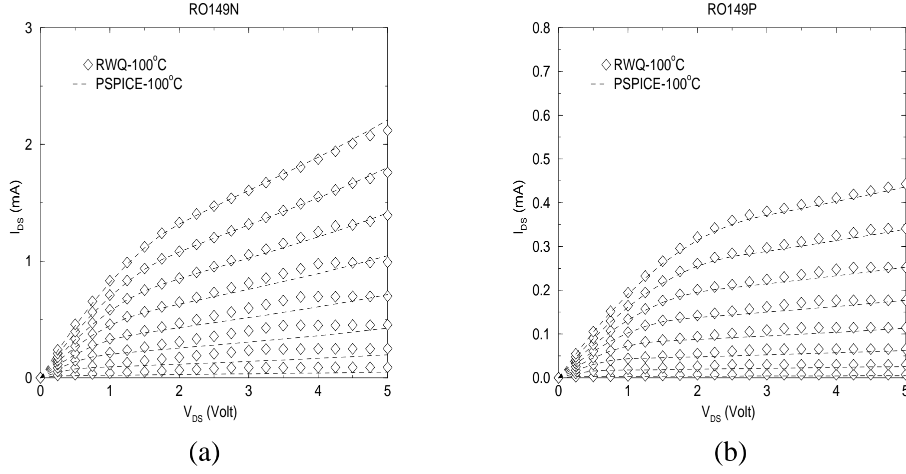
Suppose we RWQ-fit a set of experimental data at T and obtain the corresponding α_0 - α_5; denote this set of α's as RWQ(T). To use RWQ Model I, we assume that: (1) $\mu_0(T)$ and $V_{T0}(T)$ need to be recalculated whenever local temperature T is updated, and (2) the regionwise partition of the RWQ model and the corresponding RWQ(T) in each region remain unchanged for *certain intervals* of temperatures. Users need to fit the experimental I_d-V_{ds} data at three (or any user-specified number) different temperatures

(T_1, T_2, T_3). Thus, RWQ(T_1), RWQ(T_2) and RWQ(T_3) are obtained. If the local temperature T of a device satisfies $T_1 < T < T_2$ during the electrothermal simulation, $\mu_0(T)$ and $V_{T0}(T)$, as well as RWQ(T_1), are used in (2.6). Similarly, $\mu_0(T)$, $V_{T0}(T)$ and RWQ(T_2) are used for $T_2 < T < T_3$, with $\mu_0(T)$, $V_{T0}(T)$ and RWQ(T_3) being used for $T > T_3$.

To demonstrate the accuracy of the above approach, we RWQ-fit the I_d-V_{ds} data generated by SPICE at $T_1 = 27°$C, $T_2 = 77°$C, and $T_3 = 127°$C. Thus, RWQ(27), RWQ(77) and RWQ(127) are obtained. The device dimensions and SPICE-model parameters are given in Table 2.1 and the mobility formula in (2.11) is used. The resulting I_{ds}-V_{ds} characteristics at $T = 100°$C are shown in Fig 2.13, where good agreement with SPICE data is observed. In RWQ Model I, we implicitly take into account V_{T0}'s degree of freedom, which is originally suppressed in the RWQ-fitting procedure, while retaining the simplicity of (2.6) without introducing extra fitting parameters. When simulating VLSI chips, we measure and fit the device I-V curves under a few operating temperatures and generate the corresponding RWQ(T)'s. Appropriate RWQ(T) will be called during ILLIADS-T simulation and highly accurate temperature-dependent power and delay values can be estimated.

2.3.6 RWQ Model II

In this section, we use the experimental data of the NMOS device introduced in Section 2.3.4 for the temperature-dependent RWQ Model II fitting. Let $I_{ds}^{(k)}(\mathbf{x})$ and $\mu_0(T)f^{(k)}(\mathbf{x})$ be the experimental and RWQ-fitted drain currents in the k-th region, respectively. The vector $\mathbf{x}$ stands for the data point (V_{ds}, V_{gse}). Instead of using the physically extracted mobility, now we extract the optimized mobility by minimizing the



(a) (b)

Figure 2.13 I-V characteristics at 100°C using RWQ(77).

objective function

$$\sum_{k=1}^{n_r}\sum_{i=1}^{N_k}(I_{ds}^{(k)}(\mathbf{x_i}) - \mu_0(T)f^{(k)}(\mathbf{x_i}))^2, \tag{2.20}$$

where n_r is the number of regions and N_k is the number of data points in region k. This provides us with a best fit $\mu_0(T)f^{(k)}$ from which $\mu_0(T)$ is extracted as follows:

$$\mu_0(T) = \frac{\sum_{k=1}^{n_r}\sum_{i=1}^{N_k} I_{ds}^{(k)}(\mathbf{x_i})f^{(k)}(\mathbf{x_i})}{\sum_{k=1}^{n_r}\sum_{i=1}^{N_k}(f^{(k)}(\mathbf{x_i}))^2}. \tag{2.21}$$

Once $\mu_0(T)$ values are extracted at several temperatures, (2.17) is used to find the optimized set of A_1, A_2, A_3 and A_4. Based on this set of A_1 - A_4 and the RWQ-fitting parameters α_0 - α_5 obtained at room temperature, the RWQ-fitted I_{ds}-V_{ds} plot at $T = 90°$C is shown in Fig. 2.14. By using RWQ Model II, the fitting quality is generally good in all regions except for a small sacrifice in accuracy in the linear region as expected, i.e., the RWQ-fitted data overestimate I_{ds} in the linear region. To improve the linear region accuracy, which may be critical in timing simulations, we have further enhanced the RWQ Model II by using the regionwise-linear-interpolation scheme. This is a method to

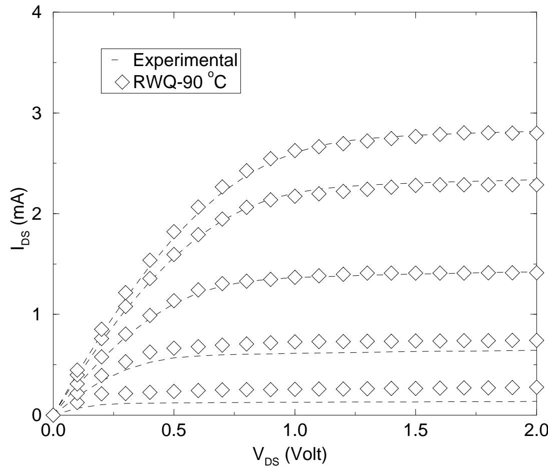


Figure 2.14 RWQ-fitting result at 90°C with mobility optimization.

optimize mobilities for each region and take into account the mobility continuity between regions by linear interpolation. This is graphically shown in Fig. 2.15(a). Assume that the V_{ds}-V_{gse} plane has been partitioned into three regions, and three mobility models have been generated according to (2.17). The dashed arrows in Fig. 2.15(a) cover the areas where mobility interpolations between adjacent regions are performed. To demonstrate the resulting accuracy by this approach, we plot the RWQ-generated I_{ds}-V_{ds} data at $T = 90°\text{C}$ in Fig. 2.15(b). It can be observed that this regionwise mobility model captures the temperature-dependent mobility scaling very well for all regions.

Finally, to demonstrate the simulation accuracy of ILLIADS-T in which the temperature-dependent RWQ Model II is implemented, we simulate a nine-stage inverter chain using both ILLIADS-T and SPICE. The BSIM3 MOSFET model is used in SPICE. The output waveforms at different temperatures are compared in Fig. 2.16.

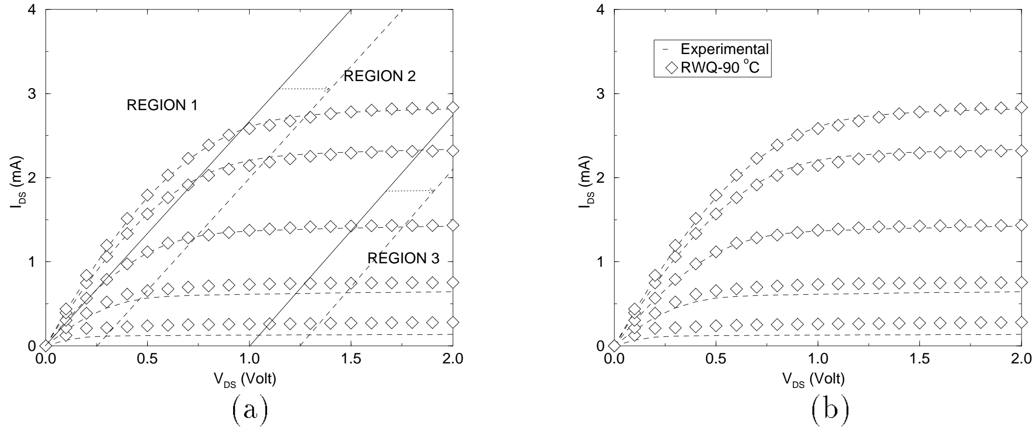


Figure 2.15 (a) Regionwise mobility fitting, and (b) fitting quality at 90°C.

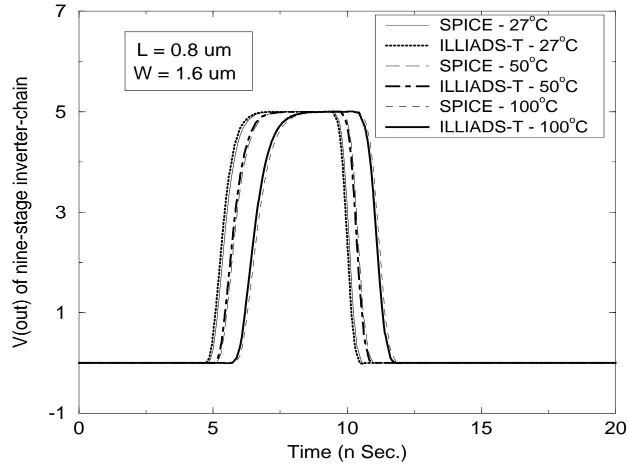


Figure 2.16 Output waveforms of a nine-stage inverter chain.

CHAPTER 3

THERMAL SIMULATION FRAMEWORK AND INCREMENTAL ELECTROTHERMAL SIMULATION

3.1 Introduction

The heat diffusion equation is the governing equation for heat conduction and temperature calculation. The general equation [27] is written as

$$\rho c_p \frac{\partial T(x,y,z,t)}{\partial t} = \nabla \cdot [k(x,y,z,T)\nabla T(x,y,z,t)] + g(x,y,z,t) \tag{3.1}$$

subject to the general thermal boundary condition (BC):

$$k(x,y,z,T)\frac{\partial T(x,y,z,t)}{\partial n_i} + h_i T(x,y,z,t) = f_i(x,y,z). \tag{3.2}$$

In (3.1) and (3.2), T is the temperature ($°C$), g is the power density of the heat source(s) (W/m^3), k is the thermal conductivity ($W/(m\,°\mathrm{C})$), ρ is the density of material (Kg/m^3), c_p is the specific heat ($J/(Kg\,°\mathrm{C})$), h_i is the heat transfer coefficient ($W/(m^2\,°\mathrm{C})$), $f_i(x,y,z)$ is an arbitrary function, and n_i is the outward direction normal to the surface i. For a steady-state case, the $\frac{\partial T}{\partial t}$ term is zero. The following three types of thermal BCs derived from (3.2) can be applied to the chip boundaries, depending on the packaging materials and the surrounding environment:

$$\text{Isothermal (Dirichlet) BC:} \quad T = f_i(x,y,z), \tag{3.3}$$

$$\text{Insulated (Neumann) BC:} \quad \frac{\partial T}{\partial n_i} = 0, \tag{3.4}$$

$$\text{Convective (Robin) BC:} \quad k_i \frac{\partial T}{\partial n_i} = h_i(T - T_a), \tag{3.5}$$

where T_a is the ambient temperature.

We have developed a thermal simulation framework, iTEMP, to solve the steady-state heat diffusion problem in the chip level. This framework is schematically shown in Fig. 3.1. iTEMP contains three parts: a fast thermal simulator, a numerical thermal simulator, and an analytical thermal simulator. The on-chip substrate temperature profile simulated by iTEMP can be used for interconnect temperature estimation, as shown in Fig. 3.1. The interconnect temperature plays an important role in the electromigration reliability diagnosis, as will be discussed in Chapter 5. The fast thermal simulator in iTEMP is designed to quickly identify the on-chip hot spots. It is extremely fast, but provides only a qualitative temperature description of the heat sources. The numerical thermal simulator is designed for the full-chip temperature profiling, while the analytical thermal simulator is used for accurately pinpointing the temperatures of the hot spots identified earlier. Both numerical and analytical thermal simulations take into account the packaging effect. The general description of our substrate and package simulation strategy will be given in Section 3.2, and the details of the package simulation will be presented in Section 3.4.

3.2 Substrate/Package Modeling

Traditional 1-D/2-D thermal simulation, as we mentioned earlier, cannot provide the required accuracy for VLSI chips with heat sink and other package structures. iTEMP,

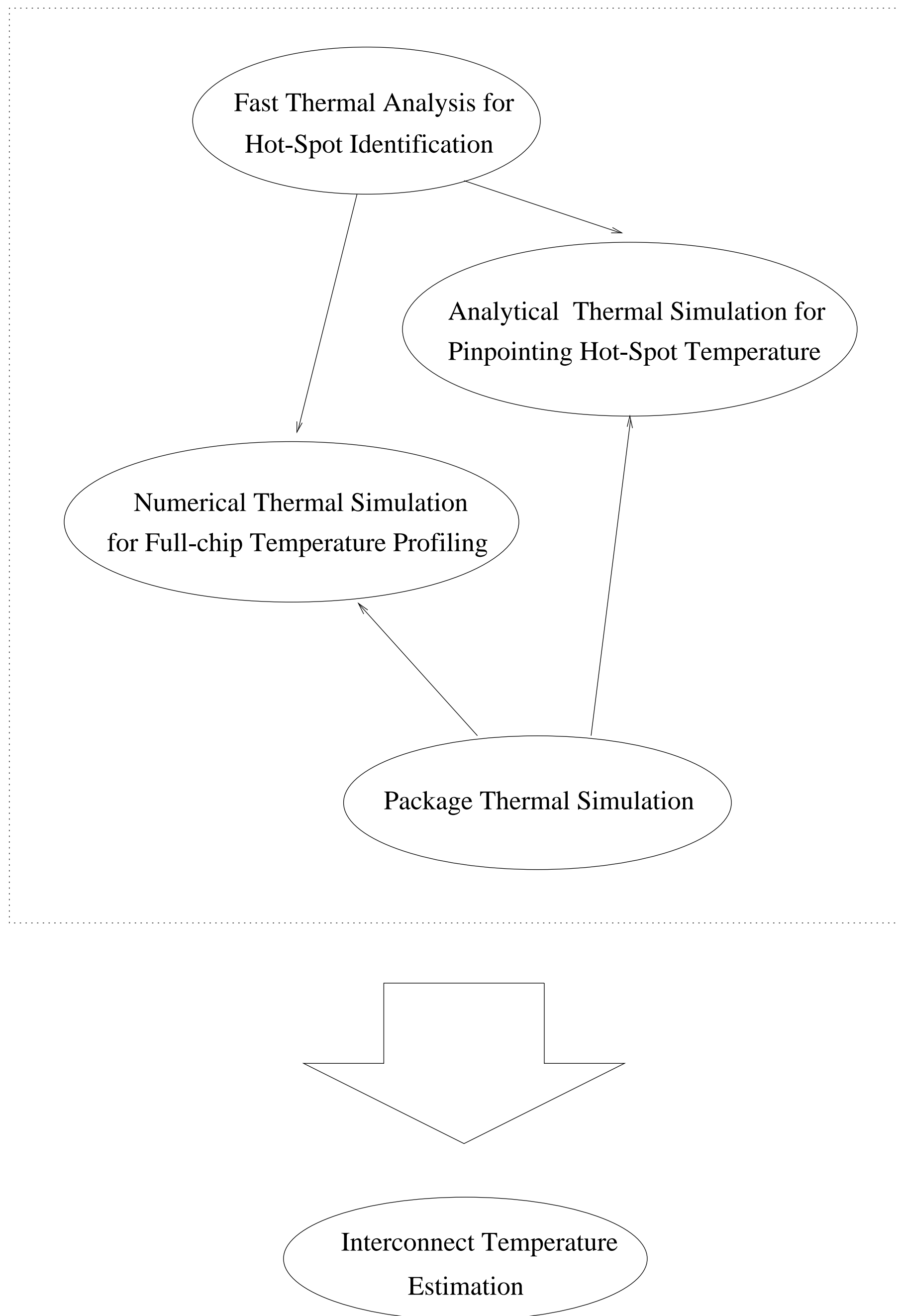


Figure 3.1 iTEMP thermal simulation framework.

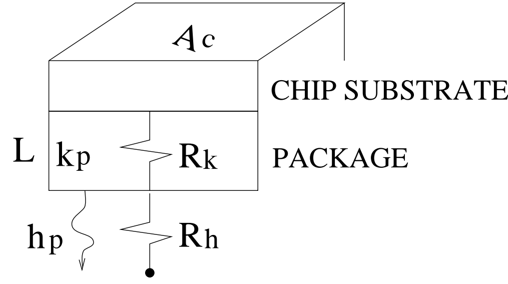


Figure 3.2 Illustration of effective heat transfer macromodeling.

however, simulates a chip by the following mixed 3-D/1-D strategies: (1) 3-D simulation is performed for the chip substrate to achieve a high degree of accuracy, and (2) packages and heat sinks are modeled as 1-D thermal resistances to reduce the computational complexity. We henceforth refer to strategy (2) as the *effective heat transfer macromodeling*. Specifically, as shown in Fig. 3.2, we serially combine the thermal resistance of the package or heat sink (R_k) with the one from package to ambience (R_h) to find the effective heat transfer coefficient h^e as given by

$$h^e = \frac{1}{A_c} \frac{1}{(R_h + R_k)}. \tag{3.6}$$

In (3.6), $R_k = \frac{L}{k_p A_c}$, $R_h = \frac{1}{h_p A_c}$, L is the thickness and k_p is the thermal conductivity of packages or heat sinks, h_p is the heat transfer coefficient from packages or heat sinks to ambience, and A_c is the chip area normal to the direction of heat flow. In other words, we merge the package and heat sink effects into the h_i term in (3.2) and form an effective h^e. The advantage of effective heat transfer macromodeling is threefold. First, it enhances the efficiency of iTEMP. Second, it removes the difficulty of analytically solving heat conduction problems of multilayered chip structures [28]. Third, it allows iTEMP to easily handle complicated chip structures, such as pins, by replacing k_p in

R_k with $k_{eff} = Xk_{pin} + (1 - X)k_p$, where k_{pin} is the thermal conductivity of pins and $X = \frac{(Area\, of\, pins)}{(Total\, package\, area)}$.

3.3 Formulation

By utilizing the different thermal simulation methods shown in Fig. 3.1, iTEMP can most efficiently generate the on-chip temperature profile, identify the hot spots, and pinpoint the hot-spot temperatures. In the following sections, the formulation of each thermal simulation technique in the iTEMP framework will be presented. The advantages and disadvantages of each technique will be also discussed.

3.3.1 Fast thermal analysis

Thermal simulation methods introduced in Section 3.1 offer different ways to find the on-chip temperature distribution. However, for a VLSI chip containing a large number of heat sources, the exact numerical and analytical methods may be very expensive. In the early chip design phase when no specific package information is given or the thermal BC is not fully characterized, a fast thermal analysis (FTA) method that emphasizes the hot-spot identification is extremely desirable. It can be used for the iterative temperature-sensitive module placement and routing in order to achieve a more uniform on-chip temperature distribution for better reliability and reducing excessive delays. We are thus motivated to develop a new FTA tool.

The FTA approach utilizes the fact that the dimensions of the gate-level or subcircuit-level heat sources in a VLSI chip are generally small when compared to the size of the chip. Therefore, all heat sources are assumed to be located in an *infinite* body. Consider

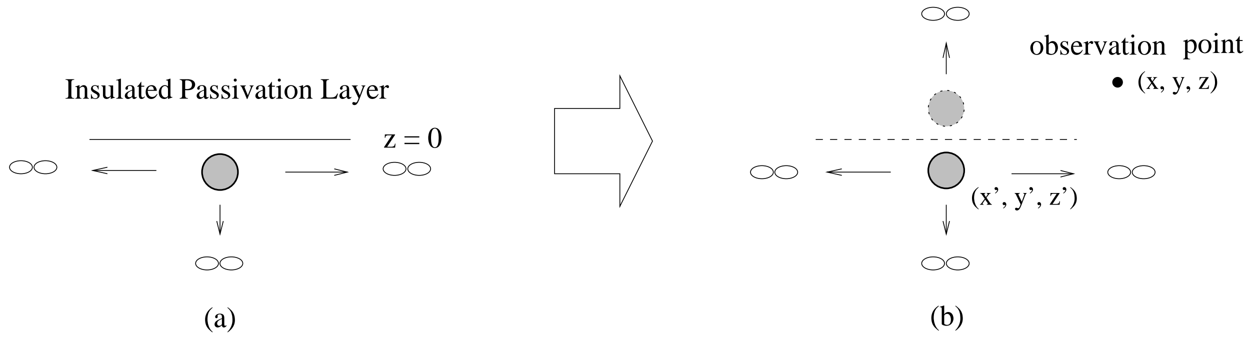


Figure 3.3 Method of images.

a point source in a chip as shown in Fig. 3.3(a). Because ICs have a passivation layer, the top of the chip is assumed to be insulated in this case. We thus have a boundary value problem with infinite dimension in the x-y plane with semi-infinite dimension in the z direction. Moreover, the BC at $z = 0$ is $\frac{\partial T(\mathbf{r},t)}{\partial z} = 0$. To find the temperature subject to this specific geometry and BCs, we use the *method of images* analogous to the electromagnetics problems [29]. We add an identical heat source that is symmetric with respect to $z = 0$ and remove the insulating boundary. Now, the problem in Fig. 3.3(a) is transformed to that in Fig. 3.3(b).

The Green's Function solution $G(\mathbf{r}, t|\mathbf{r}', \tau)$ to the heat diffusion equation for the point source in Fig. 3.3 can be derived as $G(\mathbf{r}, t|\mathbf{r}', \tau) = G_x \cdot G_y \cdot G_z$, where

$$\begin{aligned}
G_x &= \frac{1}{[4\pi\alpha(t-\tau)]^{1/2}} \exp[-\frac{(x-x')^2}{4\alpha(t-\tau)}], \\
G_y &= \frac{1}{[4\pi\alpha(t-\tau)]^{1/2}} \exp[-\frac{(y-y')^2}{4\alpha(t-\tau)}], \\
G_z &= \frac{1}{[4\pi\alpha(t-\tau)]^{1/2}} [\exp(-\frac{(z-z')^2}{4\alpha(t-\tau)}) + \exp(-\frac{(z+z')^2}{4\alpha(t-\tau)})], \qquad (3.7)
\end{aligned}$$

where α is the thermal diffusivity. We formulate the resulting temperature rise above the ambience at observation point $\mathbf{r}$ due to the parallelepiped heat source with dimensions

$a \times b \times c$ as

$$\triangle T(\mathbf{r},t) = \frac{\alpha P_0}{k(abc)} \int_{\tau=0}^{t} \int_{-c}^{c} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} G(\mathbf{r},t|\mathbf{r}',\tau) dv' d\tau, \quad (3.8)$$

where the coordinate origin has been set to be at the center of the source and P_0 is the source power (W). To proceed, we integrate (3.8) by using error functions:

$$\triangle T(x,y,0,t) = \frac{\alpha P_0}{k(abc)} \int_0^t G(x,a,\tau) \cdot G(y,b,\tau) \cdot G(0,c,\tau) d\tau, \quad (3.9)$$

where the observation point is set to be on the chip surface ($z = 0$), and

$$\begin{aligned} G(x,a,\tau) &= \frac{1}{2}[\text{erf}(\frac{a/2+x}{2\sqrt{\alpha(t-\tau)}}) + \text{erf}(\frac{a/2-x}{2\sqrt{\alpha(t-\tau)}})], \\ G(y,b,\tau) &= \frac{1}{2}[\text{erf}(\frac{b/2+y}{2\sqrt{\alpha(t-\tau)}}) + \text{erf}(\frac{b/2-y}{2\sqrt{\alpha(t-\tau)}})], \text{ and} \\ G(0,c,\tau) &= \text{erf}(\frac{c}{2\sqrt{\alpha(t-\tau)}}). \end{aligned} \quad (3.10)$$

If we define $\mathcal{A}_1 = 2(a/2+x)$, $\mathcal{A}_2 = 2(a/2-x)$, $\mathcal{B}_1 = 2(b/2+y)$, $\mathcal{B}_2 = 2(b/2-y)$, and $\mathcal{C} = 2c$, along with the change of variables, (3.9) can be rewritten as

$$\begin{aligned} \triangle T(x,y,0,t) = \; & \frac{\alpha P_0}{4k(abc)} \int_0^t [\text{erf}(\frac{\mathcal{A}_1}{4\sqrt{\alpha\tau}}) + \text{erf}(\frac{\mathcal{A}_2}{4\sqrt{\alpha\tau}})] \cdot [\text{erf}(\frac{\mathcal{B}_1}{4\sqrt{\alpha\tau}}) + \text{erf}(\frac{\mathcal{B}_2}{4\sqrt{\alpha\tau}})] \cdot \\ & \text{erf}(\frac{\mathcal{C}}{4\sqrt{\alpha\tau}}) d\tau. \end{aligned} \quad (3.11)$$

In order to perform the integration in (3.9) analytically, we piecewise linearize the error functions by (referring to Fig. 3.4) [8]

$$\begin{aligned} \text{erf}(x) &\approx 2x/\sqrt{\pi} && \text{for} \quad x \le \sqrt{\pi}/2; \\ &\approx 1 && \text{for} \quad x \ge \sqrt{\pi}/2. \end{aligned} \quad (3.12)$$

As a result, by defining $t_{a1} = \mathcal{A}_1^2/(4\pi\alpha)$, $t_{a2} = \mathcal{A}_2^2/(4\pi\alpha)$, $t_{b1} = \mathcal{B}_1^2/(4\pi\alpha)$, $t_{b2} = \mathcal{B}_2^2/(4\pi\alpha)$, and $t_c = \mathcal{C}^2/(4\pi\alpha)$, we have the approximations in Table 3.1, where $m = 0$ for $\mathcal{A}_2 > 0$

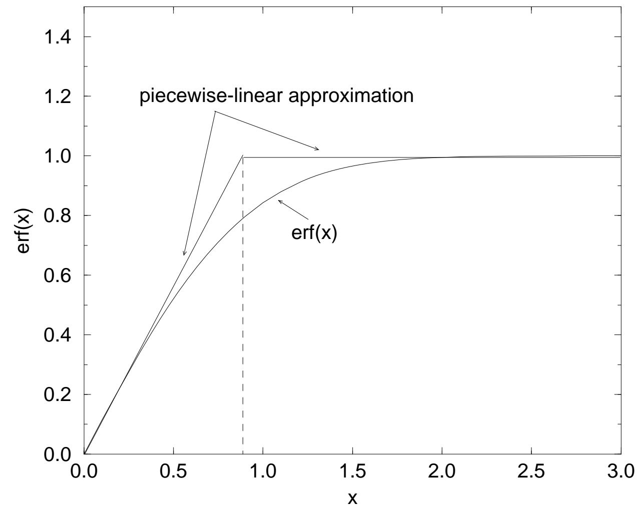


Figure 3.4 Error function approximation.

Table 3.1 Error function approximations.

$\mathrm{erf}(\frac{\mathcal{A}_1}{4\sqrt{\alpha\tau}})$	$\approx$	$(\frac{t_{a1}}{\tau})^{1/2}$	for	$\tau \geq t_{a1}$	$\mathrm{erf}(\frac{\mathcal{A}_2}{4\sqrt{\alpha\tau}})$	$\approx$	$(-1)^m(\frac{t_{a2}}{\tau})^{1/2}$	for	$\tau \geq t_{a2}$
	$\approx$	1	for	$\tau \leq t_{a1}$		$\approx$	$(-1)^m$	for	$\tau \leq t_{a2}$
$\mathrm{erf}(\frac{\mathcal{B}_1}{4\sqrt{\alpha\tau}})$	$\approx$	$(\frac{t_{b1}}{\tau})^{1/2}$	for	$\tau \geq t_{b1}$	$\mathrm{erf}(\frac{\mathcal{B}_2}{4\sqrt{\alpha\tau}})$	$\approx$	$(-1)^n(\frac{t_{b2}}{\tau})^{1/2}$	for	$\tau \geq t_{b2}$
	$\approx$	1	for	$\tau \leq t_{b1}$		$\approx$	$(-1)^n$	for	$\tau \leq t_{b2}$
$\mathrm{erf}(\frac{\mathcal{C}}{4\sqrt{\alpha\tau}})$	$\approx$	$(\frac{t_c}{\tau})^{1/2}$	for	$\tau \geq t_c$					
	$\approx$	1	for	$\tau \leq t_c$					

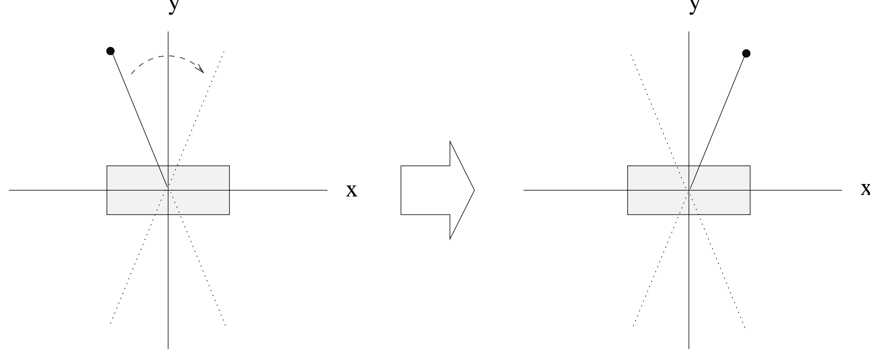


Figure 3.5 Transformation 1: Constrain the observation point to the first quadrant.

and $n = 0$ for $\mathcal{B}_2 > 0$; otherwise, $m = 1$ for $\mathcal{A}_2 < 0$ and $n = 1$ for $\mathcal{B}_2 < 0$. To obtain the analytical solution of (3.11), we need to reduce the number of possible permutations (i.e., 120) of t_{a1}, t_{a2}, t_{b1}, t_{b2} and t_c. To achieve this, we specify the following six constraints:

1. $x \geq 0$, $y \geq 0$
2. $(\frac{a}{2} + x) \geq |\frac{a}{2} - x| \Longrightarrow \mathcal{A}_1^2 \geq \mathcal{A}_2^2 \Longrightarrow t_{a1} \geq t_{a2}$
3. $(\frac{b}{2} + y) \geq |\frac{b}{2} - y| \Longrightarrow \mathcal{B}_1^2 \geq \mathcal{B}_2^2 \Longrightarrow t_{b1} \geq t_{b2}$
4. $2(\frac{a}{2} + x) \geq 2c \Longrightarrow \mathcal{A}_1^2 \geq \mathcal{C}^2 \Longrightarrow t_{a1} \geq t_c$
5. $2(\frac{b}{2} + y) \geq 2c \Longrightarrow \mathcal{B}_1^2 \geq \mathcal{C}^2 \Longrightarrow t_{b1} \geq t_c$
6. $t_{a1} \geq t_{b1}$

Constraints 2 and 3 are straightforward algebraically. Constraints 4 and 5 are valid because the thickness of heat sources is in the order of 0.1 μm, which is much smaller than the physical dimensions (i.e., $a \times b$) of a logic gate. Constraint 1 is equivalent to transforming all the observation points to the first quadrant by using the symmetric property, as graphically shown in Fig. 3.5. To satisfy Constraint 6, we use the coordinate

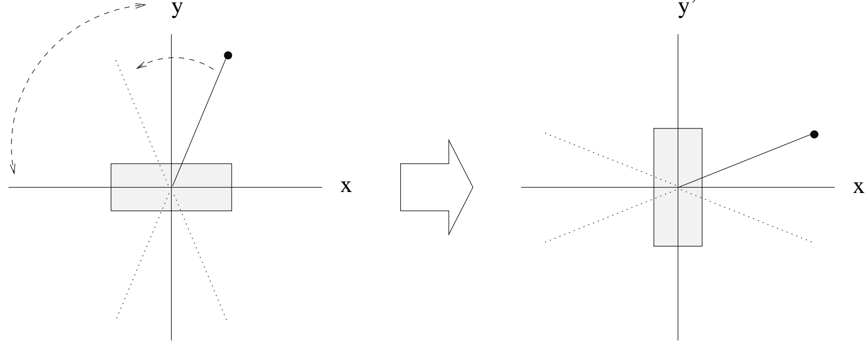


Figure 3.6 Transformation 2: Constrain t_{a1} to be larger than t_{b1}.

Case 1. $t_{a1} \geq t_{b1} \geq t_{b2} \geq t_{a2} \geq t_c$	Case 2. $t_{a1} \geq t_{b1} \geq t_{a2} \geq t_{b2} \geq t_c$
Case 3. $t_{a1} \geq t_{a2} \geq t_{b1} \geq t_{b2} \geq t_c$	Case 4. $t_{a1} \geq t_{a2} \geq t_{b1} \geq t_c \geq t_{b2}$
Case 5. $t_{a1} \geq t_{b1} \geq t_{a2} \geq t_c \geq t_{b2}$	Case 6. $t_{a1} \geq t_{b1} \geq t_{b2} \geq t_c \geq t_{a2}$
Case 7. $t_{a1} \geq t_{b1} \geq t_c \geq t_{b2} \geq t_{a2}$	Case 8. $t_{a1} \geq t_{b1} \geq t_c \geq t_{a2} \geq t_{b2}$

Figure 3.7 Eight cases under six constraints.

transformation as shown in Fig. 3.6. Although the observation point in Fig. 3.6 meets Constraint 1, we have to exchange x and y coordinates in order to force it to satisfy Constraint 6, again by using symmetry. With the above specified constraints, (3.11) now becomes tractable. In other words, the precedence of t_{a1}, t_{a2}, t_{b1}, t_{b2}, and t_c must belong to one of the eight cases, as shown in Fig. 3.7. We have derived the analytical solutions for all cases and the proper one will be used during simulations, depending on the geometry and the size of the heat source, as well as the relative locations between the heat source and the observation point.

For a VLSI chip with n heat sources, the temperature rise at the center of source i is obtained by considering the heat diffusion from i itself, plus that from other $n-1$ sources using superposition:

$$\triangle T_i^{\Sigma} = \triangle T_i(\mathbf{0}, t) + \sum_{k=1}^{n-1} \triangle T_k(\mathbf{r}_k, t), \tag{3.13}$$

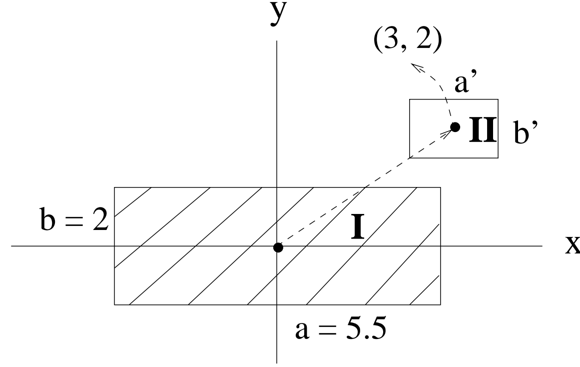


Figure 3.8 FTA example.

where $\triangle T_i^{\Sigma}$ is the temperature rise at the center of source i, $\triangle T_i(\mathbf{0}, t)$ is the temperature rise due to i itself, and $\triangle T_k(\mathbf{r}_k, t)$ is the temperature rise due to source k. The $\triangle T_i(\mathbf{0}, t)$ term and $\triangle T_k(\mathbf{r}_k, t)$ term can both be found by combining (3.11) with one of the eight cases in Fig. 3.7. Take Fig. 3.8 as an example, where one source with power P_I is located at (0,0) while the other one with power P_{II} is at (3,2). To find the temperature rise at the center of source II due to source I (i.e., $\triangle T_k(\mathbf{r}_k, t)$ in (3.13)), we observe that it belongs to Case 1, where $t_{a1} \geq t_{b1} \geq t_{b2} \geq t_{a2} \geq t_c$. Thus, the integration in (3.11) can be explicitly performed as

$$
\begin{aligned}
\triangle T_I(x, y, 0, t) = \frac{\alpha P_I}{4k(abc)} \{ & \int_0^{t_c} [1 + (-1)^m] \cdot [1 + (-1)^n] \cdot 1 \cdot d\tau \\
& + \int_{t_c}^{t_{a2}} [1 + (-1)^m] \cdot [1 + (-1)^n] \cdot (\frac{t_c}{\tau})^{1/2} \cdot d\tau \\
& + \int_{t_{a2}}^{t_{b2}} [1 + (-1)^m (\frac{t_{a2}}{\tau})^{1/2}] \cdot [1 + (-1)^n] \cdot (\frac{t_c}{\tau})^{1/2} \cdot d\tau \\
& + \int_{t_{b2}}^{t_{b1}} [1 + (-1)^m (\frac{t_{a2}}{\tau})^{1/2}] \cdot [1 + (-1)^n (\frac{t_{b2}}{\tau})^{1/2}] \cdot (\frac{t_c}{\tau})^{1/2} \cdot d\tau \\
& + \int_{t_{b1}}^{t_{a1}} [1 + (-1)^m (\frac{t_{a2}}{\tau})^{1/2}] \cdot [(\frac{t_{b1}}{\tau})^{1/2} + (-1)^n (\frac{t_{b2}}{\tau})^{1/2}] \cdot (\frac{t_c}{\tau})^{1/2} \cdot d\tau \\
& + \int_{t_{a1}}^{t} [(\frac{t_{a1}}{\tau})^{1/2} + (-1)^m (\frac{t_{a2}}{\tau})^{1/2}] \cdot [(\frac{t_{b1}}{\tau})^{1/2} + (-1)^n (\frac{t_{b2}}{\tau})^{1/2}] \cdot (\frac{t_c}{\tau})^{1/2} \cdot d\tau \}. \quad (3.14)
\end{aligned}
$$

Because $m = n = 1$ for the situation in Fig. 3.8, we have the following result for the steady state:

$$\begin{aligned}\triangle T_I(x, y, 0, \infty) &= \frac{\alpha P_I}{4k(abc)}[(\sqrt{t_{b1}t_c} - \sqrt{t_{b2}t_c}) \cdot (4 + 2\log(\frac{\mathcal{A}_1}{\mathcal{B}_1})) \\ &\quad -(\sqrt{t_{b2}t_c} + \sqrt{t_{a2}t_c}) \cdot 2\log(\frac{\mathcal{B}_1}{\mathcal{B}_2})] \text{ for } m = 1 \text{ and } n = 1, \end{aligned} \tag{3.15}$$

where $(x, y) = (3, 2)$. Similarly, the temperature rise at the center of source II caused by itself (i.e., $\triangle T_i(\mathbf{0}, t)$ in (3.13)) can be found because it is simply a special case ($t_{a1} = t_{a2} \geq t_{b1} = t_{b2} \geq t_c$) of Case 3. The result is

$$\triangle T_{II}(\mathbf{0}, \infty) = \frac{P_{II}}{\pi k a'}[2 + \log(\frac{a'}{b'}) - \frac{c}{b'}]. \tag{3.16}$$

Finally, the steady-state temperature rise at the center of source II can be obtained according to (3.13) as

$$\triangle T_{II}^{\Sigma} = \triangle T_{II}(\mathbf{0}, \infty) + \triangle T_I(3, 2, 0, \infty). \tag{3.17}$$

Mathematical formulation of the FTA method is based on the closed-form Green's function with the assumption of a semi-infinite boundary condition. This assumption does not practically hold due to the existence of package and heat sink in a chip. The temperature rise in (3.13), therefore, represents the relative value instead of the absolute and accurate temperature rise. However, the FTA method provides a quick qualitative estimate of the temperature distribution. This is particularly useful when the number of heat sources is large, or when a large number of repeated thermal simulations needs to be performed. In order to take into account the effects of package and heat sink, detailed thermal simulation using a numerical or analytical approach is needed.

3.3.2 Numerical approach

The numerical approach in iTEMP makes use of the 3-D finite-difference (FD) technique. Because the gate count in a VLSI chip is large, it is impractical to allocate one or more grids to each gate in the FD method. Instead, the grid number and spacings in iTEMP are determined by taking into account the chip size, the gate density, and the temperature field density. We employ an adaptive meshing technique to determine the grid spacing in iTEMP. First, iTEMP uniformly deploys the on-chip grids according to the user-specified initial grid number. After obtaining an initial estimate of the temperature distribution, iTEMP further refines or redistributes the grids by sensing the temperature gradient and adding extra grids in the regions with larger gradients based on the following *weight function* and *equidistribution* criteria [30]:

$$w(r) = \sqrt{1 + \alpha^2(\frac{\partial T}{\partial r})^2}, \text{ and} \tag{3.18}$$

$$\int_{r_i}^{r_{i+1}} w(r)dr = \text{Constant}, \tag{3.19}$$

where α is a user-specified parameter and r denotes x or y. Temperature solutions found using the current and previous grid systems are compared. If the percentage difference is less than a prescribed threshold, then the grid refinement process is terminated. The stopping criterion can also be the user-specified maximum number of grids. According to our empirical observation, tens of neighboring gates can be covered by a single grid rectangle given a 1% error bound. Only a few grids are placed in the z-direction (thickness) with most grids concentrated at the chip surface near heat sources. This is because the temperature drops rapidly away from the surface in the z-direction and larger grid sizes can be used.

A schematic representation of a part of a chip containing several heat sources and the variable grid system is shown in Fig. 3.9. After the coordinates of each heat source are identified, the corresponding grid points that the heat flows into, and the proportionate power values in the analogous thermal circuit are found as shown in Fig. 3.10. The heat flow coming from source i is denoted as P_i in Fig. 3.10. In Figs. 3.9(a) and 3.10(a), the solid lines represent the chosen grid lines and the dashed lines are in the middle of two adjacent grid lines, and A_{eff} is the effective area of a grid point. Every heat source that overlaps the effective area of a grid point serves as a power source feeding into that grid, and the corresponding power value is calculated based on the ratio of the source area within A_{eff} to the total area of the source. In Fig. 3.9(b), h_{x+}, h_{x-}, h_{y+}, h_{y-}, h_{z+} and h_{z-} are *halves* of the distances from grid (i, j, k) to grids $(i+1, j, k)$, $(i-1, j, k)$, $(i, j+1, k)$, $(i, j-1, k)$, $(i, j, k+1)$, and $(i, j, k-1)$, respectively. The thermal conductances G_1, G_2 and G_3 in Fig. 3.10(b) can be found by applying the first law of thermodynamics on the grid point (i, j, k):

$$
\begin{aligned}
& k(h_{y+} + h_{y-})(h_{z+} + h_{z-})\frac{T_{i+1,j,k} - T_{i,j,k}}{2h_{x+}} + k(h_{y+} + h_{y-})(h_{z+} + h_{z-})\frac{T_{i-1,j,k} - T_{i,j,k}}{2h_{x-}} \\
& +k(h_{x+} + h_{x-})(h_{z+} + h_{z-})\frac{T_{i,j+1,k} - T_{i,j,k}}{2h_{y+}} + k(h_{x+} + h_{x-})(h_{z+} + h_{z-})\frac{T_{i,j-1,k} - T_{i,j,k}}{2h_{y-}} \\
& +k(h_{x+} + h_{x-})(h_{y+} + h_{y-})\frac{T_{i,j,k+1} - T_{i,j,k}}{2h_{z+}} + k(h_{x+} + h_{x-})(h_{y+} + h_{y-})\frac{T_{i,j,k-1} - T_{i,j,k}}{2h_{z-}} \\
& = 0.
\end{aligned}
\tag{3.20}
$$

From (3.20), we recognize that the heat conduction in a thermal circuit is similar to the current conduction in an electrical circuit with the analogy shown in Fig. 3.11. In other words, we can always transform a finite-difference heat conduction problem into an electrical RC network problem. The thermal conductances connected to grid (i, j, k)

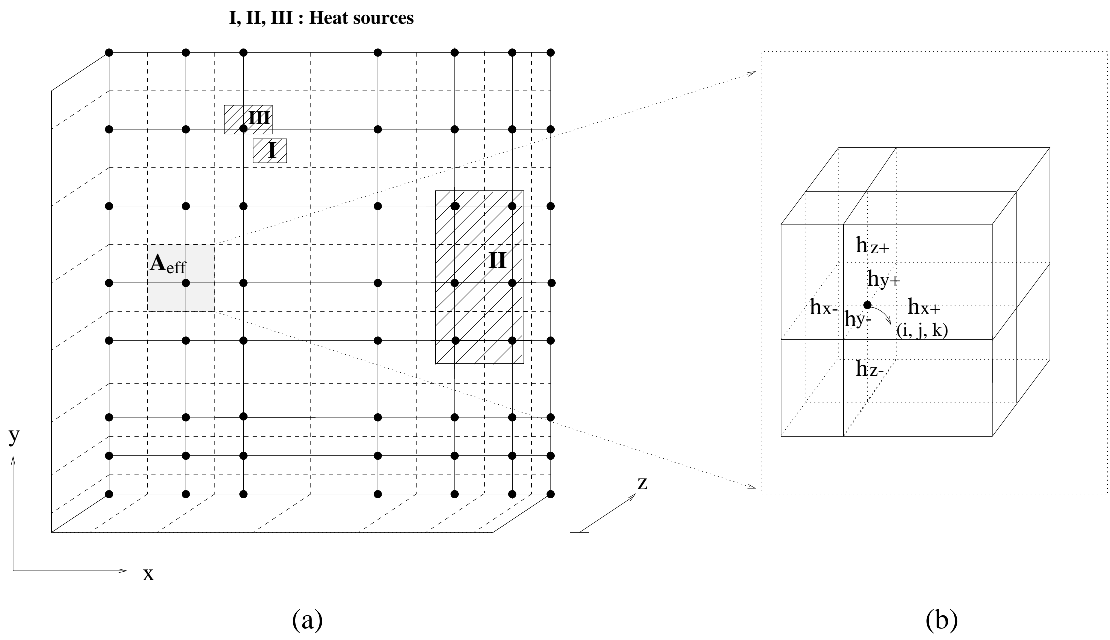


Figure 3.9 (a) Top view of a part of the chip containing heat sources, and (b) 3-D view of grid point (i, j, k).

(a) (b)

Figure 3.10 (a) Analogous thermal circuit to Fig. 3.9(a), and (b) thermal conductances from (i, j, k) to adjacent grids.

Figure 3.11 Analogy between thermal and electrical circuits.

can therefore be found as

$$G_1 = \frac{k \cdot (h_{y+} + h_{y-})(h_{z+} + h_{z-})}{2h_{x+}}, \tag{3.21}$$

$$G_2 = \frac{k \cdot (h_{x+} + h_{x-})(h_{z+} + h_{z-})}{2h_{y+}}, \text{ and} \tag{3.22}$$

$$G_3 = \frac{k \cdot (h_{x+} + h_{x-})(h_{y+} + h_{y-})}{2h_{z+}}, \tag{3.23}$$

where k is the thermal conductivity. Similar expressions for G_4, G_5 and G_6 can be derived. For a composite material system, as shown in Figs. 3.12 and 3.13, G_1, G_2, and G_3 are found as

$$G_1 = \frac{k1}{2h_{x+}}(h_{y+} + h_{y-})(h_{z+} + h_{z-}), \tag{3.24}$$

$$G_2 = \frac{k1 \cdot h_{x+} + k2 \cdot h_{x-}}{2h_{y+}}(h_{z+} + h_{z-}), \text{ and} \tag{3.25}$$

$$G_3 = \frac{k1 \cdot h_{x+} + k2 \cdot h_{x-}}{2h_{z+}}(h_{y+} + h_{y-}). \tag{3.26}$$

The resulting analogous electrical circuit is solved by either the sparse-matrix or the successive-over-relaxation (SOR) technique, and the on-chip temperatures are obtained. We compute the average temperature of each gate by averaging the temperature values of grids that a gate covers, and then use the updated values as the input to the fast-timing simulator for the next simulation run.

The above discussion was on the temperature calculation for interior grids. However, special care must be taken to model the chip boundaries subject to different BCs. Figure 3.14 illustrates the thermal circuit used to model the top of a chip with a convective BC, where T_a is the ambient temperature. To find the equivalent thermal resistances for this system, we again apply the first law of thermodynamics on the grid point (i, j, k):

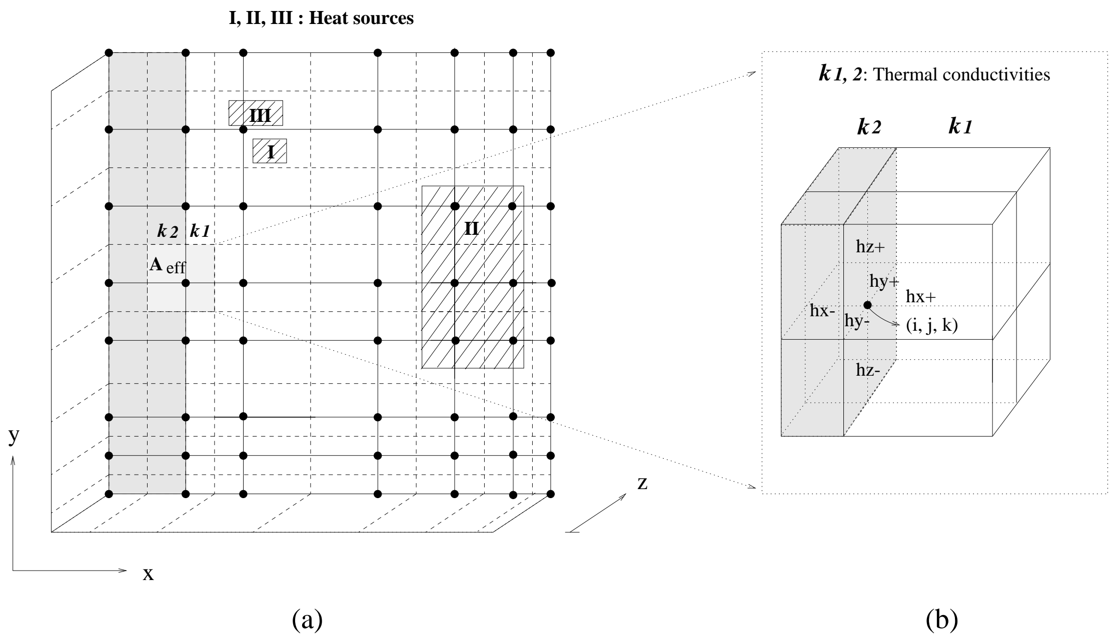


Figure 3.12 (a) Top view of a part of the chip comprised of composite materials, and (b) 3-D view of grid point (i, j, k).

$$\begin{aligned}
&k(h_{y+}+h_{y-})h_{z-}\frac{T_{i+1,j,k}-T_{i,j,k}}{2h_{x+}}+k(h_{y+}+h_{y-})h_{z-}\frac{T_{i-1,j,k}-T_{i,j,k}}{2h_{x-}}\\
&+k(h_{x+}+h_{x-})h_{z-}\frac{T_{i,j+1,k}-T_{i,j,k}}{2h_{y+}}+k(h_{x+}+h_{x-})h_{z-}\frac{T_{i,j-1,k}-T_{i,j,k}}{2h_{y-}}\\
&+k(h_{x+}+h_{x-})(h_{y+}+h_{y-})\frac{T_{i,j,k-1}-T_{i,j,k}}{2h_{z-}}+h^e(T_a-T_{i,j,k})(h_{x+}+h_{x-})(h_{y+}+h_{y-})\\
&=0, \qquad (3.27)
\end{aligned}$$

where h^e is the effective heat transfer coefficient. Using the analogy in Fig. 3.11, we recognize the thermal conductances G_1, G_2 and G_3 in Fig. 3.14(a) as

$$G_1 = \frac{kh_{z-}(h_{y+}+h_{y-})}{2h_{x+}}, \qquad (3.28)$$

$$G_2 = \frac{kh_{z-}(h_{x+}+h_{x-})}{2h_{y+}}, \text{ and} \qquad (3.29)$$

$$G_3 = \frac{k(h_{x+}+h_{x-})(h_{y+}+h_{y-})}{2h_{z-}}. \qquad (3.30)$$

Defining $A_{eff} = (h_{x+}+h_{x-})(h_{y+}+h_{y-})$ as the effective area of the grid (i, j, k), we obtain the thermal resistance related to the convective heat transfer (R_h^e in Fig. 3.14(a)) as

$$R_h^e = \frac{1}{h^e A_{eff}}. \qquad (3.31)$$

We solve this boundary value problem as follows. First, the circuit in Fig. 3.14(a) is transformed to the equivalent circuit in Fig. 3.14(b), and a 3-D network containing only resistive elements and independent current sources is obtained (capacitive elements are open-circuited in the steady state). Next, a nodal analysis of this network is performed and the admittance matrix is constructed. Finally, the admittance matrix is solved and the temperature of each node is found. For BCs other than the convective condition, a similar procedure follows after replacing h^e in (3.31) with ∞ for the isothermal condition or 0 for the insulated condition.

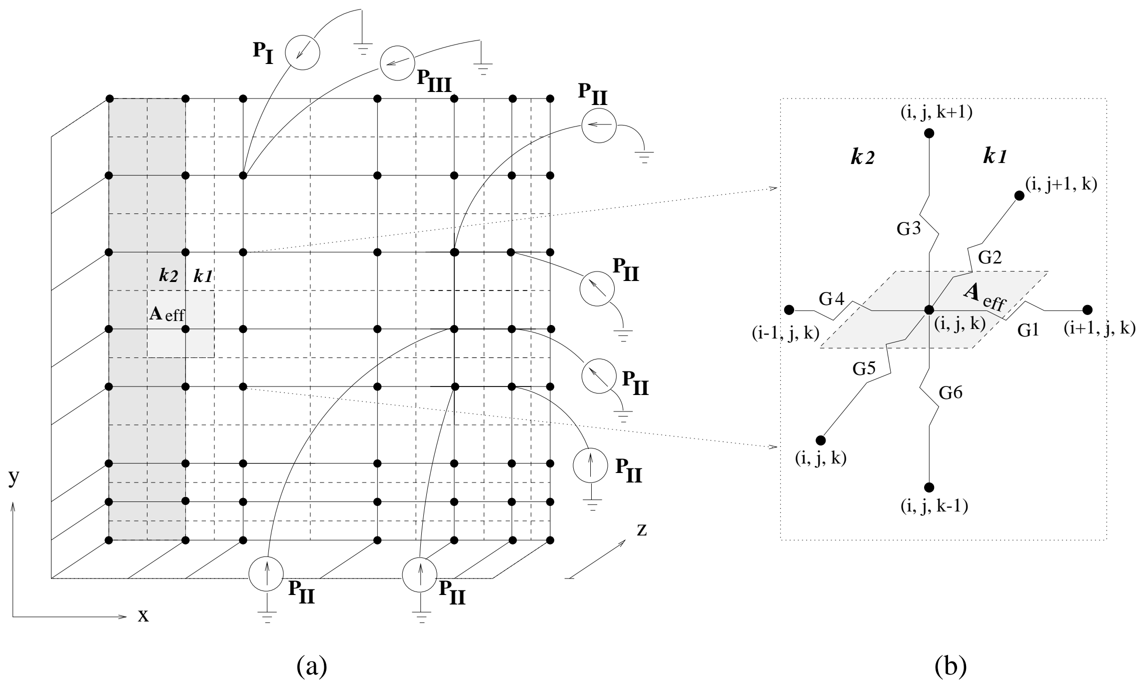


Figure 3.13 (a) Analogous thermal circuit to Fig. 3.12(a), and (b) thermal conductances from (i, j, k) to adjacent grids.

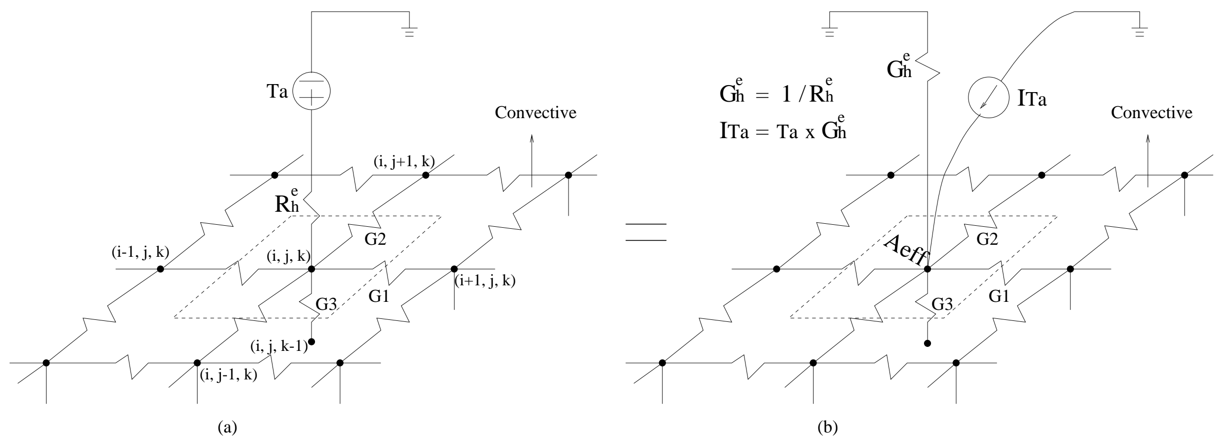


Figure 3.14 Equivalent thermal circuit at the convective boundary.

3.3.3 Analytical approach

Equations (3.1) and (3.2) can also be solved by the application of multiple-integral transform and multiple-inversion formulae [31] in the finite ranges of $0 \leq x \leq a$, $0 \leq y \leq b$ and $0 \leq z \leq c$, where a, b, c are the chip dimensions. We define the triple-integral and inversion formulae as

$$\begin{aligned}(Triple - Integral) \quad \overline{T}(\beta_m, \nu_n, \eta_p) &= \int_0^a \int_0^b \int_0^c K(\beta_m, x') \cdot K(\nu_n, y') \cdot K(\eta_p, z') \cdot \\ & \quad T(x', y', z') \cdot dx' \cdot dy' \cdot dz', \text{ and} \end{aligned} \tag{3.32}$$

$$\begin{aligned}(Triple - Inversion) \quad T(x, y, z) &= \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \sum_{p=0}^{\infty} K(\beta_m, x) \cdot K(\nu_n, y) \cdot K(\eta_p, z) \cdot \\ & \quad \overline{T}(\beta_m, \nu_n, \eta_p), \end{aligned} \tag{3.33}$$

where $K(\beta_m, x)$, $K(\nu_n, y)$, $K(\eta_p, z)$ are eigenfunctions and β_m, ν_n, η_p are eigenvalues. We now take the integral transform of (3.1) by applying (3.32),

$$\begin{aligned}\int_0^a \int_0^b \int_0^c K(\beta_m, x) \cdot K(\nu_n, y) \cdot K(\eta_p, z) \cdot (\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}) \cdot dx \cdot dy \cdot dz \\ + \frac{1}{k} \overline{g}(\beta_m, \nu_n, \eta_p) = 0, \end{aligned} \tag{3.34}$$

where $\overline{g}(\beta_m, \nu_n, \eta_p)$ is the integral transform of $g(x, y, z)$. By using Green's theorem, (3.34) can be transformed into the following expression which takes the BCs into account:

$$\begin{aligned}& k(\beta_m^2 + \nu_n^2 + \eta_p^2) \cdot \overline{T}(\beta_m, \nu_n, \eta_p) \\ &= A(\beta_m, \nu_n, \eta_p) \\ &= \overline{g}(\beta_m, \nu_n, \eta_p) + K(\beta_m, x)|_{x=0} \int_0^b \int_0^c f_1(y, z) \cdot K(\nu_n, y) \cdot K(\eta_p, z) \cdot dy \cdot dz \\ &\quad + K(\beta_m, x)|_{x=a} \int_0^b \int_0^c f_2(y, z) \cdot K(\nu_n, y) \cdot K(\eta_p, z) \cdot dy \cdot dz \\ &\quad + K(\nu_n, y)|_{y=0} \int_0^a \int_0^c f_3(x, z) \cdot K(\beta_m, x) \cdot K(\eta_p, z) \cdot dx \cdot dz \end{aligned}$$

$$
\begin{aligned}
&+K(\nu_n, y)|_{y=b} \int_0^a \int_0^c f_4(x,z) \cdot K(\beta_m, x) \cdot K(\eta_p, z) \cdot dx \cdot dz \\
&+K(\eta_p, z)|_{z=0} \int_0^a \int_0^b f_5(x,y) \cdot K(\beta_m, x) \cdot K(\nu_n, y) \cdot dx \cdot dy \\
&+K(\eta_p, z)|_{z=c} \int_0^a \int_0^b f_6(x,y) \cdot K(\beta_m, x) \cdot K(\nu_n, y) \cdot dx \cdot dy, \qquad (3.35)
\end{aligned}
$$

where f_1 - f_6 correspond to $f_i(x, y, z)$ in (3.2) for the six sides of the chip.

For brevity, we only present the solution for the case where all four sides and the top surface of the chip are insulated, while the bottom surface is convective. In this case, we have

$$
\begin{aligned}
K(\beta_m, x) &= \sqrt{2/a}\, \cos(\beta_m x) \quad \text{with} \quad \sin(\beta_m a) = 0, \\
K(\nu_n, y) &= \sqrt{2/b}\, \cos(\nu_n y) \quad \text{with} \quad \sin(\nu_n b) = 0, \quad \text{and} \\
K(\eta_p, z) &= \sqrt{2}\, [\frac{\eta_p^2 + H^2}{c(\eta_p^2 + H^2) + H}]^{1/2} \cos(\eta_p(c - z)) \\
&= Q_p \cos(\eta_p(c - z)) \quad \text{with} \quad \eta_p \tan(\eta_p c) = H \quad \text{and} \quad H = h_z^e / k,
\end{aligned}
$$

where h_z^e is the effective heat transfer coefficient of the bottom surface. Note that when β_m in $K(\beta_m, x)$ is zero, the coefficient $\sqrt{2/a}$ of $K(\beta_m, x)$ has to be replaced by $\sqrt{1/a}$ in order to retain the eigenfunction normalities. Same argument also applies to ν_n in $K(\nu_n, y)$. Now $A(\beta_m, \nu_n, \eta_p)$ in (3.35) becomes

$$
\begin{aligned}
A(\beta_m, \nu_n, \eta_p) &= \overline{g}(\beta_m, \nu_n, \eta_p) \\
&\quad + K(\eta_p, z)|_{z=0} \int_0^a \int_0^b (h_z^e \cdot T_a) \cdot K(\beta_m, x) \cdot K(\nu_n, y) \cdot dx \cdot dy. \qquad (3.36)
\end{aligned}
$$

In (3.36), $\overline{g}(\beta_m, \nu_n, \eta_p)$ can be expressed as

$$
\begin{aligned}
\overline{g}(\beta_m, \nu_n, \eta_p) = \frac{16 Q_p}{\sqrt{ab}} \frac{1}{\beta_m \nu_n \eta_p} \sum_{i=1}^{n_r} g_i \cos(\beta_m x_{ic}) \sin(\frac{\beta_m}{2} x_{id}) \cos(\nu_n y_{ic}) \sin(\frac{\nu_n}{2} y_{id}) \cdot \\
\cos(\eta_p(c - z_{ic})) \sin(\frac{\eta_p}{2} z_{id}) \qquad (3.37)
\end{aligned}
$$

for $m \neq 0$ and $n \neq 0$, where (x_{ic}, y_{ic}, z_{ic}), (x_{id}, y_{id}, z_{id}) and g_i are the center coordinates, dimensions, and the power density of heat source i, respectively, and n_r is the number of heat sources. For $m = 0$ and/or $n = 0$, a similar expression for $\overline{g}$ can be derived. Once $\overline{T}$ has been determined, the on-chip temperature at any position can subsequently be found by (3.33). The number of terms used in series expansion is actually finite and it is terminated once the additional term does not change the temperature by more than a small specified amount (e.g., 0.01°C).

3.3.4 Discussion

As an accuracy check on our numerical and analytical thermal simulation methods, we perform the following experiments. Consider a chip containing ten heat sources, all with dimensions of 50 μm $\times$ 50 μm. The sources are confined within the area (source area) with dimensions of 500 μm $\times$ 500 μm inside a chip, and the distance between the boundary of the source area and the chip's bonding pad is 500 μm. This is graphically shown in Fig. 3.15. Next, the heat sources are randomly distributed within the source area, and the power values ranging from 10 mW to 100 mW are randomly assigned to the heat sources. A convective BC is used for the the bottom surface of the chip, while the side and top surfaces are assumed perfectly insulated. Thermal simulation results are compared with the results from a 3-D thermal simulator, THUNDER [32]. Different values of the bottom heat transfer coefficient are tested as shown in Table 3.2, where Max_$\triangle T_{num}$ denotes the maximum percentage error of numerical simulation results compared to THUNDER (i.e., temperatures are compared everywhere in the chip and

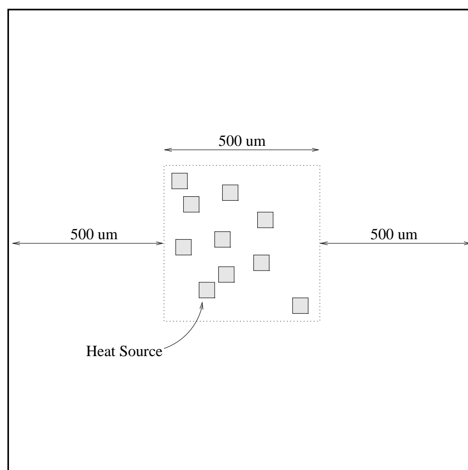


Figure 3.15 Chip structure and heat source locations in the experiment.

Table 3.2 Comparison between iTEMP and THUNDER simulation results.

h $[W/m^2 \cdot K]$	5,000	8,000	10,000	15,000	20,000	25,000	30,000
Max_$\triangle T_{num}$ (%)	3.67	2.36	4.77	3.20	4.38	3.02	4.46
Max_$\triangle T_{ana}$ (%)	2.14	2.49	3.01	2.95	3.34	3.27	3.63

the largest error is recorded). In the third row of Table 3.2, Max_$\triangle T_{ana}$ is similar to Max_$\triangle T_{num}$ but the analytical simulation results are compared to THUNDER.

Among three different thermal simulation methods, the FTA method is primarily used for fast hot-spot identification in the early chip design phase. In order to observe how accurately the FTA method can identify the hot spots, we perform the following experiment for the same layout shown in Fig. 3.15. Ten heat sources are randomly distributed and the power values from 10 mW to 100 mW are randomly assigned to the sources, and this process is repeated fifty times (i.e., 50 tests). We define that a violation occurs in a test if the hot spot identified by the FTA method is different from the one

Table 3.3 Violation rate by using the FTA method.

h $[W/m^2 \cdot K]$	5,000	8,000	10,000	15,000	20,000	25,000	30,000
Violation rate (%)	8	10	10	8	8	12	6
$\triangle T_{hot_spot}$ $[\ ^\circ C]$	1.1	0.83	0.6	0.8	0.35	0.75	0.21

identified by our numerical method. The violation rates among fifty tests using the FTA method are shown in the second row of Table 3.3 for different h values. The $\triangle T_{hot_spot}$ term stands for the averaged difference of the actual temperatures of the two distinct hot spots identified by the FTA and the numerical methods. The data shown in Table 3.3 imply that a violation occurs only when the two spots have very close temperature values.

In our ILLIADS-T application, the numerical thermal simulation method is generally preferred to the analytical method for the following two reasons. First, the nonuniformity (i.e., the location dependency) of the thermal conductivity k cannot be handled in the analytical approach. Second, the nonclosed form of the triple series summation in (3.33) is computationally more expensive than the numerical method [33], and this is aggravated when the number of points at which the temperature needs to be calculated is large (e.g., full-chip temperature profile estimation). For example, the chip shown in Fig. 3.15 is simulated by both methods, and the temperatures of 400 mesh points on the chip are calculated. The numerical method requires 26.09 seconds and the analytical one requires 448 seconds of CPU time on SPARC 10. The FTA method is also used for simulating this case, and it requires only 0.2 seconds to find the hot spot among the mesh points. The analytical method, however, has its own advantage. Because it provides an explicit expression for the temperature of a point (x,y,z), it is very useful if we want to directly calculate the temperature of some specified points (e.g., hot spots identified after using

the FTA method) rather than solving the temperature profile of the whole chip. In other words, the analytical method has the resolution high enough to efficiently pinpoint the temperature of the on-chip hot spot.

3.4 Package Simulation

3.4.1 Modeling of convective boundaries

The effective heat transfer coefficient h^e in (3.6) describes how significantly heat transfers between the chip and the ambience. Its value is determined by both the package (or sink) structure and the efficiency of the heat removal process. Consider a chip with dimensions of 1000 μm $\times$ 1000 μm. It contains three heat sources with power values shown in Fig. 3.16. The heat transfer coefficient of the bottom sink is given as 10,000 (W/m^2 °C).

Assuming that the top and four sides of the chip all have the h^e values of 0 (perfect insulation), 8.0 (natural convection), and 5,000, respectively, we perform thermal simulation of the chip and the temperature profiles along the x direction at $y = 500$ μm are shown in Fig. 3.17 for different h^e values. For a packaged chip under a natural convective condition, the boundaries of the top and all four sides are often approximated as perfectly insulated in the simulation. It is verified by the results shown in Fig. 3.17, where the temperature difference in the two conditions is very small. However, if a chip is under a forced-convective condition (e.g., $h^e = 5,000$), the chip boundaries can no longer be modeled as perfectly insulated. As can be seen from Fig. 3.17, the forced convection greatly reduces the overall chip temperatures.

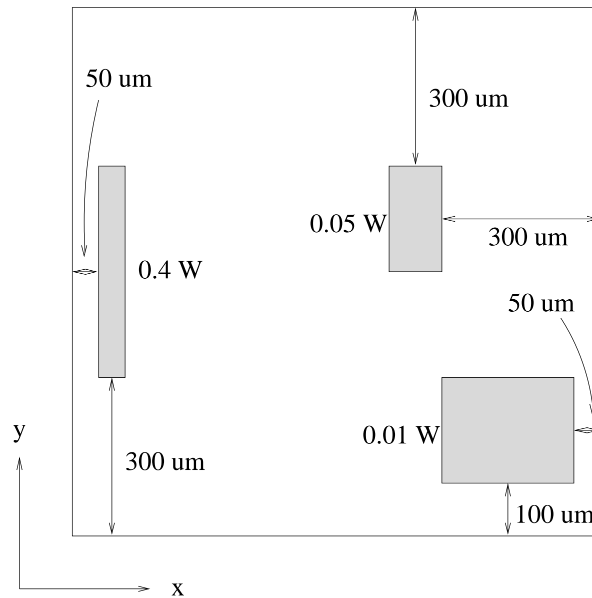


Figure 3.16 Layout of the chip containing three heat sources.

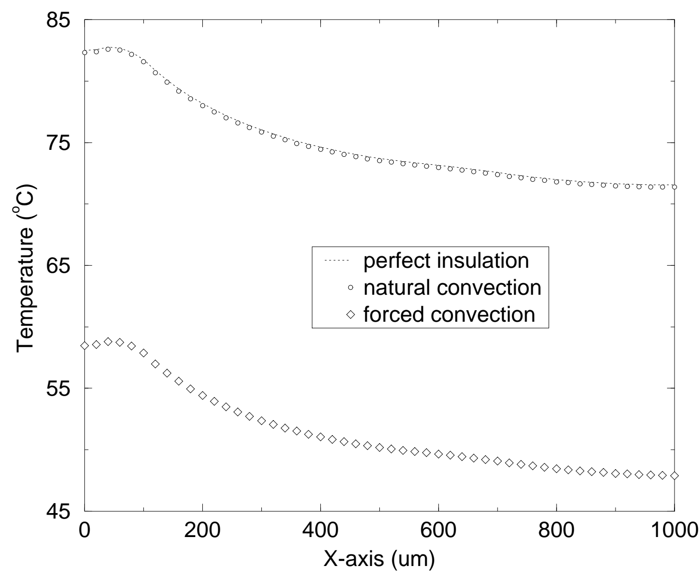


Figure 3.17 Temperature profiles along the x direction at $y = 500\ \mu$m for three different h^e values.

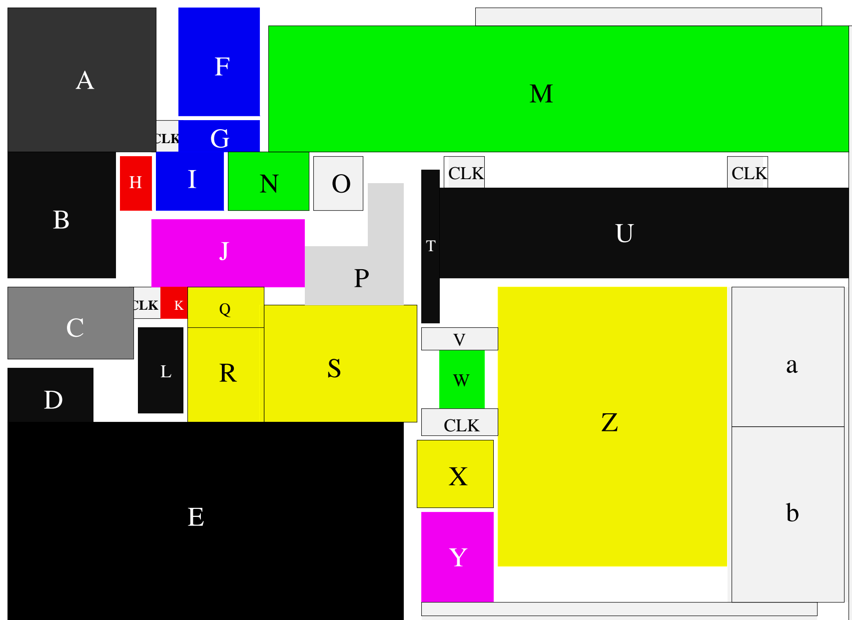


Figure 3.18 Unit-level layout of the microprocessor chip.

3.4.2 Modeling of heat flow paths

iTEMP has been applied to simulate a microprocessor chip with the most advanced packaging technology. The unit-level layout of the chip is shown in Fig. 3.18. Each unit contains several functional unit blocks (FUBs), and the power values of all FUBs are given. There are a total of 310 FUBs in the chip. A cross-sectional view of the flip-chip package is shown in Fig. 3.19. The flip-chip bonding technology offers a better packaging solution, but also brings challenges for heat removal from the chip to the package (through the bumps in Fig. 3.19). Furthermore, the heat sink (i.e., heat pipe) must be efficient enough to serve as the major heat removal path. From measurements, the temperature at the surface of the heat pipe is estimated to be 45°C. The equivalent thermal circuit of Fig. 3.19 is shown in Fig. 3.20, and the symbol definitions are listed in

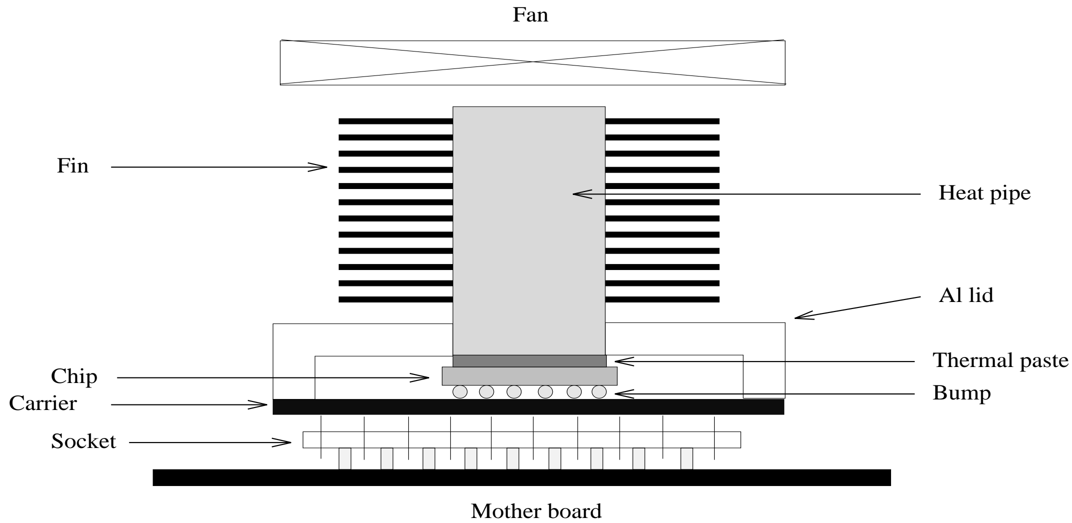


Figure 3.19 Cross-sectional view of a flip-chip package.

Table 3.4. To formulate the effective heat transfer macromodel, we have to determine the thermal resistances of the carrier (R_{car_down}, R_{car_side}) and lids (R_{lid1}, R_{lid2}), etc. Using R_{car_side} as an example, to determine its lumped value, we fix the boundaries at four sides of the carrier to be in constant temperature T_a, and the top (except for the chip-carrier interface) and bottom surfaces of the carrier to be insulated. The lumped thermal resistance of the carrier can therefore be calculated as $R = (\frac{T_{avg} - T_a}{I/4})$, where T_{avg} is the average temperature at the chip-carrier interface which is precharacterized by using FD thermal simulation, and I is the chip power. This procedure is graphically shown in Fig. 3.21.

The heat flow path from the carrier to the heat pipe via the aluminum lid is shown in Fig. 3.22. To find its equivalent thermal resistance (R_{lid2} in Fig. 3.20), we can use the similar approach (i.e., heat transfer macromodeling) as for the carrier. Instead, we calculate $R_{lid2} = R_1 + R_2 + R_3 + R_4$ by using the analytical formulae. From [34], R_1 and

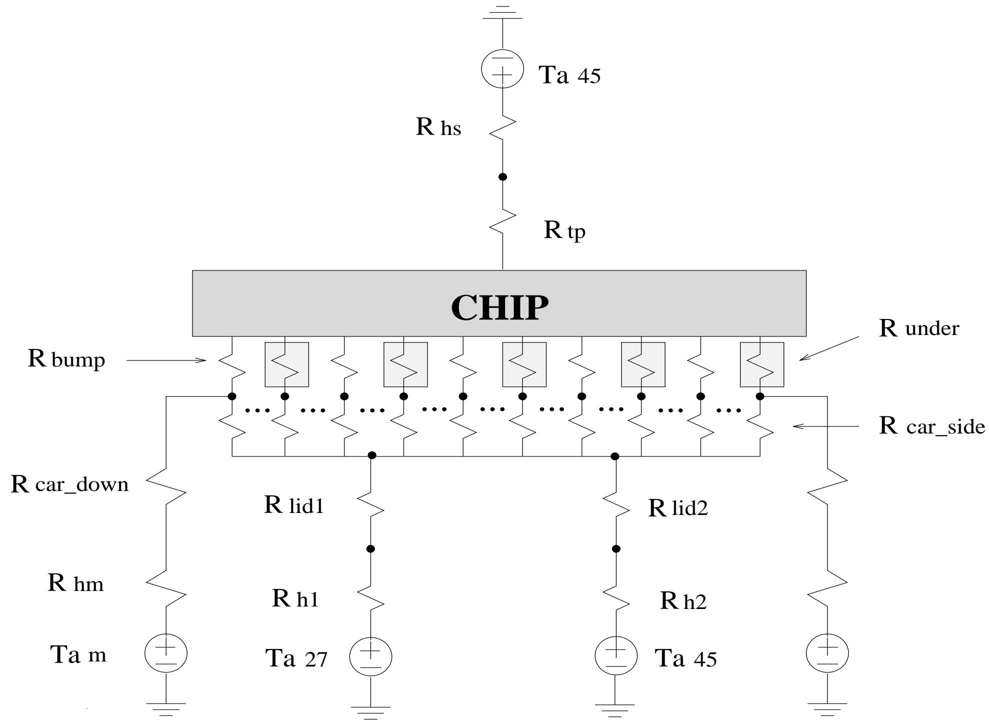


Figure 3.20 Equivalent thermal circuit of the flip-chip package.

Table 3.4 Definition of the symbols in Fig. 3.20.

Symbol	Definition
Ta_{27}	Ambient temperature (27°C)
Ta_{45}	Ambient temperature (45°C)
Ta_m	Ambient temperature near mother board
R_{car_down}	Thermal resistance for heat flowing through carrier down to the mother board
R_{car_side}	Thermal resistance for heat flowing through carrier aside to the lids
R_{bump}	Thermal resistance for heat flowing through bumps
R_{under}	Thermal resistance for heat flowing through underfills
R_{lid1}	Thermal resistance for heat flowing through lids to air
R_{lid2}	Thermal resistance for heat flowing through lids to the heat pipe
R_{tp}	Thermal resistance for heat flowing through the thermal paste
R_{h1}	Thermal resistance for heat transfer between lid surface and Ta_{27}
R_{h2}	Thermal resistance for heat transfer between lid surface and Ta_{45}
R_{hs}	Thermal resistance for heat transfer between the thermal paste and Ta_{45}
R_{hm}	Thermal resistance for heat transfer between the carrier and mother board

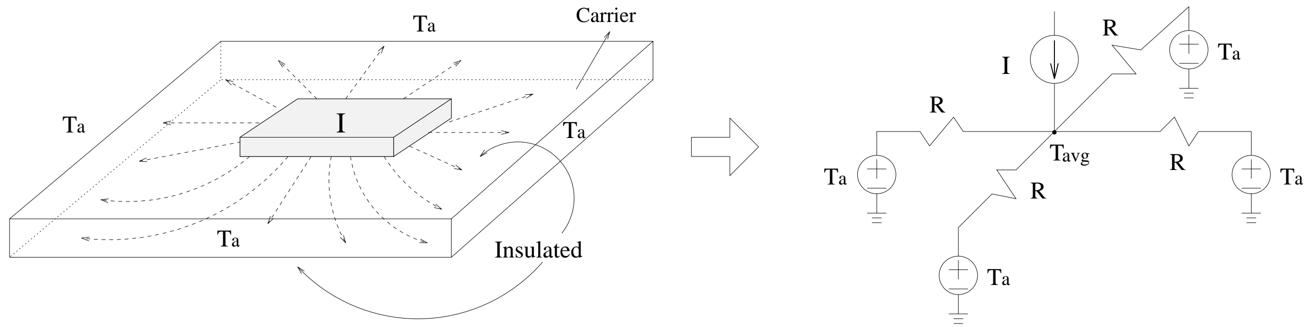


Figure 3.21 Method to determine the thermal resistances for heat flowing through the carrier aside to the lids.

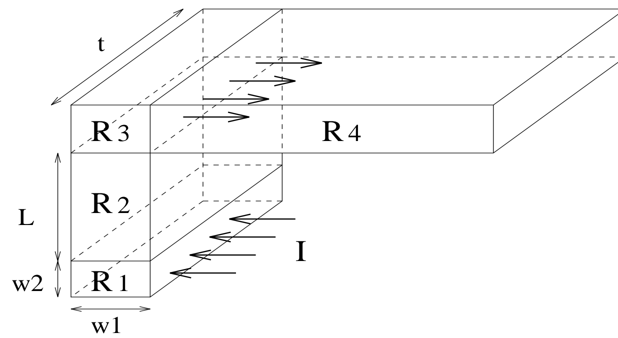


Figure 3.22 A bend structure of the aluminum lid.

R_2 can be expressed as

$$R_1 = 0.5 \cdot \frac{1}{K_{Al} \cdot t}[\frac{1}{a} - \frac{2}{\pi}\ln(\frac{4a}{a^2+1}) + (\frac{a^2-1}{\pi a})\cos^{-1}(\frac{a^2-1}{a^2+1})], \quad (3.38)$$

$$R_2 = \frac{L}{K_{Al} \cdot (w_1 t)}, \quad (3.39)$$

and K_{Al} is the thermal conductivity of aluminum and $a = w_1/w_2$ with $w_1 \geq w_2$. The values of R_3 and R_4 can be calculated in similar manners.

After obtaining all the necessary information, we can now refer back to Fig. 3.20 and start the thermal simulations. In order to see how important the packaging effect is to the overall on-chip temperature, three different experiments are performed. In the first experiment, we ignore the contact resistance between the thermal paste and the heat pipe (≈ 0.15°C/W). Figure. 3.23 shows the simulated temperature contour. In the second experiment, we take into account the contact resistance, but assume that there is no heat flowing through the carrier (i.e., $R_{car_side} = R_{car_down} = \infty$). The simulation result is shown in Fig. 3.24. In the third experiment, we consider both the contact resistance and the finite carrier thermal resistance, and the result is shown in Fig. 3.25.

3.5 Incremental Electrothermal Simulation

As we mentioned previously, the decoupled electrothermal simulation method usually takes less than three iterations to find the steady-state temperature; the iteration process stops when the current temperature converges to the value in the previous simulation run. This is illustrated in Fig. 3.26, where a nine-stage ring oscillator is simulated by ILLIADS-T. The power and temperature values were recorded during the iteration process. In Fig. 3.26, the temperature difference between two successive simulation runs

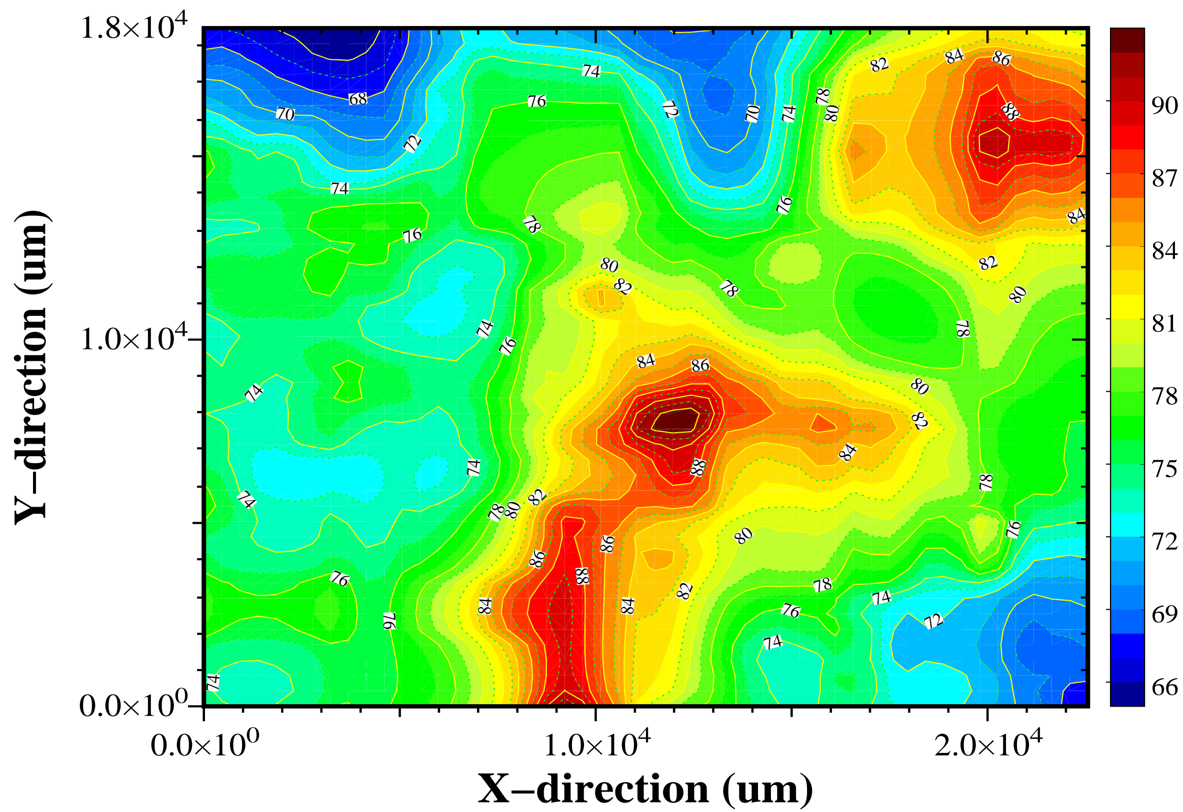


Figure 3.23 On-chip temperature contour for the first experiment.

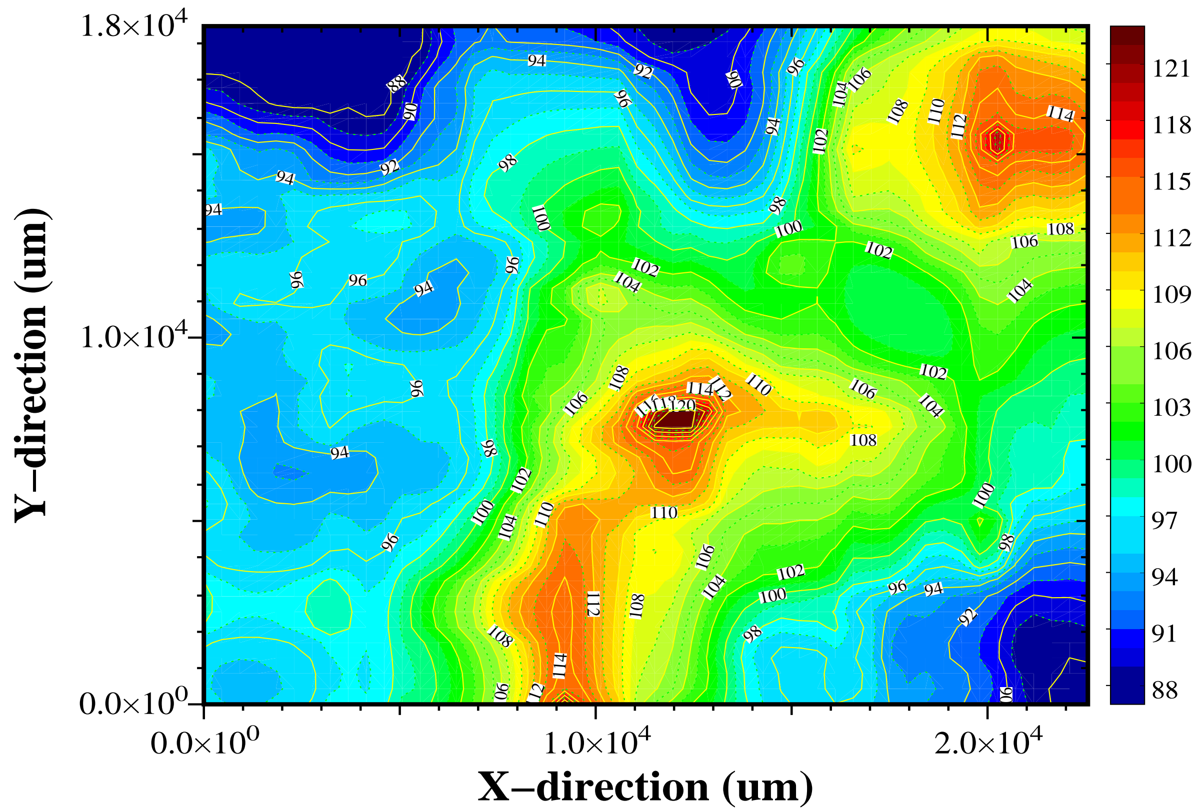


Figure 3.24 On-chip temperature contour for the second experiment.

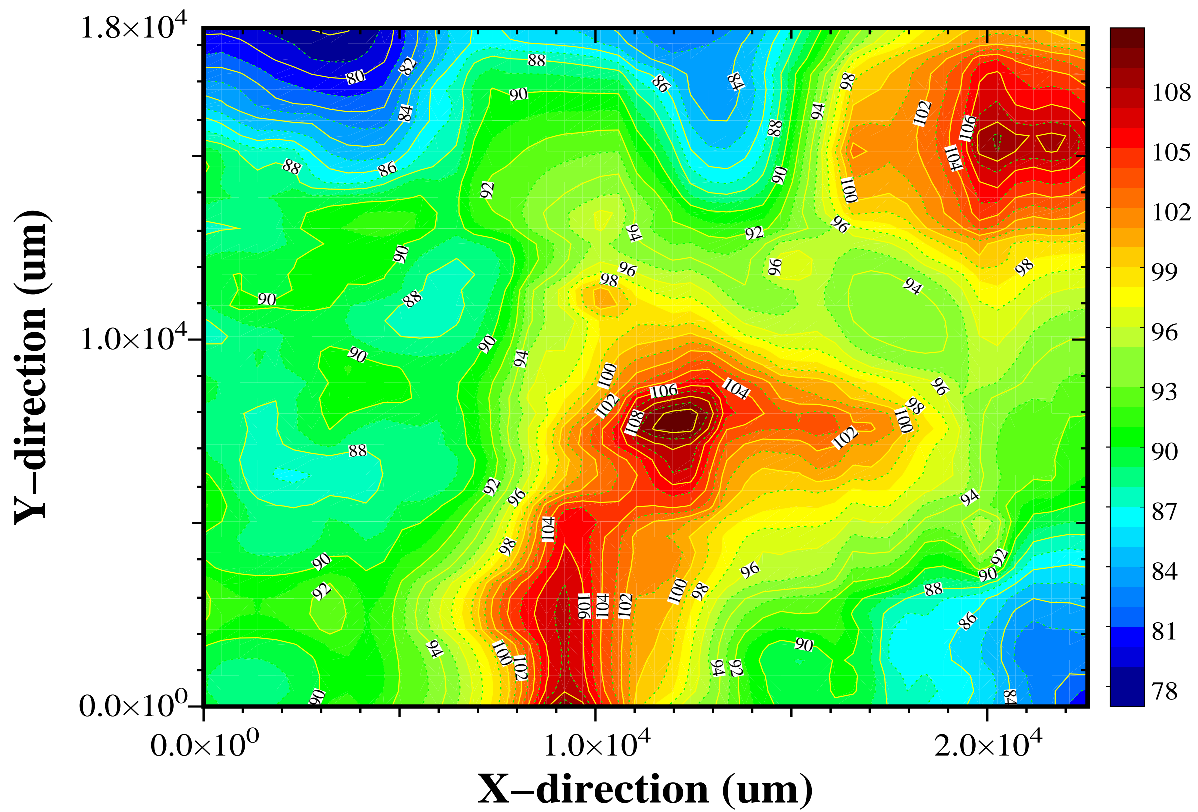


Figure 3.25 On-chip temperature contour for the third experiment.

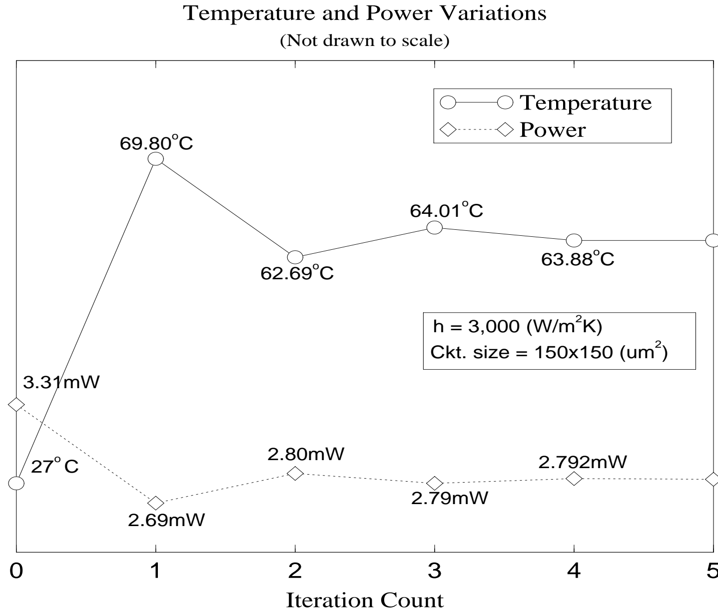


Figure 3.26 Convergence plot for power and temperature.

becomes smaller as the number of iterations increases. This information indicates the feasibility of *incremental simulation*, in which the circuit parameters vary by small amounts compared to their previous values.

Consider a circuit containing blocks that are ordered for simulation based on the fan-in-fan-out relationship. In ILLIADS-T, if the temperature difference in a block between current (perturbed) and previous (nominal) runs exceeds a prescribed threshold (T_THRLD), then we deem that the block has local temperature variation and it will be marked with T_VAR. For a block marked with T_VAR, it is not considered to be *latent* and it needs to be simulated. The resimulated waveforms then serve as the nominal waveforms for the next simulation run. However, if a block is not marked with T_VAR, then the nominal and perturbed waveforms for all of the inputs to the block are compared. If

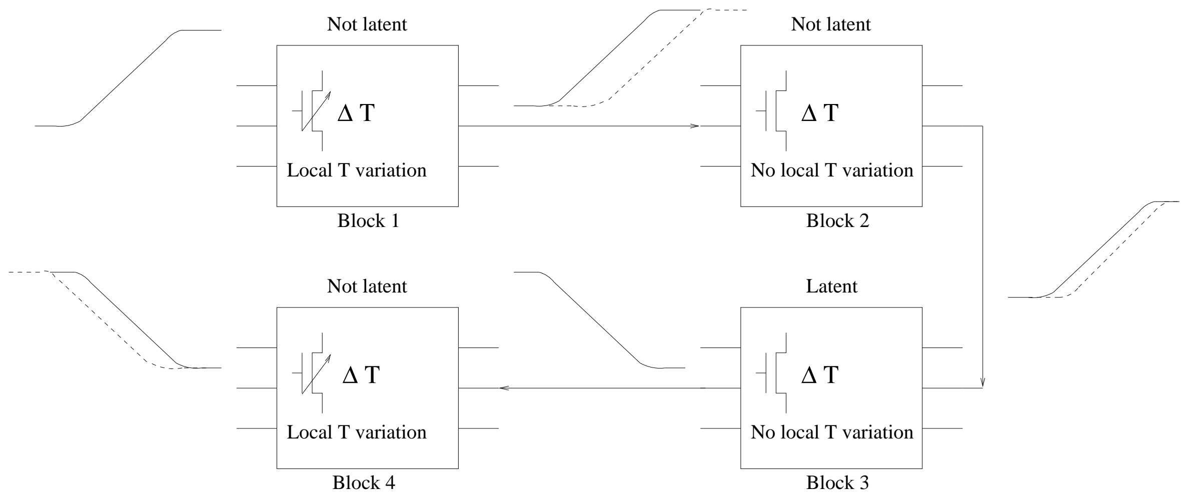


Figure 3.27 Illustration of incremental latency; the nominal waveforms are shown in solid lines, while the perturbed waveforms are in dashed lines.

the difference between them is less than a user-specified threshold, the block is marked as being incrementally latent and is not simulated (i.e., its perturbed solution is the same as its nominal solution). On the other hand, if the difference in any of the inputs is larger than the threshold, the block is not considered to be latent and is simulated.

This procedure is illustrated in Fig. 3.27, where Block 1 and Block 4 have local temperature variations and are incrementally resimulated. For Block 2, there is no local temperature variation, but the difference between its nominal and perturbed input signals is large and it, too, is resimulated. However, Block 3 is considered to be latent because besides having no temperature variation, the difference between its nominal and perturbed inputs is very small. Thus, for Block 3, the incremental simulation is skipped and its perturbed solution is assumed to be the same as its nominal results.

Note that ILLIADS-T identifies the blocks having local temperature variations *dynamically* for each new simulation run. In other words, a block can be marked with

T_VAR in one run, but is not marked in another. As the number of iterations increases in the ILLIADS-T simulation, the advantage of the incremental approach will appear even greater because it is expected that a large number of latent cases will be detected. For larger circuits, we also expect that the computational savings of latency will be more pronounced, because a larger number of blocks will be latent. Furthermore, for circuits with a larger temperature gradient (e.g., due to either a large power density variation or a special kind of boundary condition), the incremental technique will be even more effective. The simulation speedup due to the incremental approach will be presented in Chapter 4.

CHAPTER 4

VERIFICATION OF ILLIADS-T AND SIMULATION RESULTS

4.1 Tester Chip Design and Calibration

A tester chip was designed for the verification of simulation accuracy. It was fabricated using 0.8 μm CMOS technology and packaged by MOSIS. Figure 4.1 shows the layout of the chip, where the blocks I, III and V are high-frequency 3-stage ring oscillators designed in a standard super-buffer configuration. Blocks II and IV are 149-stage ring oscillators, and the three small dots (D1, D2, D3) are diodes. Henceforth, we denote the 3-stage and 149-stage oscillators as Rosc3s and Rosc149s, respectively. Each ring oscillator has an enable signal which is used to activate or deactivate the oscillator. Because the operating frequency of the Rosc149s is much lower than that of Rosc3s, power is mainly dissipated from the Rosc3s. The on-chip temperature can be determined by measuring the voltage drop across the forward-biased diodes according to $V_F = (kT/q)\ln(I_F/I_s(T) + 1)$, where I_F is the forward-bias current provided by a constant current source. The oscillation frequency of the Rosc149s can also be measured before and after Rosc3s are turned on to observe any change due to the on-chip temperature rise.

Figure 4.2 shows the diode circuit designed for the temperature measurement. Because the voltage drop across the lead resistance of the diode is also a function of

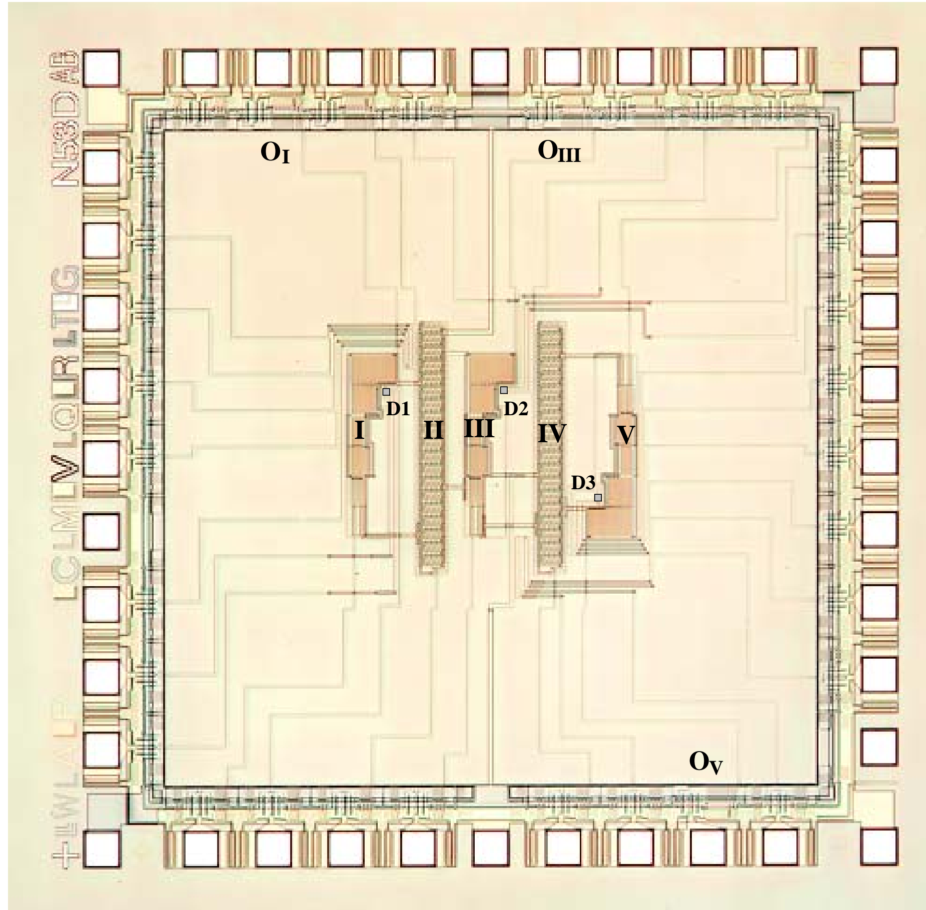


Figure 4.1 Layout of tester chip, where long blocks are Rosc149s and short blocks are Rosc3s.

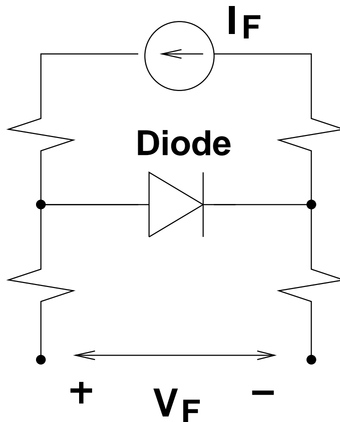


Figure 4.2 Four-terminal configuration for diode measurement.

temperature, a four-terminal configuration is used to cancel out the voltage drop in the test leads. The diodes are calibrated individually by measuring V_F at different temperatures. The diode temperature is controlled by placing the chip upside-down on a hot plate after the chip lid has been removed. The temperatures on the surface of the hot plate are accurately determined by placing a *thermistor* on the plate and measuring its resistance values. These values are then translated to the temperatures of the thermistor, namely, the temperatures of the hot plate. The I_F values are kept small, so that self-heating from the diode may be ignored. An example of the calibration data is shown in Fig. 4.3. When the tester chip is operating, the local temperature near the diode is determined by comparison with the calibration data. The package thermal parameters are also calibrated based on the MOSIS handbook for the DIP40 package. The effective heat transfer coefficient of the chip bottom (h^e in (3.6)) is determined to be 8,689 (W/(m^2 °C)) with all other sides insulated.

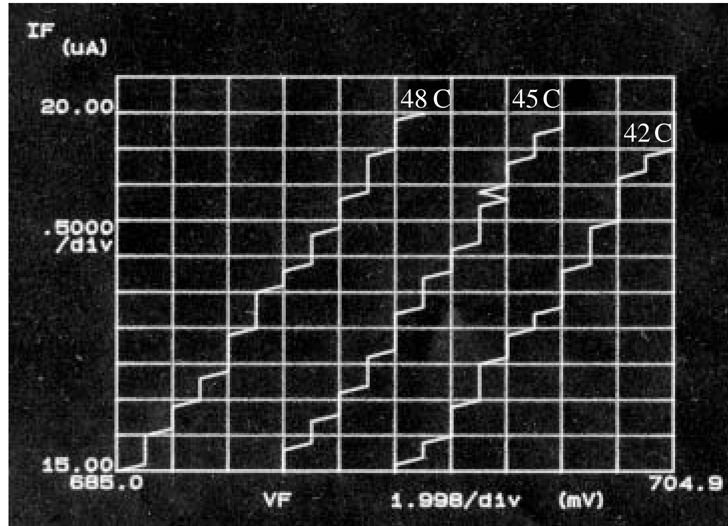


Figure 4.3 Diode calibration example (D1).

4.2 Verification of ILLIADS-T

During the tester chip experiments, Rosc149s were always activated while the chip power consumption was varied by activating different Rosc3s. Depending on the on/off status of the Rosc3s, there are eight unique experiments as shown in Table 4.1. For example, the ILLIADS-T-simulated temperature profiles for Expt. 2 (block I and III on, block V off) and Expt. 1 (all blocks on) are shown in Figs. 4.4 and 4.5, respectively. Note that the power dissipation from the output buffers of blocks I, III, and V (O_I, O_{III}, and O_V in Fig. 4.1) was also taken into account. The simulated and measured diode temperatures for all eight experiments are compared in Figs. 4.6 - 4.8. In these figures,

Table 4.1 Activation status of Rosc3s.

Expt. #	1	2	3	4	5	6	7	8
Block I III V	111	110	101	100	011	010	001	000

1: on 0: off.

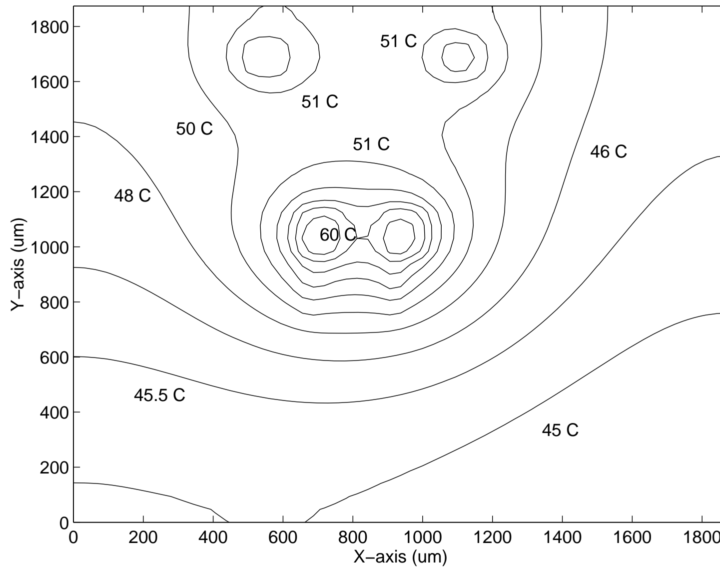


Figure 4.4 Simulated temperature profile for Expt. 2.

the error bars show the spread of the measured data. Good agreement between measured and simulated temperatures was found.

ILLIADS-T was also used to predict the frequency shift of Rosc149s due to the local temperature rise. The mobility-temperature relationship was extracted from frequency measurements on block II, and the mobility model (2.17) was employed to obtain the optimized fitting parameters A_1 - A_4. Next, the mobility model was used in ILLIADS-T to predict the frequency shift of block IV for the eight experiments, and the results are compared with the measured data as shown in Figs. 4.9 - 4.12. Additional simulation results are presented in Table 4.2, where P_{avg} is the average power consumption of the chip (including output buffers), and T_{blk4} is the average temperature of block IV. The oscillation frequencies of block IV before and after electrothermal simulation are shown in the fourth column of Table 4.2. Note that as the temperature increases, the oscillation

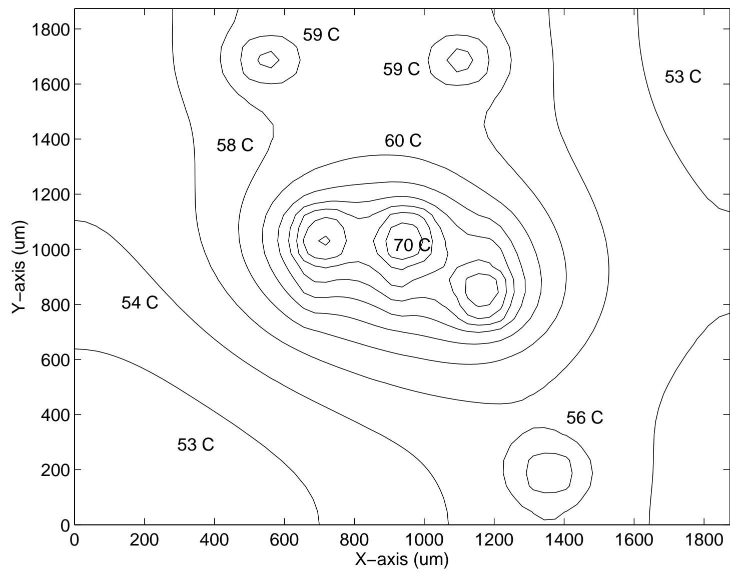


Figure 4.5 Simulated temperature profile for Expt. 1.

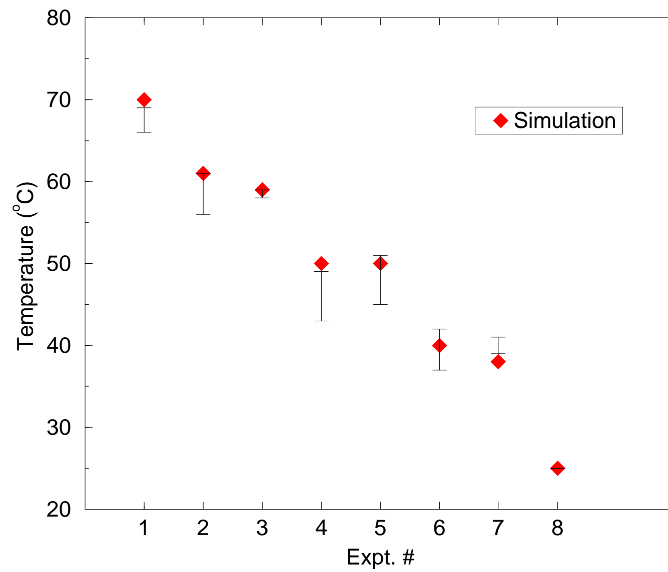


Figure 4.6 Comparison between simulated and measured temperatures for D1.

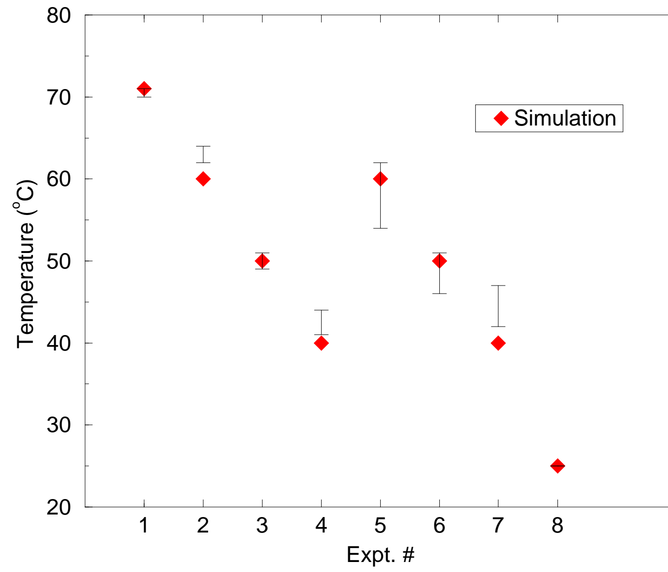


Figure 4.7 Comparison between simulated and measured temperatures for D2.

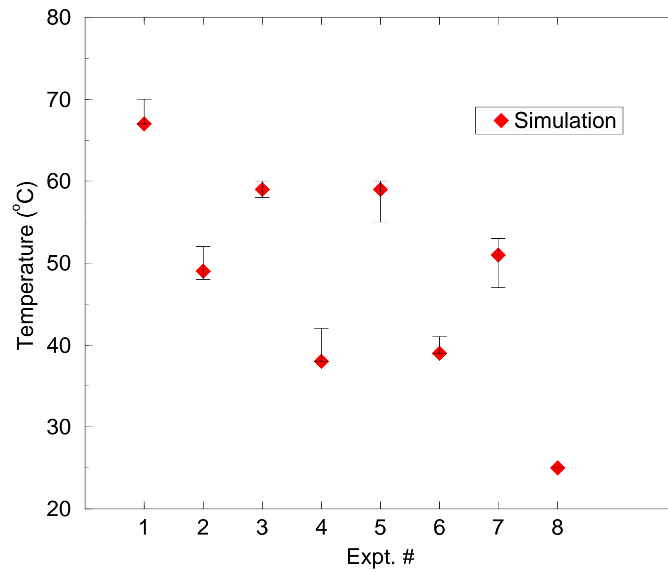
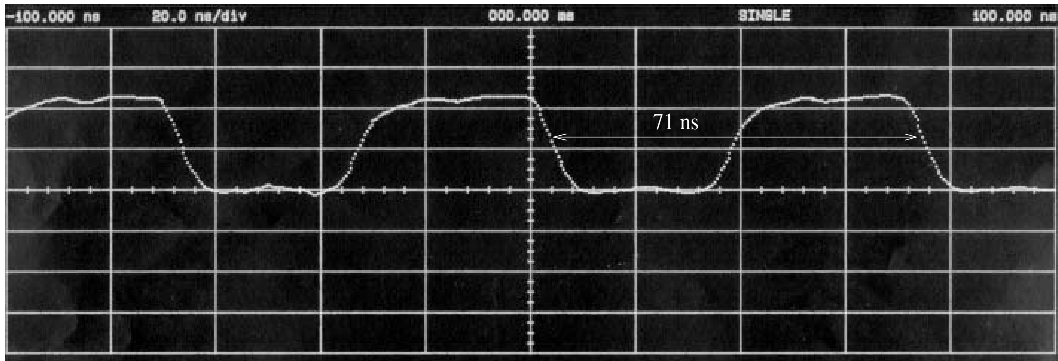
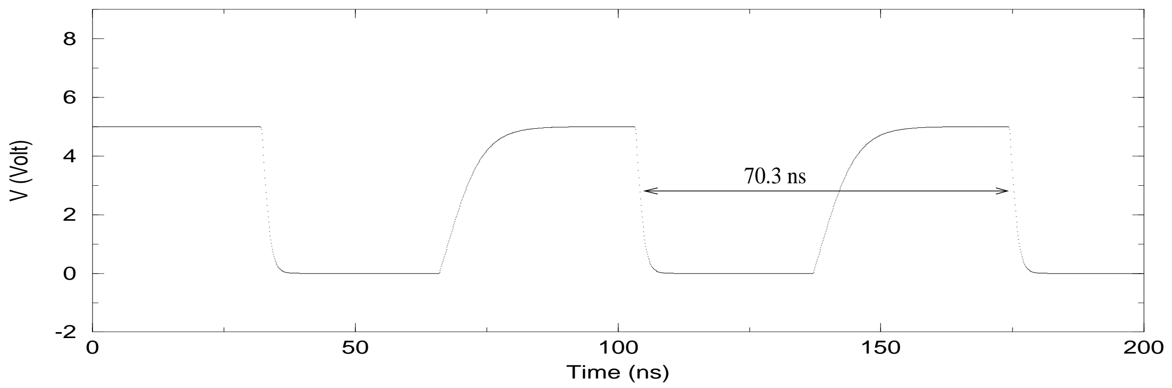


Figure 4.8 Comparison between simulated and measured temperatures for D3.

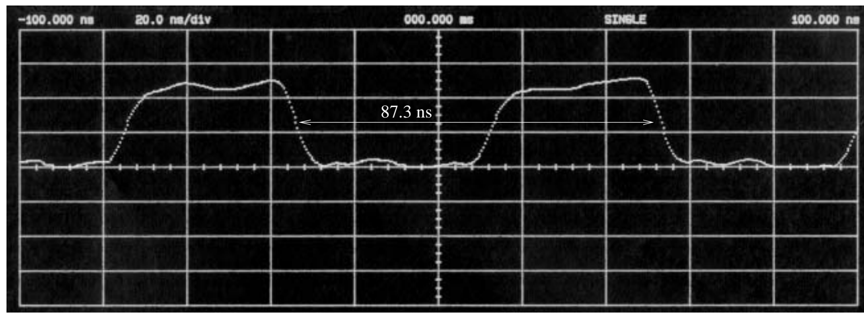


(a)

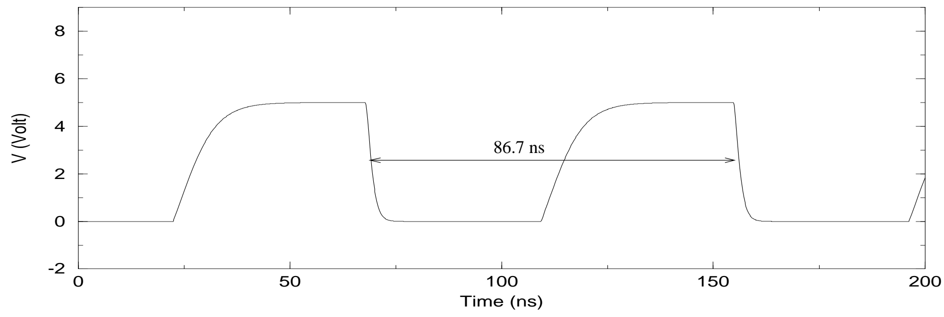


(b)

Figure 4.9 (a) Measured and (b) simulated waveforms for Expt. 8.

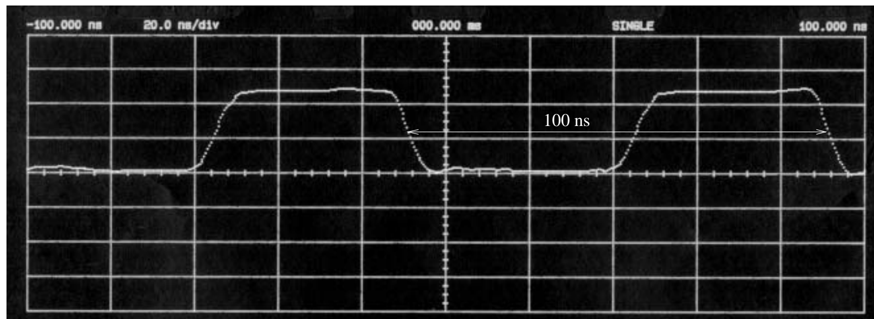


(a)

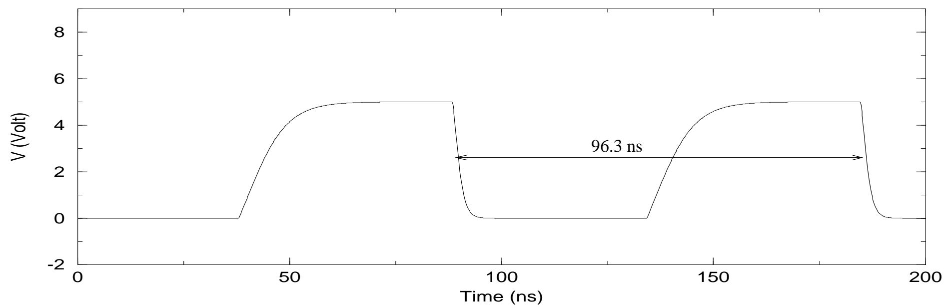


(b)

Figure 4.10 (a) Measured and (b) simulated waveforms for Expt. 7.

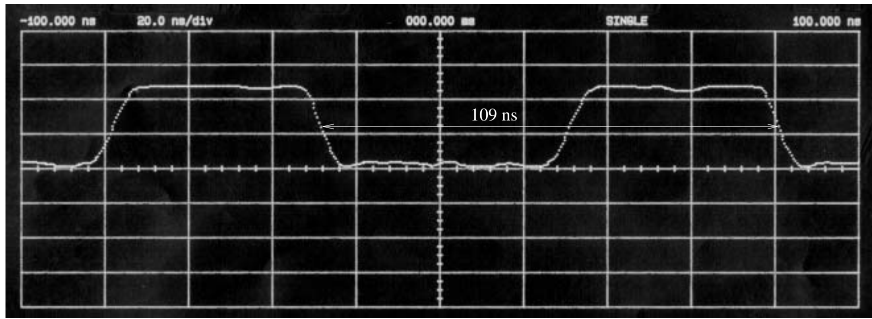


(a)

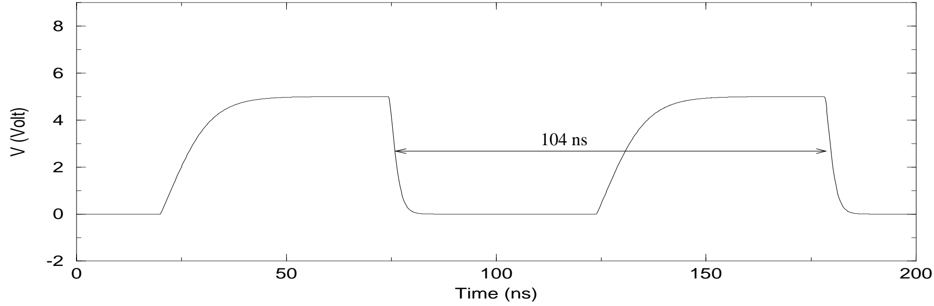


(b)

Figure 4.11 (a) Measured and (b) simulated waveforms for Expt. 5.



(a)



(b)

Figure 4.12 (a) Measured and (b) simulated waveforms for Expt. 1.

Table 4.2 ILLIADS-T simulation results of the tester chip.

Tester chip	P_{avg} [Watt]	T_{blk4} [$°C$]	Freq. shift [MHz]	CPU time[1] [sec]
Expt. 7	0.350	44.17	14.07 → 11.53	422
Expt. 5	0.636	56.03	14.07 → 10.38	650
Expt. 1	0.882	63.43	14.07 → 9.62	822

[1]On SUN SPARCstation 10.

frequency is significantly lowered and, consequently, so is the power. Therefore, the power values listed in Table 4.2 were calculated at the simulated operating temperature.

4.3 ILLIADS-T Simulation Examples

Next, we demonstrate ILLIADS-T simulation results for a number of other circuits. Here we assume that the top and the four sides of the circuits are insulated, while the effective heat transfer coefficient of the bottom surface is assumed to be 3,000 for all test

ROW-BASED FLEXIBLE-CELL LAYOUT

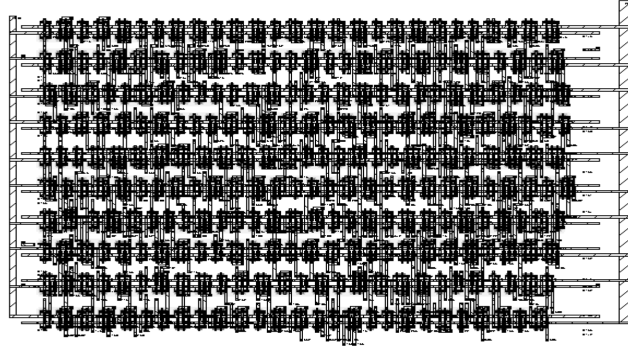


Figure 4.13 Layout of the 10-bit negative adder.

circuits. We first consider a 10-bit negative adder with the layout shown in Fig. 4.13. The layout was generated by a synthesis tool iCGEN [35]. Simulation results are listed in the third row of Table 4.3, where n_{tran} and n_{hsrc} are the numbers of transistors and heat sources in the circuit, respectively; P_{avg} is the average total power consumed in the circuit; T_{avg} is the average circuit temperature; n_{run} is the number of repeated simulation runs; and S_{fac} is the speedup factor of the electrothermal timing simulation, which is computed as the ratio of the total (i.e., including all simulation runs) transient analysis time without the incremental simulation to the transient analysis time with the incremental simulation. Table 4.3 also presents the ILLIADS-T simulation results for several other circuits such as HIGHWAY [36], ALU and control, and a 16-bit multiplier.

Another simulation example is shown in Fig. 4.14. This chip contains two ISCAS85 benchmark circuits, C3540 and C6288, one negative adder, and two three-stage ring oscillators identical to Rosc3. The packaging structure for the chip is shown in Fig. 4.15 and the corresponding thermal parameters are given in Table 4.4. The heat transfer coefficient between the heat sink and the ambience is assumed to be 12,000 (W/(m^{2}K)).

Table 4.3 ILLIADS-T simulation results.

Circuit	Ckt. size	n_{tran}	n_{hsrc}	Freq.	P^1_{avg}	T_{avg}	n^2_{run}	CPU time[3]	S_{fac}
Unit	μm^2	-	-	MHz	mW	°C	-	sec	-
10-bit Neg. Adder	450 × 300	868	216	100	12.07	49.40	3	31.35	1.24
HIGHWAY	150 × 210	248	17	33	1.92	46.15	3	23.01	1.21
ALU and Control	1730 × 1540	5842	1656	200	67.67	35.03	2	211.76	1.00
16-bit Multiplier	2180 × 2330	11016	3001	100	289.32	45.75	3	738.08	1.28

[1]Under steady-state temperature distribution. [2]Convergence criterion: $(\Delta T/T_{rise}) < \%1$.
[3]On SUN SPARCstation 10.

Table 4.4 Materials and thermal parameters for the packaging structure.

Parameter Description	Units	Bulk	Bump	Polymer	Substrate	Gel	Bottom Sink
Material	-	doped Si	Sn/Pb	polymide	doped Si	silicone gel	Al
Therm. Cond.[1]	$W/(mK)$	98.40	53.4	0.25	98.40	0.4	216.5
Thickness	mm	0.25	0.025	0.005	0.5	0.005	1.0

[1]Data from [37].

Simulation results are presented in Table 4.5, where T_{max} and T_{min} are the pinpointed maximum and minimum temperatures of individual circuits. To demonstrate the importance of performing the temperature-dependent simulation, the output waveforms at bit ten of the negative adder, with and without electrothermal simulation, are compared in Fig. 4.16. A logic fault due to a timing problem is identified via the electrothermal simulation. Note that even when the temperature of the negative adder is assigned to 35°C, which is the average chip temperature that would be used in conventional simulations, the thermally induced fault still cannot be detected as shown in Fig. 4.16. This suggests that the on-chip temperature variation must be considered in timing verification, and

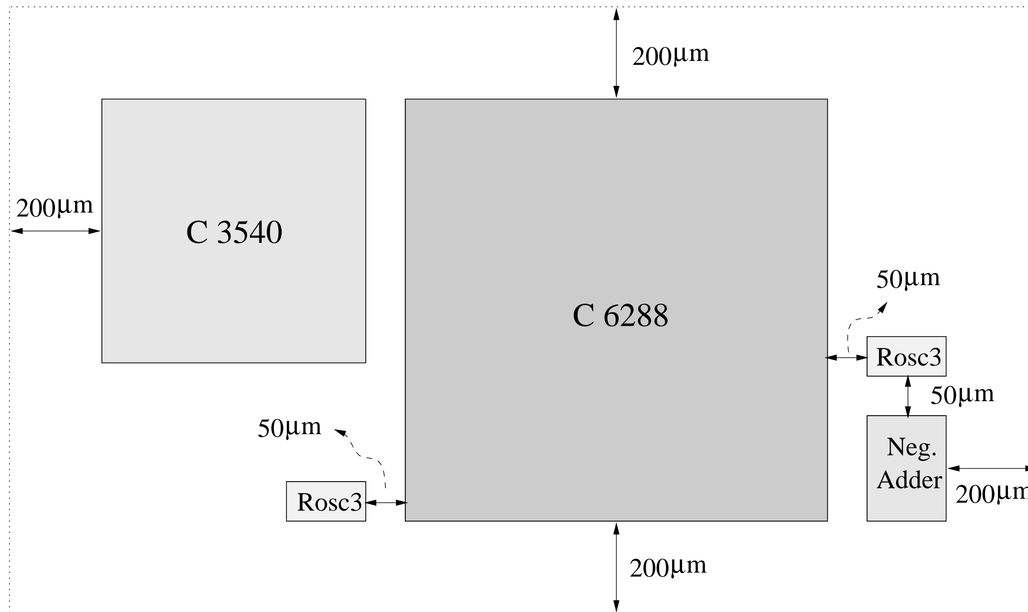


Figure 4.14 Layout of the simulated chip.

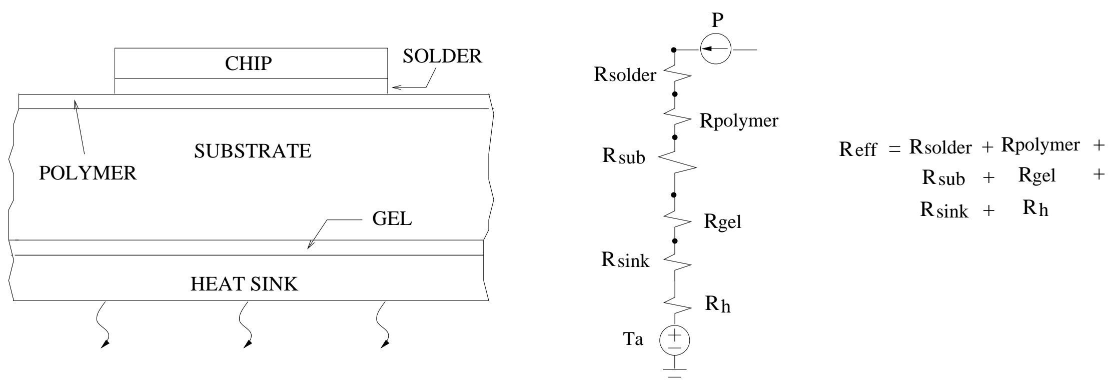


Figure 4.15 Packaging structure used in the simulation example.

ILLIADS-T may serve as a useful tool to ensure that the specified timing constraints are met.

Table 4.5 ILLIADS-T simulation results.

Circuit	Ckt. size	n_{tran}	n_{hsrc}	Freq.	T_{max}	T_{min}	n_{run}^1	CPU[2]
Unit	μm^2	-	-	MHz	°C	°C	-	sec
10-bit Neg. Adder	450 × 300	868	216	100	47.17	38.07	-	-
C3540	1730 × 1540	5842	1656	200	36.82	31.61	-	-
C6288	2180 × 2330	11016	3001	100	44.55	33.10	-	-
							3	3142

[1]Convergence criterion: $(\Delta T/T_{rise}) < \%1$. [2]On SUN SPARCstation 10.

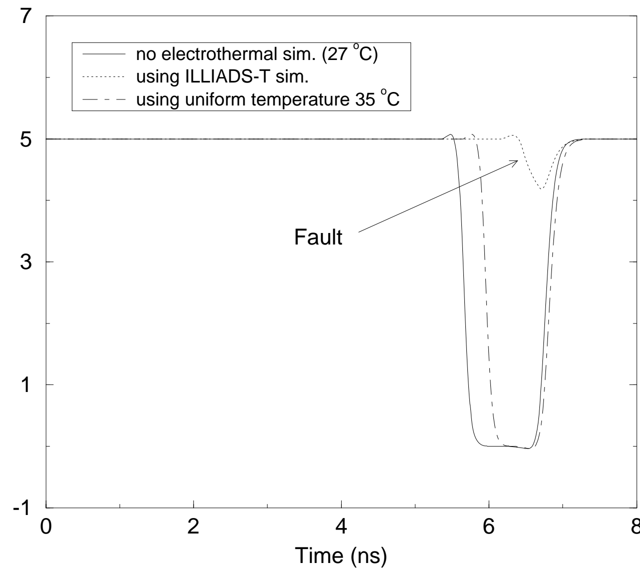


Figure 4.16 Output waveforms of the 10-bit negative adder.

CHAPTER 5

TEMPERATURE-SENSITIVE RELIABILITY AND PERFORMANCE ANALYSIS USING ILLIADS-T

5.1 Motivation

The device packing density in modern VLSI chips increases steadily; therefore, the temperature rise in a packaged chip can be very dramatic. Because many known IC failure mechanisms are either thermally activated or related, on-chip temperature profile must be predicted prior to any reliability diagnosis and performance analysis. In addition, temperature profile plays an important role in the application of thermal stress evaluation and package design at the chip or the printed-circuit-board (PCB) level. Note that, due to the large temperature variation across a packaged chip, the assumption of the uniform on-chip temperature is not acceptable.

In this chapter, we will demonstrate two applications of ILLIADS-T: temperature-dependent electromigration diagnosis and timing analysis. Some other applications such as hot-carrier reliability diagnosis, temperature-driven module placement, and package design can be addressed in the future.

5.2 Electromigration Diagnosis

5.2.1 Introduction

Electromigration (EM) is defined as structural damage caused by ion transport in metal thin films, due to high current densities. Metal ions that have been thermally activated are essentially free of the metal lattice [38]. When a conductor is subjected to a sufficiently high current density, these thermally activated ions will begin to move gradually. It is this movement of atoms that causes damage to the structure of the conductor. In recent years, the metal line width has been scaled down and the increasing current density has aroused serious EM reliability concerns. EM-induced voids lead to resistance increase of the metal line and even a catastrophic failure. Moreover, EM-induced hillock can grow to the point that it forms a short to a neighboring conductor. Both situations will result in the malfunction of the integrated circuits.

SPIDER [39] was the first interconnect reliability diagnostic tool. It takes user-specified transient current to load the metal system at specified contact points. It then extracts the RC network of the system and uses SPICE-like circuit simulator for current waveform computation for each interconnect segment of the system. RELIANT [40] is another EM reliability diagnostic tool. It extracts the RC network and transistors from the given layout. A switch-level simulator is employed to calculate the current drawn by each transistor and the current waveform of each interconnect. The Berkeley reliability tool (BERT) [41] uses a similar approach to [40], but it uses SPICE for accurate transistor and interconnect current calculations. The postprocessor of BERT calculates and reports the failure rates of the interconnects.

In a state-of-the-art chip, the interconnect temperature can rise by as much as 100°C above the ambient temperature attributed to different heat flow mechanisms. These mechanisms include heat conduction from the substrate, heat conduction from the nearby interconnects, and heat generated in interconnect itself (Joule heating). Temperature affects the rate of diffusion of the metal ions because the diffusivity of the ions is exponentially dependent on temperature. This means that ions diffuse more rapidly in areas where the temperature is elevated. If there is a temperature gradient in the direction of current flow, an ion flux divergence will be created. At locations where the temperature increases, vacancies are likely to form. The above physical phenomenon can be described by the well-known Black's equation:

$$\mathrm{MTF} = A \cdot J^{-2} \cdot \exp(E_a/kT). \tag{5.1}$$

In (5.1), MTF is the EM-induced mean time-to-failure, A is a proportionality constant as a function of line length and width, J is the current density, E_a is the activation energy, k is the Boltzmann's constant, and T is the temperature in degrees Kelvin. The ratio of MTF($T = 300K$) over MTF($T = 300K + \triangle T$) as a function of $\triangle T$, based on (5.1), is shown in Fig. 5.1. From Fig. 5.1, it can be observed that if a metal line has a temperature equal to 340 K, its MTF will be twenty times shorter than the MTF when it is subject to room temperature. It also suggests that neglecting the temperature effect on EM failure can substantially overestimate the metal lifetime and lead to unacceptable prediction error. None of the interconnect reliability tools introduced above take into consideration the metal temperature. Therefore, a temperature-dependent EM reliability diagnostic tool is greatly needed.

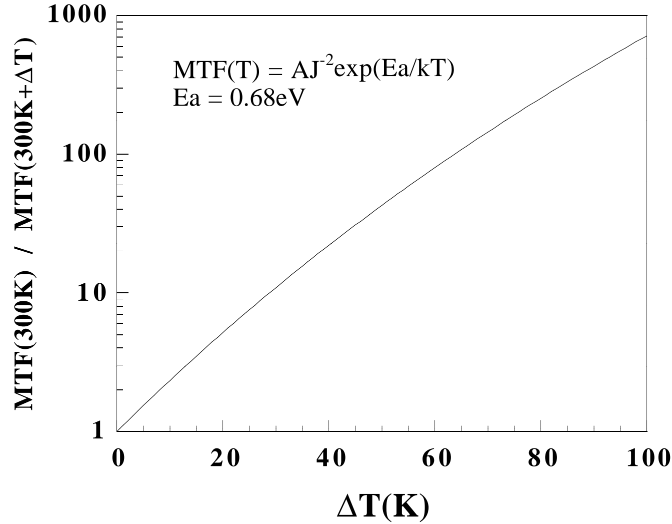


Figure 5.1 The temperature effect on EM reliability.

5.2.2 Temperature-dependent electromigration reliability diagnosis flow

Based on the substrate temperature predicted by ILLIADS-T, we have developed a temperature-dependent EM reliability diagnostic tool, iTEM [42]. Its simulation flow is shown in Fig. 5.2. After the ILLIADS-T electrothermal simulation, iTEM extracts power and ground buses from the layout and identifies the transistors in contact with the ground/power buses. The correspondence between the bus and the transistors that are connected to it is also identified concurrently. The extraction procedures are similar to those used in [43]. Next, iTEM extracts the resistive networks from the buses and builds the admittance matrices for the networks. The currents drawn from the transistors connected to the buses serve as the constant current sources of the networks, and the admittance matrices are solved by using the sparse matrix-solving technique. At this

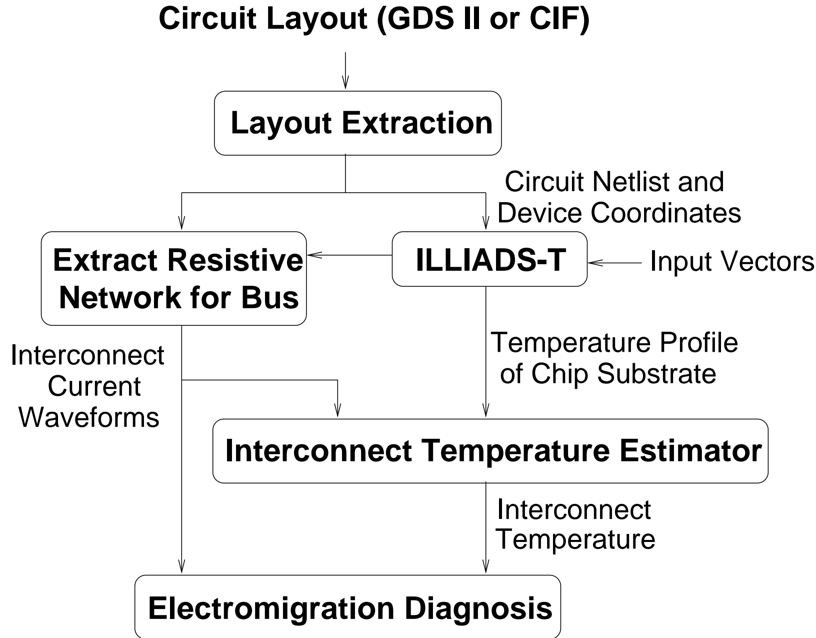


Figure 5.2 Simulation flowchart of iTEM.

stage, the current waveform of each metal rectangle, via and metal-diffusion contact is found for the ground and power buses.

5.2.3 Interconnect temperature estimation

The previous discussion described the procedures for finding the substrate temperature profile and the interconnect current waveform for the ground and power buses. The next step is to estimate the interconnect temperature in order to accurately predict the EM-induced MTF. Here we make the following assumptions for interconnect temperature estimation:

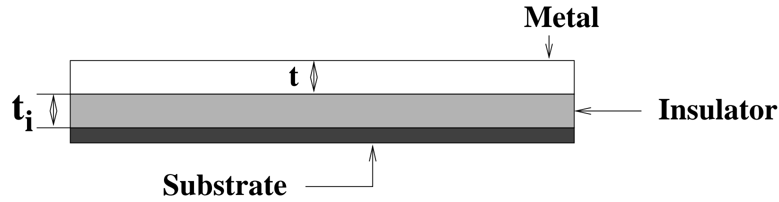


Figure 5.3 The interconnect on insulator structure.

1. The heat conduction from the nearby interconnects can be ignored. In other words, the heat flow mechanisms are dominated by the heat conduction from the substrate and Joule heating.

2. The heat coupling between the substrate and interconnects is ignored. In other words, the interconnect temperature is determined once the substrate temperature profile has been found. No further iteration is performed between the substrate and interconnect temperatures.

The first assumption is generally true because most of the heat generated in the interconnects is conducted away through the bottom substrate to the heat sink. The heat contributed from the nearby interconnects is therefore not significant. The second assumption implies that the interconnect temperature depends on the substrate temperature, but the substrate temperature does not depend on the interconnect temperature. This is practically true because the power consumption of the substrate is much larger than that of the interconnects.

To estimate the interconnect temperature, we first examine the case shown in Fig. 5.3. This structure consists of a long metal wire and an insulator on the substrate. The metal

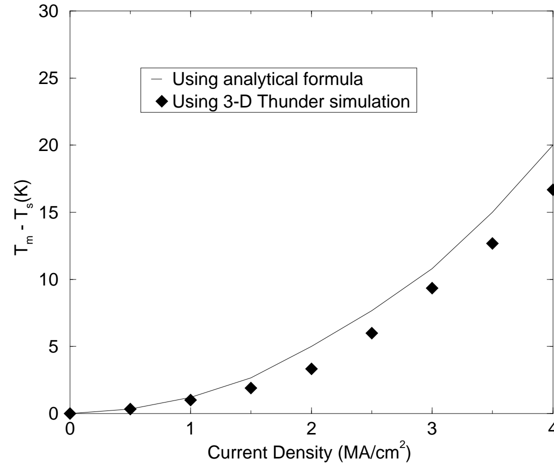


Figure 5.4 Interconnect temperature as a function of its current density, assuming the following physical parameters are used: $t_i = 2\mu$m, $t = 0.5\mu$m, $w = 2\mu$m, $\rho_0 = 3.6 \times 10^{-6}\Omega\cdot$cm, $\beta = 4.04 \times 10^{-3}$K^{-1}, $K_i = 1.835$W/(K$\cdot$m), and $T_s = 300$ K.

width is w and the metal thickness is t. From [44], the metal temperature T_m can be expressed as

$$T_m = T_s + \frac{J^2\rho_0(1+\beta T_s)}{K_{i,eff}/(t \cdot t_i) - J^2\rho_0\beta}, \tag{5.2}$$

where T_s is the substrate temperature, J is the current density, ρ_0 is the resistivity at room temperature, and β is the temperature coefficient of resistivity. The variable $K_{i,eff}$ in (5.2) is the effective thermal conductivity of the insulator by taking into account the fringing effect of heat conduction, which can be written as

$$K_{i,eff} = K_i \cdot \left(1 + 0.88\frac{t_i}{w}\right), \tag{5.3}$$

where K_i is the thermal conductivity of the insulator. The 3-D thermal simulator THUNDER [32] has been used to simulate the structure in Fig. 5.3, and the results predicted by (5.2) match quite well with the 3-D simulation results as shown in Fig. 5.4

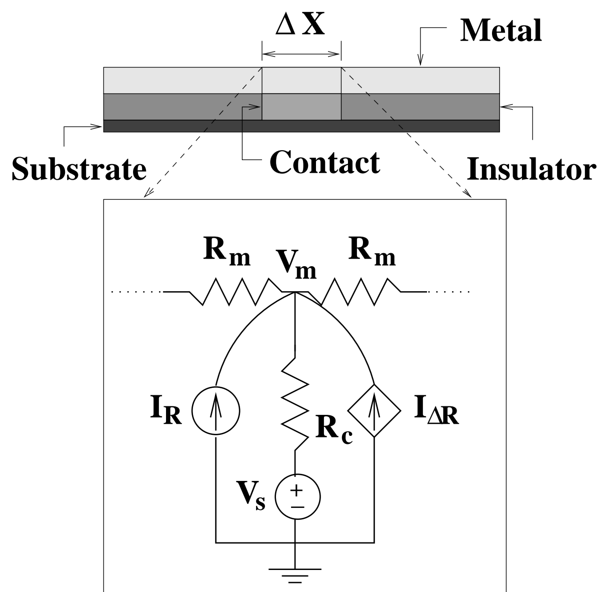


Figure 5.5 Lumped thermal model for a general interconnect structure.

Although (5.2) provides a simple expression for estimating the interconnect temperature with good accuracy, it cannot accurately calculate the interconnect temperature near the contacts or pads [42]. We have developed a lumped thermal model to estimate the interconnect temperature to resolve the above problem. Consider a structure similar to Fig. 5.3 but with an additional contact between the the metal and the substrate, as shown in Fig. 5.5. For a segment of interconnect with length $\triangle x$, we can map the local thermal system into the electrical equivalent network shown in Fig. 5.5. The node voltages V_s and V_m represent the temperatures of the substrate and the metal, respectively, and R_m and R_c denote the thermal resistances associated with the metal and the contact. The sources of Joule heating of the metal are represented by two current sources I_R and $I_{\triangle R}$: I_R is a constant current source which represents the primary Joule heating of the metal, and $I_{\triangle R}$ is a voltage-dependent current source which represents the amount of

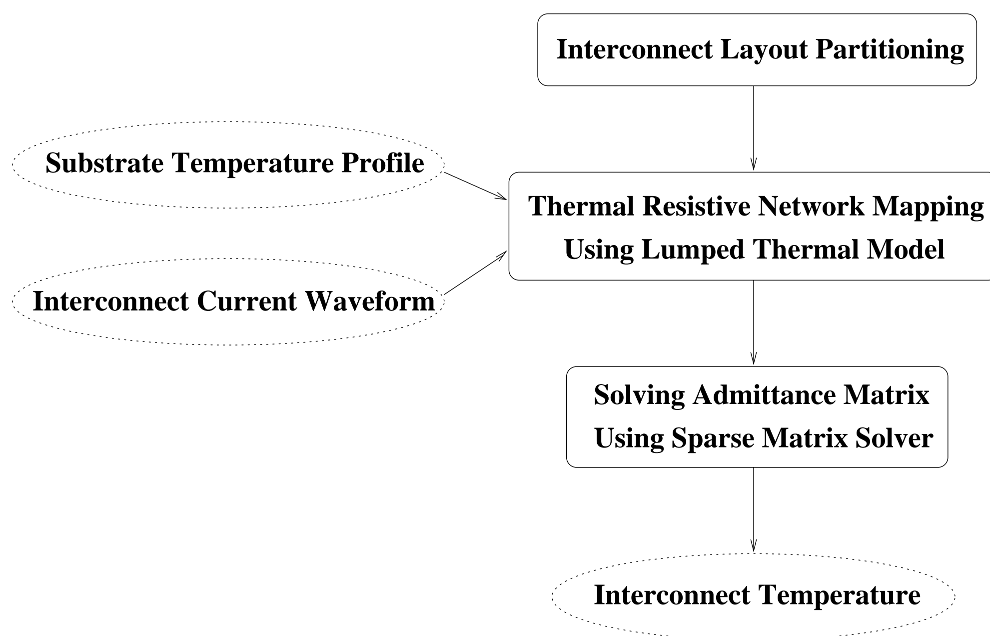


Figure 5.6 The procedures for interconnect temperature estimation.

Joule heating due to the resistivity increase caused by the temperature rise in the interconnect. The accuracy of the lumped thermal model has been verified using THUNDER [42].

Using the lumped thermal model, the procedure for interconnect temperature estimation is shown in Fig. 5.6. The first step is to partition the interconnect layout according to the geometry and contact locations of the layout. After partitioning, every segment of the interconnects is mapped into a thermal resistive network as shown in Fig. 5.5. Thus, a resistive network describing the interconnect thermal system can be obtained. A sparse matrix solver is then executed to compute the temperature of each interconnect.

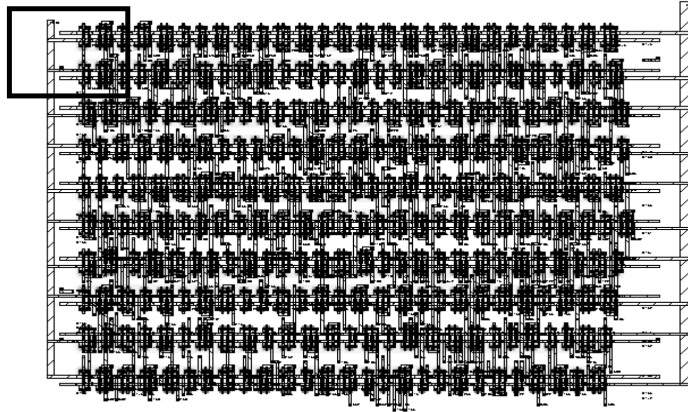


Figure 5.7 The layout of the 10-bit negative adder.

5.2.4 iTEM simulation examples and summary

This temperature-dependent EM diagnostic tool has been tested on several circuits. Consider the 10-bit negative adder introduced earlier in Chapter 4. The layout is again shown in Fig. 5.7. The EM reliability is diagnosed by iTEM, and the predicted MTF for the area inside the square box in Fig. 5.7 is shown in Fig. 5.8. The number marked in the metal is the MTF in hours. Without considering the temperature effect, the MTF is overestimated by as much as seventeen times.

In summary, our electrothermal simulator, ILLIADS-T, has been successfully applied to the temperature-dependent EM reliability diagnostic tool, iTEM. ILLIADS-T first finds the reference substrate temperature for each interconnect, and the interconnect temperature is calculated by using the lumped thermal model. Simulation results show that the estimated MTF would be unacceptably optimistic without taking into account the interconnect temperature.

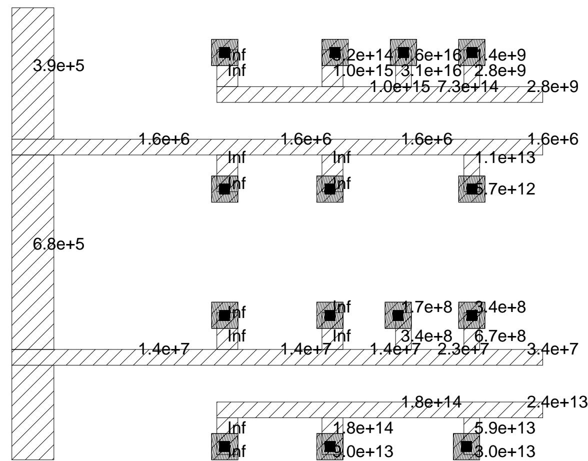


Figure 5.8 iTEM-predicted MTF for the 10-bit negative adder.

5.3 Timing Analysis

5.3.1 Introduction

Timing analysis is an important issue in high-performance ULSI circuit design. Over the last decade, designers have increasingly resorted to timing analysis tools to check whether a given circuit meets the performance goal (i.e., clock speed). Timing analysis of ULSI circuit design consists of checking for short and long path (critical path) problems. The idea of timing analysis stems from the PERT project at IBM [45]. PERT uses the topological sort to find the longest path in the circuit. Since then, the timing analysis research has focused on three major areas: delay model improvement, critical path enumeration, and false path detection. The general timing analysis flow is illustrated in Fig. 5.9.

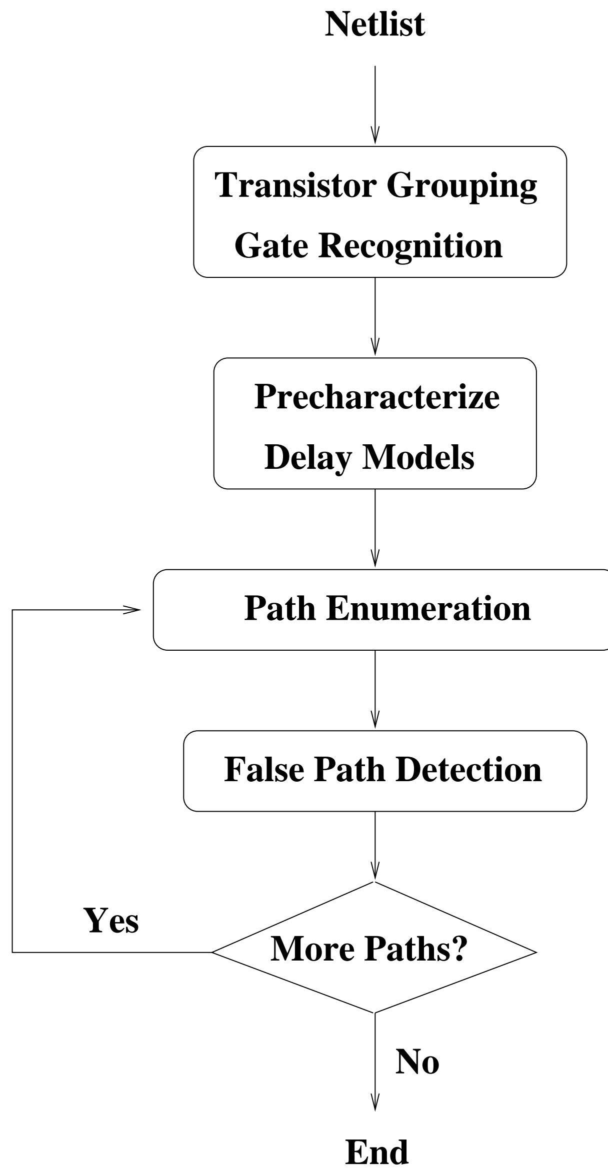


Figure 5.9 Generic flowchart of timing analysis.

The thrust of the delay model improvement area is in the development of more accurate delay models for the use in the timing verification programs such as Crystal [46] and TV [47]. Both programs use the event-driven simulation techniques and model the transistors as the bidirectional switches. More recently, the gate delay modeling with a single switching input has received a lot of attention [48][49][50]. The case of multiple-input switching is just beginning to be addressed in the literature [51][52][53].

The second area, critical path enumeration, has concentrated on the algorithm development to extract the k-most critical paths. One approach is to enumerate only the most critical path, and three types of algorithms have been developed which include the algorithms based on the breadth-first search [54], PERT method [45], and the depth-first search with pruning [46]. The above algorithms are efficient, but extracting only one critical path often fails to provide enough information for correcting the timing violations. Other approaches have tried to enumerate all paths and report the paths that violate the timing constraints [55][56][57]. However, enumerating all paths is an NP-complete problem and these approaches suffer from the path explosion problem. In 1989, Yen et al. developed an algorithm which traces the k-most critical paths [58] and the sorted path delays are reported. A more efficient algorithm using the idea of branch slacks was proposed to extract the k-most critical paths [59].

While the path-enumeration algorithms are quite efficient, they often lead to serious overestimation of the critical path delay due to the *false path* problem. A false path is not a true path along which signals can actually propagate. One example of a false path problem is shown in Fig. 5.10 (from [60]). Path $P = < b, d, e, x, y >$ is considered a false path because in order for signals to propagate from gate d through gate e, c has to be 1

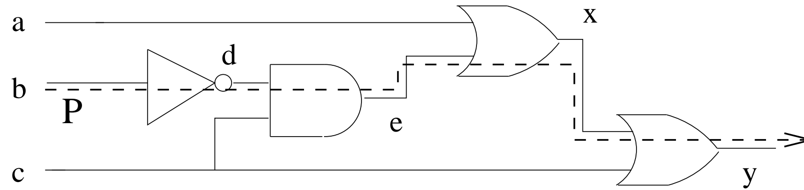


Figure 5.10 False path example.

which blocks signals from x through gate y. Several approaches have been developed to resolve the false path problem, and the most primitive one is called *static sensitization*. A statically sensitizable path is the one that can be activated in isolation from other paths, with all of its side-inputs held at constant noncontrolling values (e.g., 1 for AND gates and 0 for OR gates). In [61], efficient algorithms and a backtracking technique have been utilized to find the statically sensitizable paths. A new idea that totally eliminates the backtracking process, which is usually very costly, has been proposed by Ju et al. [59]. It transforms the sensitization problem into a satisfiability problem and applies the binary decision diagram (BDD) [62][63] to construct the output functions of the paths. Although the BDD package often has a memory shortage problem, the logic functions for all of the internal nodes of the slowest primary output function can be constructed in only a few CPU seconds. Other approaches for solving the false path problem, such as those based on the dynamic sensitization, the viability condition [64], and the Du's criterion [60], have also been proposed.

5.3.2 Temperature-dependent gate and RC delays

From both the experimental and simulation results in Chapter 4, we observe that the on-chip temperature profile substantially affects the circuit delay. The critical path timing, affected by the delays of logic gates and interconnects, is also strongly temperature-dependent. Thus, temperature-sensitive timing analysis is important for high-performance ULSI chip development.

The temperature-dependent gate delay can be calculated using the RWQ and mobility models introduced in Section 2.3. As for interconnects at given temperatures, we use the following equation to find the resistance value for the temperature-dependent RC delay estimation:

$$R(T) = R_0[1 + \alpha_T(T - T_0)], \tag{5.4}$$

where $R(T)$ is the resistance at temperature T, R_0 is the resistance at room temperature T_0, and α_T is the temperature coefficient of resistivity (e.g., $0.004°\mathrm{C}^{-1}$ for aluminum). As a general rule of thumb, the RC delay increases about 5% for a 10°C interconnect temperature rise if the Elmore delay concept is used.

To facilitate the RC delay calculation, we have extended the layout extractor developed earlier [65] to extract the signal-line interconnect resistance in the form of a distributed RC tree, shown as an example in Fig. 5.11. A more indepth description of our delay modeling will be presented in Section 5.3.3.

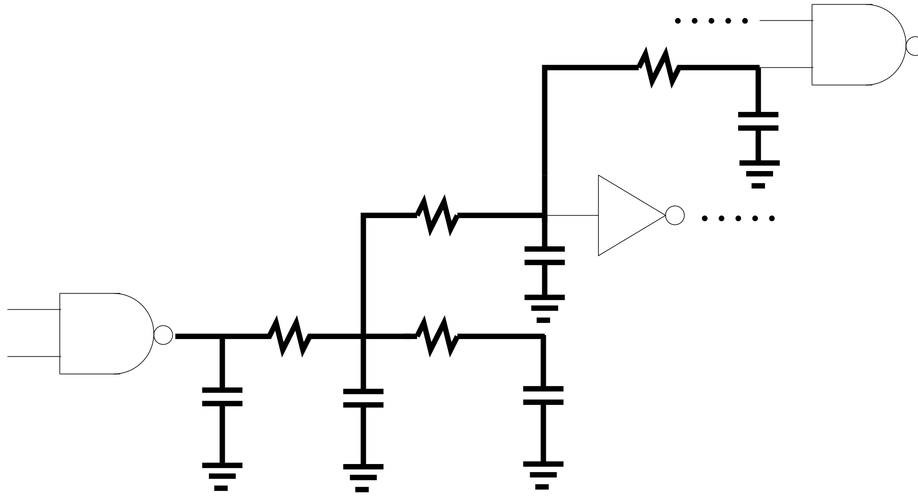


Figure 5.11 Example of a distributed RC tree.

5.3.3 Monte-Carlo power estimation for on-chip temperature profiling

The power-estimation method introduced in Section 2.2.3 is strongly input pattern-dependent because it requires the user to specify complete information about the input patterns. However, input signals are generally unknown during the early design phase and it is practically impossible to estimate the power by simulating the circuit for all possible inputs. In order to estimate the steady-state temperature profile for the temperature-dependent timing analysis, it is more meaningful to obtain the average power in a statistical manner. Therefore, a Monte-Carlo-based approach is more suitable for finding the typical on-chip temperature profile.

Here, we employ the technique called mean estimator of density (MED) [66] for the average power estimation. According to the *central limit theorem*, $\bar{x}$ is a value of a random variable with mean η whose distribution approaches the normal distribution for

large N. With $(1-\alpha)$ confidence, it then follows that [67]

$$-z_{\alpha/2} \leq \frac{\bar{x}-\eta}{\sigma/\sqrt{N}} \leq z_{\alpha/2}, \tag{5.5}$$

where σ is the standard deviation, and $z_{\alpha/2}$ is defined so that the area to its right under the standard normal distribution curve is equal to $\alpha/2$. The random variable $\bar{x}$ is here defined as the sample mean of the *transition density* or the power value of a given DCCB in the circuit. For a sufficiently large number of N (i.e., $N \geq 30$), σ can be approximated by the sample standard deviation s. By using (5.5), one can show that the number of samples needed is

$$N \geq (\frac{z_{\alpha/2}s}{\bar{x}\epsilon_1})^2 \tag{5.6}$$

such that we have $(1-\alpha)$ confidence that the following equation is satisfied.

$$\frac{|\bar{x}-\eta|}{\eta} \leq \frac{\epsilon_1}{1-\epsilon_1} = \epsilon \tag{5.7}$$

In (5.7), ϵ is defined to be a user-specified error tolerance. Thus, (5.6) provides a stopping criterion to yield the accuracy specified in (5.7) with confidence $(1-\alpha)$. It is clear from (5.6) that for a small value of $\bar{x}$, the number of samples required can be very large to meet the specified accuracy level. In MED-like approaches, the stopping criterion (5.6) is used for the DCCBs that have $\bar{x}$ larger than a user-specified threshold value, η_{min}. These DCCBs are referred to as regular-density DCCBs. A different stopping criterion is used for the DCCBs that have $\bar{x}$ less than η_{min}:

$$N \geq (\frac{z_{\alpha/2}s}{\eta_{min}\epsilon_1})^2. \tag{5.8}$$

These DCCBs are referred to as low-density DCCBs. From (5.8), an absolute error bound can be provided for the low-density DCCBs. Although it usually requires longer time for

these DCCBs to converge, they have the least effect on the circuit power and reliability [66].

Additional remarks are noted here. First, we do not consider the external spatial correlation of the input signal, and the circuit is given a sequence of two input vectors for one iteration of logic simulation. All possible input patterns (high, low, high-to-low, low-to-high) are assumed to have equal probability to occur. Second, our logic simulator calculates the power and delay for each gate by taking into account different input slopes, load capacitances, MOS device parameters, temperatures, and interconnects. Each signal-line interconnect network is transformed to an equivalent π-model and is lumped to the corresponding driving gate. The state equation (Riccati differential equation) of the gate is then solved analytically to give the accurate power and delay values [68].

5.3.4 Thermal simulation for timing analysis

To determine the typical steady-state temperature profile of the chip substrate, the gate power values obtained by using the Monte-Carlo simulation are input to our 3-D thermal simulator. Recall that in order to find the steady-state temperature, an iterative procedure should be invoked between the power and temperature calculations (i.e., power and temperature are functions of each other, and the details were discussed in Chapter 1). Here, we provide two different stopping criteria for the iteration process. The first criterion is to limit the number of iterations to two. From our empirical observation, when the iteration count of the Monte-Carlo power and temperature calculations exceeds

two, the resulting temperature values are actually very close to the final steady-state values.

The second criterion is based on two factors: the temperature difference between iterations and the power-estimation error inherited from the Monte-Carlo simulation. Suppose the confidence level $(1-\alpha)$ and the percentage error ϵ are used in the Monte-Carlo power estimation for the regular-density DCCBs. After power simulation in iteration k, the power of each regular-density DCCB is compared with the one found in iteration $k-1$. We calculate the ratio of the number of regular-density DCCBs that have a percentage power difference which is smaller than $2\epsilon/(1-\epsilon)$, to the total number of regular-density DCCBs in the circuit. If this ratio is larger than $1-2\alpha$, the iteration process is stopped. Otherwise, we perform the thermal simulation based on the power distribution in iteration k and find the updated temperature profile. The resulting temperature of each DCCB is then compared with the one in iteration $k-1$, in order to determine whether the iteration can be stopped according to the user-specified accuracy level. The above $2\epsilon/(1-\epsilon)$ term accounts for the possible overestimation and underestimation of the power values inherited from using the Monte-Carlo approach. The value $1-2\alpha$ places an upper bound for the temperature effect to be considered important during iterations.

To find the signal-line interconnect temperature, we first extract the coordinates of each metal. Next, we assign the localized substrate temperature found earlier to the metals at the same X-Y location. It implies that the temperature difference between the substrate and the multilayered signal-line metals is ignored. It is true for a typical signal line carrying a normal amount of current. In such a case, the temperature rise due to Joule heating is relatively small because of the small current density. However, if the

line width decreases or the current increases substantially, the Joule heating needs to be taken into account for the multilayered interconnect system.

5.3.5 Simulation results

Finding the critical path and the possible input patterns that trigger this path involves the work of path enumeration and false path detection. In the current implementation, we do not intend to identify the real critical path. Instead, we assume that the input pattern(s) that triggers the critical path, i.e., *critical pattern*, is given. The temperature-dependent critical path is then identified based on the provided critical pattern.

During the Monte-Carlo power-simulation phase, we concurrently search for the longest path delay and its corresponding input pattern. In other words, if the number of samples needed in the Monte-Carlo simulation is N, the longest delay and its triggering pattern are obtained from the N input patterns. We then regard the currently obtained pattern as the critical pattern which will be used to identify and report the DCCBs along the longest path. It is clear that our critical pattern is acquired as the byproduct of the Monte-Carlo power simulation.

Table 5.1 shows six ISCAS85 benchmark circuits and their functions. In the remainder of this section, we will use these circuits to demonstrate our simulation results. Figure 5.12 shows the thermal boundary conditions used for all circuits under simulation. The four sides are set to be in the isothermal condition, i.e., constant temperatures, the top is perfectly insulated, and the bottom is convective to room temperature with a heat transfer coefficient of $5,000$. Simulation results for the temperature-dependent Monte-Carlo power estimation and critical path delay calculation are demonstrated in

Table 5.1 The ISCAS85 benchmark circuits.

Circuit	Function	#inputs	#outputs	#transistors	#gates
C432	Priority decoder	36	7	1152	364
C499	ECAT	41	32	2266	719
C880	ALU and Control	60	26	1768	538
C1355	ECAT	41	32	2442	703
C3540	ALU and Control	50	22	5842	1656
C6288	16-bit Multiplier	32	32	10706	3001

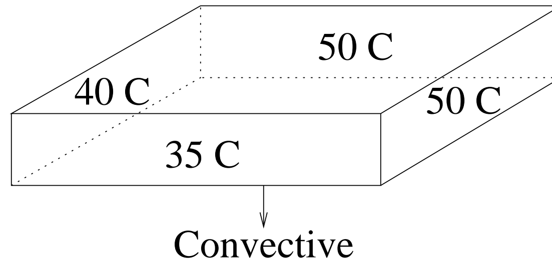


Figure 5.12 Thermal boundary conditions used for the temperature-dependent timing simulation.

Table 5.2 Simulation results of ISCAS85 benchmark circuits.

Circuit	Power	T_{dccb_max}	T_{dccb_min}	Delay(T)	Delay(27)	MC CPU	Therm CPU
Unit	mW	°C	°C	ns	ns	sec	sec
C432	7.1	47.69	36.15	6.68	5.66	228.4	63.8
C499	15.0	49.90	37.14	5.51	4.59	397.6	131.6
C880	11.1	49.43	38.29	4.67	3.92	340.7	96.3
C1355	12.8	51.02	36.20	5.52	4.52	656.8	413.9
C3540	41.1	49.88	35.38	8.39	6.87	1522.9	1331.2
C6288	380.7	50.04	35.23	17.8	14.71	10094	3494

Table 5.2. The power and temperature iterations are limited to two as the stopping criterion. The 95% confidence ($1-\alpha = 0.95$), 5% error tolerance ($\epsilon = 0.05$), and $\eta_{min} = 0.2$ are used in the Monte-Carlo power simulation. The estimated circuit powers are shown in the second column in Table 5.2; T_{dccb_max} and T_{dccb_min} are the simulated maximum and minimum temperatures of the DCCBs on the longest path, respectively. The estimated temperature-dependent longest path delays are shown in the fifth column. The longest path delays, without considering the temperature effect, are listed in the sixth column for comparison. The CPU times (on SUN SPARCstation 10) used for the Monte-Carlo power and thermal simulations are also given in Table 5.2. Finally, the temperature profile of C6288 is demonstrated in Fig. 5.13, and the DCCBs on the longest path of C6288 are also shown.

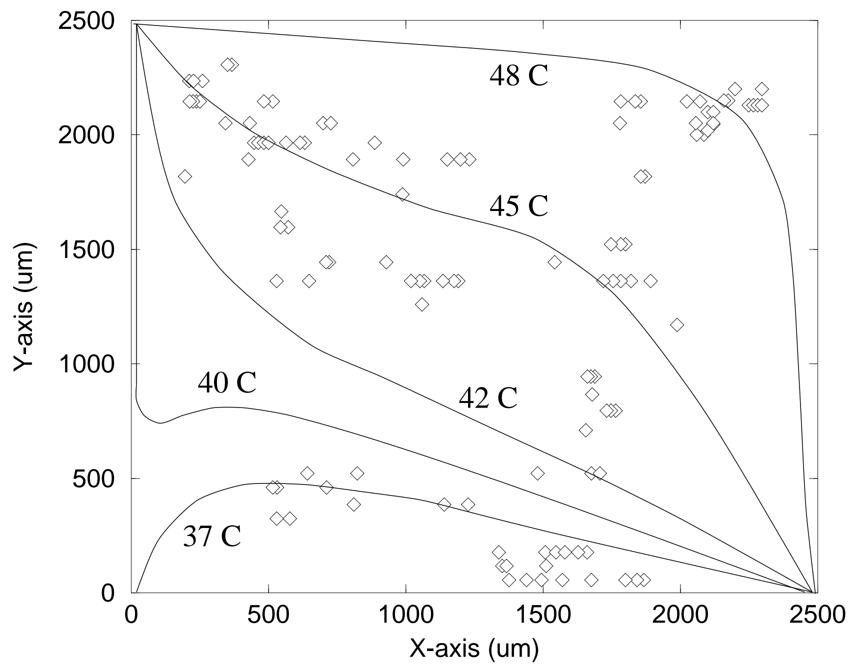


Figure 5.13 The temperature profile and the DCCB distribution on the longest path of C6288. The solid lines are the simulated temperature contour and the small diamonds are DCCBs on the longest path.

CHAPTER 6

CONCLUSIONS

6.1 Summary

In this dissertation, we have presented a methodology for the on-chip steady-state temperature profiling and for the prediction of the thermally induced reliability and performance degradation. This methodology has been implemented in an electrothermal simulator, ILLIADS-T.

ILLIADS-T uses the decoupled power and temperature iterative approach to achieve overall simulation efficiency. Unlike coupled electrothermal simulation, the steady-state temperature can be found in only two to three iterations. ILLIADS-T performs the following major functions: layout extraction, fast-timing simulation, and 3-D thermal simulation. We have developed the temperature-dependent RWQ device models for the accurate delay and power calculation in the fast-timing simulator. A new temperature-dependent mobility model is also proposed. This model takes into account three important scattering mechanisms and it is formulated in a semiempirical form. Our temperature-dependent RWQ device model is particularly useful when only the measured data are available and the SPICE-like models have not yet been fully developed or characterized.

We have developed a complete thermal simulation framework for the steady-state temperature estimation in a chip level. An FTA technique is used to quickly identify

the hot spots in the early chip design phase. The results can also be used for further detailed thermal analysis. A numerical thermal simulator based on the finite-difference method has been presented. It utilizes the adaptive mesh-generating technique to efficiently profile the full-chip temperature. A counterpart analytical thermal simulator has also been developed based on the multiple-integral approach. It focuses on pinpointing the hot-spot temperatures accurately, and the efficiency manifests itself when the temperature calculations are needed only for a few locations. The above thermal simulation techniques have their own advantages and disadvantages and they can be employed for different purposes. An incremental simulation method for speeding up the electrothermal iteration process is proposed. This method is particularly useful when the circuit size or the number of iterations increases.

We have designed a tester chip for the purpose of simulation verification. Our ILLIADS-T simulation results agree well with the experimental data; any discrepancies are generally less than 5%. Simulation correctly predicts both the on-chip temperature and the ring oscillator frequency. The experimental results indicate that thermal effects significantly impact the overall circuit performance and can be accurately predicted by ILLIADS-T. Several other benchmark circuits are also simulated by ILLIADS-T, and its capability to detect the thermally induced faults in signal integrity is demonstrated.

ILLIADS-T can be applied to various temperature-sensitive reliability and performance analyses. Currently, it has been utilized for the temperature-dependent electromigration (EM) diagnosis and the timing analysis. For EM diagnosis, the interconnect temperature is obtained based on the substrate temperature profile calculated by ILLIADS-T. For timing analysis, the Monte-Carlo approach is adopted to estimate the

typical on-chip steady-state temperature profile. Temperature-dependent gate and RC delays are both considered in the timing analysis. Simulation results show that it is important to perform temperature simulation in order to avoid inaccurate diagnosis result and to ensure that the specified timing constraints are met.

6.2 Future Research

There are still many other research topics that need to be addressed in order to enhance the capability of ILLIADS-T. They are summarized as follows.

1. The temperature rise in the signal-line interconnect due to Joule heating is ignored in current implementation of ILLIADS-T. It is a reasonable assumption in most cases. However, when the line width decreases or the amount of current increases, Joule heating can substantially raise the interconnect temperature above the substrate temperature. The capability to handle signal-line Joule heating needs to be incorporated to ILLIADS-T in the future.

2. Due to the large problem size, the current electrothermal simulation approach relies on an iterative procedure which appears to converge after a few iterations. However, no formal proof has been developed for the convergence criterion. It is desirable to develop and apply some tight error bounds in each iteration in order to assure convergence for a broad class of MOS circuits. This development will require an indepth analysis of numerical errors in each module of ILLIADS-T.

3. At present, ILLIADS does not take the IR drops in power and ground buses into consideration. Although the IR drop problem has been considered recently

[69][70][71], for fine-line technology chips with high packing density, the temperature-dependent IR drops need to be evaluated carefully. The direct incorporation of IR drops, however, can significantly lower the computational efficiency of the Riccati equation-solver-based simulation method. It is conceivable that some local transistor mapping technique can be developed and applied to account for temperature-dependent IR drops.

4. ILLIADS-T needs further enhancement in order to handle general path enumeration and false path detection (path sensitization) problems in temperature-dependent timing analyses.

5. Other practical issues also merit further investigations, such as the electrothermal simulation on silicon-on-insulator (SOI) circuits, the temperature-sensitive placement and routing, and the package design.

REFERENCES

[1] K. Fukahori and P. R. Gray, "Computer simulation of integrated circuits in the presence of electrothermal interaction," *IEEE Journal of Solid-State Circuits*, vol. 11, pp. 834–846, Dec. 1976.

[2] S. S. Lee and D. J. Allstot, "Electrothermal simulation of integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 1283–1293, Dec. 1993.

[3] C. H. Díaz, S. M. Kang, and C. Duvvury, "Circuit-level electrothermal simulation of electrical overstress failures in advanced MOS I/O protection devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, pp. 482–493, Apr. 1994.

[4] M. Latif and P. R. Bryant, "Network analysis approach to multidimensional modeling of transistors including thermal effects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 94–101, Apr. 1982.

[5] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, pp. 352–366, Apr. 1990.

[6] G. A. Baker Jr., *Essentials of Padé Approximants*. New York, NY: Academic Press, 1975.

[7] C. H. Díaz and S. M. Kang, "New algorithms for circuit simulation of device breakdown," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, pp. 1344–1354, Nov. 1992.

[8] V. Dwyer, A. Franklin, and D. Campbell, "Thermal failure in semiconductor devices," *Solid-State Electronics*, vol. 33, pp. 553–560, May 1990.

[9] A. Dharchoudhury, S. M. Kang, K. H. Kim, and S. H. Lee, "Fast and accurate timing simulation with regionwise quadratic models of MOS I-V characteristics," in *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design*, Nov. 1994, pp. 208–211.

[10] R. Darveaux, I. Turlik, L. T. Hwang, and A. Reisman, "Thermal stress analysis of a multichip package design," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. 12, pp. 663–672, Dec. 1989.

[11] C. P. Wan and B. J. Sheu, "Temperature dependence modeling for MOS VLSI circuit simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, pp. 1065–1073, Oct. 1989.

[12] A. Gupta, "ACE: A circuit extractor," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1983, pp. 721–725.

[13] A. M. Hill, "Switching density analysis for power and reliability in VLSI circuits," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1996.

[14] Y. H. Shih, Y. Leblebici, and S. M. Kang, "ILLIADS: A fast timing and reliability simulator for digital MOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 1387–1402, Sept. 1993.

[15] Y. H. Shih and S. M. Kang, "Analytic transient solution of general MOS circuit primitives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, pp. 719–731, June 1992.

[16] W. T. Reid, *Riccati Differential Equations.* New York, NY: Academic Press, 1972.

[17] H. Buchholz, *The Confluent Hypergeometric Function.* New York, NY: Springer Verlag, 1969.

[18] R. E. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146–160, June 1972.

[19] J. K. White and A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits.* Norwell, MA: Kluwer, 1987.

[20] S. M. Kang, "Accurate simulation of power dissipation in VLSI circuits," *IEEE Journal of Solid-State Circuits*, vol. 21, pp. 889–891, Oct. 1986.

[21] M. S. Lin, "A better understanding of the channel mobility of Si MOSFET's based on the physics of quantized subbands," *IEEE Transactions on Electron Devices*, vol. 35, pp. 2406–2411, Dec. 1988.

[22] M. S. Liang, J. Y. Choi, P. K. Ko, and C. Hu, "Inversion-layer capacitance and mobility of very thin gate-oxide MOSFET's," *IEEE Transactions on Electron Devices*, vol. 33, pp. 409–412, Mar. 1986.

[23] C. T. Sah, T. H. Ning, and L. L. Tschopp, "The scattering of electrons by surface oxide charges and by the lattice vibrations at the Si-SiO_2 interface," *Surface Science*, vol. 32, pp. 561–575, Sept. 1972.

[24] H. Ezawa, S. Kawaji, and K. Nakamura, "Surfons and the electron mobility in silicon inversion layers," *Japanese Journal of Applied Physics*, vol. 13, pp. 126–155, Sept. 1974.

[25] A. Hartstein, T. H. Ning, and A. B. Fowler, "Electron scattering in silicon inversion layers by oxide and surface roughness," *Surface Science*, vol. 58, pp. 178–181, Aug. 1976.

[26] D. W. Marquart, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, pp. 431–441, June 1963.

[27] M. N. Ozisik, *Boundary Value Problems of Heat Conduction.* New York, NY: Dover, 1968.

[28] V. Koval, I. W. Farmaga, A. J. Strojwas, and S. W. Director, "MONSTR: A complete thermal simulator of electronic systems," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1994, pp. 570–575.

[29] C. A. Balanis, *Advanced Engineering Electromagnetics.* New York, NY: John Wiley & Sons, 1989.

[30] J. F. Thompson, Z. Warsi, and C. W. Mastin, *Numerical Grid Generation.* New York, NY: North-Holland, 1985.

[31] J. W. Brown and R. V. Churchill, *Fourier Series and Boundary Value Problems.* New York, NY: McGraw-Hill, 1993.

[32] *THUNDER User's Manual*, SILVACO Data Systems, Santa Clara, CA, 1993.

[33] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow.* New York, NY: Hemisphere, 1980.

[34] S. L. Su, "Extraction of MOS VLSI circuit models including critical interconnect parasitics," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1987.

[35] J. Kim, S. M. Kang, and S. Sapatnekar, "High performance CMOS macromodule layout synthesis," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1994, pp. 179–182.

[36] *VPNR Users Guide*, Microelectronics Center for North Carolina, 1988.

[37] Y. C. Lee, H. T. Ghaffari, and J. M. Segelken, "Internal thermal resistance of a multichip packaging design for VLSI based systems," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. 12, pp. 163–169, June 1989.

[38] J. R. Black, "Electromigration failure modes in aluminum metallization for semiconductor devices," in *Proceedings of the IEEE*, vol. 57, Sept. 1969, pp. 1587–1594.

[39] J. E. Hall, D. E. Hocevar, P. Yang, and M. J. McGraw, "SPIDER - a CAD system for modeling VLSI metallization patterns," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, pp. 1023–1031, Nov. 1987.

[40] D. F. Frost and K. F. Poole, "RELIANT: A reliability analysis tool for VLSI interconnects," *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 458–462, Apr. 1989.

[41] R. H. Tu, E. Rosenbaum, W. Y. Chan, C. C. Li, E. Minami, K. Quader, P. K. Ko, and C. Hu, "Berkeley reliability tools - BERT," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 1524–1534, Oct. 1993.

[42] C. C. Teng, Y. K. Cheng, E. Rosenbaum, and S. M. Kang, "iTEM: A new electromigration (EM) reliability diagnosis tool using electrothermal timing simulation," in *Proceedings of the IEEE International Reliability Physics Symposium*, Apr. 1996, pp. 172–179.

[43] R. M. Iimura, "iCHARM: Hierarchical CMOS circuit extraction with power bus extraction," M.S. thesis, Dept. of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1990.

[44] H. Katto, M. Harada, and Y. Higuichi, "Wafer-level JRAMP and J-CONSTANT electromigration testing of conventional and SWEAT patterns assisted by a thermal and electrical simulator," in *Proceedings of the IEEE International Reliability Physics Symposium*, Apr. 1991, pp. 85–88.

[45] T. W. Kirkpatrick and N. Clark, "PERT as an aid to logic design," *IBM Journal of Research and Development*, vol. 10, pp. 135–141, Mar. 1966.

[46] J. Ousterhout, "A switch-level timing verifier for digital MOS VLSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 4, pp. 336–349, July 1985.

[47] N. Jouppi, "Timing analysis and performance improvement of MOS VLSI designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 6, pp. 650–665, July 1987.

[48] H. Y. Chen and S. Dutta, "A timing model for static CMOS gates," in *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design*, Nov. 1989, pp. 72–75.

[49] T. Sakurai and A. R. Newton, "Delay analysis of series connected MOSFETs," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 122–131, Feb. 1991.

[50] J. T. Kong and D. Overhauser, "Methods to improve digital MOS macromodel accuracy," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 868–881, July 1995.

[51] A. Nabavi-Lishi and N. C. Rumin, "Inverter models of CMOS gates for supply current and delay evaluation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, pp. 1271–1279, Oct. 1994.

[52] S. Z. Sun, D. H. Du, and H. C. Chen, "Efficient timing analysis for CMOS circuits considering data dependent delays," in *Proceedings of the IEEE International Conference on Computer Design*, Oct. 1994, pp. 156–159.

[53] V. Chandramouli and K. A. Sakallah, "Modeling the effects of temporal proximity of input transitions on gate propagation delay and transition time," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1996, pp. 617–622.

[54] R. B. Hitchcock, G. L. Smith, and D. D. Cheng, "Timing analysis of computer hardware," *IBM Journal of Research and Development*, vol. 26, pp. 100–105, Jan. 1982.

[55] T. Sasaki, A. Yamada, T. Aoyama, K. Hasegawa, S. Kato, and S. Sato, "Hierarchical design verification for large digital systems," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1981, pp. 105–112.

[56] R. B. Hitchcock, "Timing verification and the timing analysis program," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1982, pp. 594–604.

[57] B. Larson, *DAMSEL User's Manual.* Honeywell Corporation, Minneapolis, MN, Apr. 1987.

[58] H. C. Yen, D. H. Du, and S. Ghanta, "Efficient algorithms for extracting the k-most critical paths in timing analysis," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1989, pp. 649–654.

[59] Y. C. Ju and R. A. Saleh, "Incremental techniques for the identification of statically sensitizable critical paths," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1991, pp. 541–546.

[60] D. H. Du, S. H. Yen, and S. Ghanta, "On the general false path problem in timing analysis," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1989, pp. 555–560.

[61] J. Benkoski, E. V. Meersch, L. J. Claesen, and H. DeMan, "Timing verification using statically sensitizable paths," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, pp. 1073–1084, Oct. 1990.

[62] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. 35, pp. 677–691, Aug. 1986.

[63] K. S. Brace, R. L. Rudell, and R. E. Bryant, "Efficient implementation of a BDD package," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1990, pp. 40–45.

[64] P. McGeer and R. K. Brayton, "Efficient algorithms for computing the longest viable path in a combinational network," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1989, pp. 561–567.

[65] C. C. Teng, "Hierarchical electromigration reliability diagnosis for ULSI interconnects," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1996.

[66] M. G. Xakellis and F. N. Najm, "Statistical estimation of the switching activity in digital circuits," in *Proceedings of the ACM/IEEE Design Automation Conference*, June 1994, pp. 728–733.

[67] I. Miller and J. E. Freund, *Probability and Statistics for Engineers*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[68] A. Dharchoudhury, "Advanced techniques for fast timing simulation of MOS VLSI circuits," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1995.

[69] S. Chowdhury and J. S. Barkatullah, "Estimation of maximum currents in MOS IC logic circuits," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, pp. 642–654, June 1990.

[70] H. Kriplani, F. Najm, and I. N. Hajj, "Improved delay and current models for estimating maximum currents in CMOS VLSI circuits," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1994, pp. 433–436.

[71] H. Kriplani, F. Najm, and I. N. Hajj, "Pattern-independent maximum current estimation in power and ground buses of CMOS VLSI circuits: Algorithms, signal correlations, and their resolution," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 998–1012, Aug. 1995.

VITA

Yi-Kan Cheng was born in Taipei, Taiwan, on December 11, 1968. He received the B.S. degree in Electronics Engineering from the National Chiao-Tung University, HsinChu, Taiwan, in 1991. He received the M.S. degree from the University of Southern California, Los Angeles, in May 1993. He served as a research assistant at the Microelectronics Laboratory from 1993 to 1994 at the University of Illinois at Urbana-Champaign where he has been a research assistant at the Coordinated Science Laboratory since 1994. In the summer of 1996, he was with the Technology Computer-Aided Design (TCAD) Department at Intel Corporation in Santa Clara, California. Upon completion of his doctoral degree, he will join the Processor Design Department at IBM Corporation in Austin, Texas.

His research interests include CAD of VLSI circuits and systems, design for IC reliability, and timing simulation. He is a student member of IEEE and the Circuits and Systems Society.